WDK 28

I MECH E

DESIGN RESEARCH – THEORIES, METHODOLOGIES, AND PRODUCT MODELLING

13TH INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN

UNIFYING ENGINEERING DESIGN – BUILDING A PARTNERSHIP BETWEEN RESEARCH AND INDUSTRY

Editors

S Culley, A Duffy, C McMahon, K Wallace

ICED 01 GLASGOW

Professional Engineering Publishing

Design Research – Theories, Methodologies, and Product Modelling

WDK Publications

WDK 1	Principles of Engineering Design
WDK 2a	Bibliography of Design Science
WDK 2b	Bibliography of Design Science (continued)
WDK 3	Terminology of the Science of Design Engineering in Six languages
WDK 4a	Case Examples (1–3)
WDK 4b	Case Examples (4–6)
WDK 4c	Case Examples (7–9)
WDK 5	Design Methodology: Proceedings ICED 81, Rome
WDK 6	Teaching Engineering: Proceedings ICED 81, Rome
WDK 7	Conference Results – ICED 81, Rome
WDK 8	J. Dietrych about Engineering Design
WDK 9	Tactics in Engineering Design (Readings)
WDK 10	Proceedings ICED 83, Copenhagen
WDK 11	Management of Engineering Design (Readings)
WDK 12	Proceedings ICED 85, Hamburg
WDK 13	Proceedings ICED 87, Boston
WDK 14	Methodical Design of Machine Elements (Readings)
WDK 15	Reliability of Technical Systems
WDK 16	Proceedings ICED 88, Budapest
WDK 17	EVAD – Evaluation and Decision in Design (Readings)
WDK 18	Proceedings ICED 89, Harrogate
WDK 19	Proceedings ICED 90, Dubrovnik
WDK 20	Proceedings ICED 91, Zurich
WDK 21	Engineering Design Education (Readings)
WDK 22	Proceedings ICED 93, The Hague
WDK 23	Proceedings ICED 95, Prague
WDK 24	EDC - Engineering Design and Creativity, Workshop Proceedings
WDK 25	Proceedings ICED 97, Tampere
WDK 26	Proceedings ICED 99, Munich
WDK 27	Manual for Design Engineering (Selected preprint)
WDK 28	Proceedings ICED 01, Glasgow

WDK

ICED 01 GLASGOW

I MECH E

13th International Conference on Engineering Design – ICED 01

Design Research – Theories, Methodologies, and Product Modelling

21–23 August 2001
Scottish Exhibition and Conference Centre, Glasgow, UK

Organized by
The Institution of Mechanical Engineers (IMechE)

Co-sponsored by
The Institute of Engineering Designers
The American Society of Mechanical Engineering (ASME)

Professional Engineering Publishing

Published by Professional Engineering Publishing Limited for The Institution of Mechanical Engineers, Bury St Edmunds and London, UK.

First Published 2001

ISBN 1 86058 354 7

A CIP catalogue record for this book is available from the British Library.

Printed by The Cromwell Press, Trowbridge, Wiltshire, UK

Conference Organizing Team

Scientific Advisory Board

Industrial Advisory Board

Contents

Preface *xv*

Invited paper 1

The contribution of design research to industry – reflections on 20 years of ICED conferences
M M Andreasen 3

Design Theory and Methodology

Nature and Philosophy of Design **13**

A contemporary survey of scientific research into engineering design
I Horváth 13

Genetic analogies in design
T Bercsey, S Vajna, and P Mack 21

Design research in perspective – a meta-research on ICED 97 and ICED 99
M Cantamessa 29

Curiosity-oriented reseach in engineering [CORE] (design)
A Samuel and W Lewis 37

Design Theory **45**

A measure for design coupling
G Arcidiacono, G Campatelli, and H Lipson 45

Analysis of conceptual design chains for the unknown input/known output pattern
R Žavbi and J Duhovnik 53

Quantitative visualization and evaluation of thought processes based on importance factors connectivity map in checking a detail drawing
S Warisawa, R M Willings, and Y Ito 61

Functions revisited
V Hubka and W E Eder 69

Design Theory and Methodology ***77***

The application of semiotics in the understanding of synthesizing product quality in design
M W Lange 77

Design syntactics – a contribution towards a theoretical framework for form design
A V Warell 85

Elementary level in the functional-technological design
C Opran and A Armeanu 93

Verification of a new model of decision-making in design
C T Hansen 101

Complexity **109**

Strategies for solving complex problems in engineering design
W Lewis, J Weir, and B Field 109

Topological structures for modelling engineering design processes
D Braha and Y Reich 117

Towards a theory of complicatedness – framework for complex systems analysis and design
V Tang and V K Salminen 125

The case for process cockpits in mechanical engineering and plant manufacturing
M Gruenenfelder and A P Hofer 133

Complexity Management **141**

Improving the product by accelerating the process
J Jarrett and P J Clarkson 141

Complexity in planning design processes
C Earl, J Johnson, and C Eckert 149

A complexity index for the design process
S Sinha, A I Thomson, and B Kumar 157

Reducing complexity of modelling in large delivery projects
H Laurikkala, E Puustinen, E Pajarre, and K Tanskanen 165

Successful strategies for complex design in early phases of the product development process – an empirical study
B Bender, U Kammerer, F P Pietzcker, W Hacker, and L T M Blessing 173

Research Methodology **181**

A methodology for computational design tool research
R H Bracewell, K Shea, P M Langdon, L T M Blessing, and P J Clarkson 181

Establishing casual links between success characteristics and project outcome
C Baumgärtner and L T M Blessing 189

A systematic approach to observations, analysis, and categorization of design heuristics
B Bender, U Kammerer, F P Pietzcker, W Hacker, and L T M Blessing 197

Design project management in practice – the e-mail diary study
D J H C Hendriks and K Dorst 205

A novel model of learning in design
S K Sim and A H B Duffy 213

The Methods Lab – a user research methods typology for an inclusive design process
A S Macdonald and C S Lebbon 221

Verification of a model of synthesis – the methods for verification and results
A Tsumaya, Y Nomaguchi, M Yoshioka, H Takeda, T Murakami, and T Tomiyama 229

Reflective Practice 237

Improving requirement satisfaction ability of the designer
S Nidamarthi, A Chakrabarti, and T P Bligh 237

Design, science, and the problems of underdetermined problems
K Dorst 245

Reflective practice as a way of observing and analysing product design teams
R C Valkenburg 253

Some useful metaphors for the designing engineer
P M Winkelman and O R Fauvel 261

Heedful action, reflection, and transfer in the design process
K Lauche 267

Models of Design Processes 275

Object-oriented modelling of the design process
N Pavkovic, D Marjonovic, and D Dekovic 275

Patterns of product development interactions
S D Eppinger and V K Salminen 283

Implementation of Petri net based design network on the automation of mechatronic design
S Güroglu, Z Erden, and A Erden 291

Model of the design process – a system approach for creating protectable solutions
J Käosaar 299

A general framework for modelling of synthesis – integration of theories of synthesis
H Takeda, M Yoshioka, and T Tomiyama 307

Design activity modelling – a performance viewpoint
F J O'Donnell and A H B Duffy 315

Application of dissipative structures to improve the generation and selection of new product ideas
C Marxt 323

Dynamic concept development – a key for future profitable innovations and new product variants
S Ottosson 331

Mass customization of creative designs
J S Gero 339

Product innovation on demand – fiction or truth?
U Lindemann 347

Experience of creativity enhancement teaching in industry and university
G Thompson and M M Lordan 355

Classification and effectiveness of different creative methods in design problems
E Mulet and R Vidal 363

Innovation in the tension of change and reuse
M Gerst, C M Eckert, P J Clarkson, and U Lindemann 371

Towards hybrid methods for synthesis
A Chakrabarti 379

Creativity and Innovation in Teams **387**

'Innoplan' – an adaptation of the metaplan-technique for a novel computer supported method of teamwork
T Kennel, A M Kunz, and S Müller 387

Understanding and supporting innovation in teams
K Lauche, R Ehbets Müller, and K Mbiti 395

Collaborative design – exploring new interface technologies
K Lauche and K Höger 403

Product and Systems Modelling

Product Data Management **413**

Visualization of product structure and product architecture for a complex product in a mass customization company
T K P Holmqvist 413

Managing mechatronic simulation models of technical products with PDM-systems
M Krastel and R Anderl 421

Reducing design development cycle by data management within the design office
M Storga, D Pavlic, and D Marjanovic 429

An engineering workbench for complex products, based on a hybrid product model
M Koch and H Meerkamm 437

Sketching/Vague Models 445

Towards 'Sketch three-dimensional prototyping' for aiding conceptual form design
J C Borg, K P Camilleri, P Farrugia, F Giannini, and J Muscat 445

Intelligent computational sketching support for conceptual design
S Lim, A H B Duffy, and B S Lee 453

Sketching behaviour and creativity in conceptual engineering design
M Pache, A Römer, U Lindemann, and W Hacker 461

Knowledge representation and processing in abstract prototyping of design support tools
E Z Opiyo, I Horváth, and J S M Vergeest 469

Functional Modelling 477

Enhanced systematics for functional product structuring
U Pulm and U Lindemann 477

Creating virtual prototypes – integrating design and simulation
S Finger, X Chan, R Lan, and B Chan 485

Functional product modelling – new methods for the generation of product functions
R Stetter, G Aßmann, and M Viertlböck 493

Creating equation handbooks to model design performance parameters
C R Bryant, M A Kurfman, R B Stone, and D A McAdams 501

Towards a mechanical systems modelling and optimal embodiment method
B Eynard and P Lafon 509

A computational model for supporting conceptual design of automatic systems
E A P Santos, V J de Negri, and J E R Cury 517

An environment decomposition-based approach to design concept generation
Y Zeng and P Gu 525

CAD 533

Product shape image creation system for industrial design using emergent process
K Sakita and M Igoshi 533

Modelling and evaluation of principle solutions of mechatronic systems, exemplified by tyre pressure control in automotive systems
J Gausemeier, M Flath, and S Möhringer 541

Using the skeleton model for preliminary geometrical synthesis of three-dimensional cinematic chains
J-C Wahl, M Sartor, and J-C Fauroux 549

A practical methodology for integrating software development and empirical techniques in design
P M Langdon, R H Bracewell, L T M Blessing, K M Wallace, and P J Clarkson 557

Shape manipulation of domain distributed vague models
Z Rusák, I Horváth, J Jansson, J S M Vergeest, and G Kuczogi 565

Analysis product devlopment tools by looking at their life-cycle
C S Araujo 573

Features **581**

Interactive model for power transformer's core design
D Žeželj, N Bojčetić, and D Marjanović 581

A feature-based system for apparatus engineering concepts and solutions
M Zirkel, S Vajna, M Jandeleit, and K Strohmeier 589

Feature extraction tools on polyhedral models for machining applications
M Labrousse and P Veron 597

Evaluation and generation of attachment concepts based upon a feature-based solid model using a feature-based recognition strategy
D H Baxter and G A Gabriele 605

Feature- and constraint-based design of solution principles
T Brix, B Brüderlin, U Döring, and G Höhne 613

Redesigning with feature grammars and FBS models
S C Chase and P Liew 621

The use of distributed viewpoint-dependent feature-based modelling and the response surface method in design assessment
K Lee, I Kaymaz, and C A McMahon 629

Virtual Reality **637**

Web-based design and manufacturing of custom mannequin model
T K K Chang, C C L Wang, and M M F Yuen 637

6DOF manipulators for curves and surfaces bi-manual modelling in virtual environment
M Fiorentino, G Monno, and A E Uva 645

An assembly tool to support the design of complex products
L Bongulielmi, S Dierssen, U Leonardt, and M Meier 653

Communication to accelerate product development in a collaborative engineering environment
K-H Grote and I Kimura 661

Visualizing the impact of tolerances on cosmetic product quality
N P Juster, M R Fitchie, S Taylor, P Dew, J Maxfield, and J Zhao 669

Industrial design engineering – facilitating communication of product properties using web-based virtual reality
P Schachinger 677

Choosing the best visualization tool in engineering design – comparing high-immersive VR with desktop-VR
P Johansson 685

Innovation using the digital product – the use of virtual reality in product development processes
A Kunz and M Meier 693

Integration **701**

Exchanging administrative product data in the automotive industry using step
K Pagenstert, H Axtner, D Lange, and P Deasley 701

Integrated development of gear units – produce- and workflow modelling
A Dyla, B-R Höhn, and K Steingröver 709

Providing supplier-components data as PDM-objects in web-based systems
D Nötzke and H Birkhofer 717

Towards 'design definition management'
S M Barker 724

Authors' Index **733**

Related Titles of Interest

Title	Editor/Author	ISBN
IMechE Engineers' Data Book – Second Edition	C Matthews	1 86058 248 6
Constraint-aided Conceptual Design (Engineering Research Series ERS)	B O'Sullivan	1 86058 335 0
Managing Engineering Knowledge MOKA Methodology for Knowledge	M Stokes	1 86058 295 8
Design for Excellence – Engineering Design Conference 2000	S Sivaloganathan and P T J Andrews	1 86058 259 1
Design Reuse – Engineering Design Conference 1998	S Sivaloganathan and T M M Shahin	1 86058 132 3
Industrial Applications of Environmentally Conscious Design	T C McAloone	1 86058 239 7
Integrated Design Education Beyond 2000	P R N Childs and E K Brodhurst	1 86058 265 6
Quality, Reliability, and Maintenance 2000	G J McNulty	1 86058 256 7
Design Management – Processes and Information Issues	S Culley, A Duffy, C McMahon, and K Wallace	1 86058 355 5
Design Methods for Performance and Sustainability	S Culley, A Duffy, C McMahon, and K Wallace	1 86058 356 3
Design Applications in Industry and Education	S Culley, A Duffy, C McMahon, and K Wallace	1 86058 357 1
Virtual Design and Manufacture	IMechE Seminar	1 86058 314 8

For the full range of titles published by Professional Engineering Publishing contact:

Preface

This is one of four books resulting from the contributions to the 13th International Conference on Engineering Design (ICED 01). The conference was held in August 2001 in the Scottish Exhibition and Conference Centre located on the River Clyde in Glasgow – the ideal place to hold the first ICED of the new millennium.

The ICED conference series was initiated by Workshop Design-Konstruktion (WDK) in 1981 with the first conference in Rome. From the very beginning, the aim of ICED was to offer a platform for the discussion of new trends, developments, and research findings in the areas of new product development, design support techniques, design processes, design science, and design education.

The conferences have been held in eleven different countries and have become one of the most pre-eminent conferences in the field of Engineering Design, with the last two conferences being held in Tampere, Finland, in 1997 and in Munich, Germany, in 1999. Both these conferences attracted well over 500 delegates from both academic institutions and industrial organizations. Nearly all the leading authorities in the field of Engineering Design attend to report their latest findings and exchange current ideas with colleagues.

All conferences have focussed on the process of planning, developing and designing technical systems and products. ICED covers all aspects and disciplines of engineering design, from general product development and innovation to feature-based geometric reasoning and design for later life-phases. As engineering design is a process to which many disciplines are contributing, an additional emphasis has been placed on design management, organization, teams, and individuals. Over the years ICED conferences have become the forum for establishing, maintaining, and improving contacts and co-operation between researchers and engineers from countries all over the world.

It is self evident that the engineering design process has changed to meet the challenges of globalization, increasing international competition, and the need for sustainable development. Equally the performance and quality of engineering products have improved in many aspects, time to market, performance, reliability, reduced environmental impact, etc. If the improvements are to be maintained, the elements that contribute to the product development process must continue to be studied and enhanced.

Improvements in the engineering design and its process have been supported by theories and methods developed by research groups around the world. The research is beginning to mature into an overall and consistent understanding of engineering design as will be seen in the pages of the four books. However the results are still fragmented and there is a need to unify the findings, and to ensure that these findings are transferred into industry.

The theme chosen for ICED 01 was **Unifying Engineering Design** – Building a Partnership between Research and Industry.

The organizing team received 664 Abstracts and this resulted in some 325 full papers. All papers are eight pages in length and went through a double blind review of the abstracts and a double review of the full papers. The books consist of contribution papers from some 35 countries.

DESIGN RESEARCH

This book consists of some 17 topics and has an overarching theme of what can be considered research frameworks and representations. It commences with an invited paper by Professor Mogens Andreasen that reviews the contribution of ICED conferences to engineering design research and practice. It then focuses on the theory and methodology issues associated with research in the Engineering Design area. It also covers the research and development issues associated with product and system modelling.

Books in the series
Book 1 *Design Research*
Book 2 Design Management
Book 3 Design Methods
Book 4 Design Applications

A large number of people and organizations have helped with the conference. The organizing team would like to express their thanks to all who have contributed to the content and execution of ICED01 in whatever way.

Steve Culley Alex Duffy Chris McMahon Ken Wallace

Invited Paper

This page intentionally left blank

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

THE CONTRIBUTION OF DESIGN RESEARCH TO INDUSTRY – REFLECTIONS ON 20 YEARS OF ICED CONFERENCES

M M Andreasen

1 Introduction

The ICED conferences have been arranged since 1981 in a biannual pattern and performed at different locations, hosted by local teams or institutions, (see the reference list [1]). The conference ownership has now been transferred from WDK [2] to the Design Society (DS). The author has been involved in the planning of all of these ICED conferences (and a couple of intermediate national ICED conferences), and has been asked to comment on the design and development over time of the conferences, in order to make a balance before the next 20 years of challenges for DS's ICED conferences.

The ICED conferences have been described by WDK as a "supermarket for exchange of ideas". But I have always believed, that ICED gave a trust-worthy picture of state-of-the-art in design research, methodology and teaching engineering design. A difficult question to answer is to what degree ICED has influenced industry and has been influenced by industry.

It is easy to see the shifting set of topics of the ICED conferences over time, but also to see the fixed core of themes. I believe this shifting focus is mirroring the changing industrial situation and agenda. But also the shifting technology push, influencing what researcher takes up in their research, for instance in the IT area.

The main influence from ICED and from design research is, I believe, through candidates and direct cooperation between academia and industry. But this influence is, as stated in several ICED papers, unfortunately low, as far as the use of methodics is concerned.

This paper focuses upon the changing direction and content of the ICED conferences over time. Do these rather industry-independent conferences mirror the industrial agenda? To what degree does ICED influence industry's way of designing? Shall we change so that ICED is designed on industry's premises?

The paper will introduce some facts about the design and content of the past conferences and try to consolidate some of the beliefs, seen as hypotheses, above.

2 Designing ICED conferences

Each ICED conference is designed by a relatively small team of 2-6 people. The design task consists principally of the following activities:

- Formulating the mission, vision and topics of the conference
- Articulating topics and criteria for papers in the 'call for papers'
- Selecting abstracts and evaluating papers
- Tentatively building abstracts, and later papers, together into a programme of streams and sessions.
- Registering and acting upon the audience's participation in sessions.

What is influencing the final result, i.e. the programme executed and the sessions selected by the audience? The following factors are relevant:

- The team behind the conference
- The call-for-paper articulation
- The papers coming in
- The configuration of the conference

Let me comment on each of these factors.

The conference design team. ICED conferences are arranged by local groups in cooperation with WDK core group members (and from now on from the Design Society). Hereby the WDK mission and vision are mixed with a certain national flavour, a conference mission and believed important, sellable actualities, into the conference's call-for-paper articulation.

Even if the planning of a conference has been proceeded with a balance of the last conference's goal and result, each new conference has been designed with an optimism of being able to bring "new papers on new topics", showing the believed direction of designing's state-of-the-art. It is interesting to register each ICED conference's sparse response from authors to "new and modern" topics in the call for papers, showing that there seem to be other and stronger forces directing design research and state-of-the-art.

The call for papers. Even if the core and identity of the ICED conferences has been maintained by a set of characteristic ICED topics, the number of papers per topic has varied substantially over the years, showing changing response. New topics have been taken up by the authors, determined by their current research, mirroring the market pull on the conference to much higher degree than a conference design push.

The papers submitted. ICED has a loyal group of customers delivering papers and at each conference a substantial group of newcomers. Only to a certain degree, and mostly by national efforts, have the authors, actively been requested to deliver contributions. But even if the pattern seems relatively static, the topics covered are moving.

The configuration of the conference. An absolute evaluation of each paper based upon the criteria is nearly meaningless. There is a necessity to bring the papers into groups showing if there is something substantial so that a stream or sessions can be configured based upon the papers. If a stream topic is taken up, there needs to be strong and complementary papers; singular papers are difficult to join into a pattern.

It is obvious to those who follow ICED, that the configurating is a difficult activity often ending up with a messy programme. Several facts make it difficult:

- Each paper has many viewpoints. The seemingly primary topic may, for instance, be 'evaluation methods', but the author has used 'computer-implementation', and the author insists that the considerations are valid for 'civil engineering'. Shall the paper belong to 'design methodics', 'computer applications' or 'special design theories'?
- The authors do not help by their own classification. The authors' classification is seldom in accordance with the reviewers' classification. It is a fact, the ICED authors use an enormous number of keywords for identifying their papers, i.e. the keywords are not used to show kinship, but derivation or uniqueness.

Concluding this section about conference design, I postulate, that the influence on the design, (i.e. topics, streams) from the people behind the conference is quite sparse. The conference topics have their own development and the foci take their own direction.

3 Method of analysis

My analysis consists of classifying and finding patterns in contributions from the ICED conferences based upon the list of contents from each conference. The classification should establish the following types of results:

- Classification by topic
- Identification of changing topics and new topics
- Identification of main approach or *Zeitgeist* of each conference
- Classification by research agenda or industrial agenda.

The information necessary for these classifications and identifications is only partly available or derivable from the lists of contents. This means that my insight and interpretation influences the analysis. The classification is also subject to fuzzyness related to design topics. It is my experience from the many tasks as programme organiser and reviewer that the use of terminology in the design area is soft and contradictory and the words used seem to change meaning over time. For instance Design for Manufacture has been defined by some authors as the synthesis of a solution with focus on manufacturing aspects, while others see it "only" as a set of manufacturing criteria.

For the analysis a list of topics is established, based mainly upon the conference themes (ICED'91 taken as a main reference), but the allocation of the papers to these topics are revised:

- Many papers (more than 30%) are incorrectly placed in the topics and sessions. The reasons for this include a false interpretation of the topics by the authors; or the chair-man present in the planning, who "construct a conference within the conference".
- There has been a strong tendency to separate papers related to computers from the rest of the conference, even if the main topic has been, for instance, evaluation (supported

by software). I have redefined the allocation of these papers. Therefore the analysis here shows a much lower number of computer-related papers than the proceedings.

- The papers investigated are oral, poster and workshop-papers.

4 Patterns found in the ICED conferences

In the following, four main trends are identified and commented related to: design research; the relation between engineering design and product development; the role of computers and information technology; and the delimitations of ICED topics.

The articulation of a design science concept

The early conferences identify the methodics of designing and bring it to practice, showing that it works in singular cases. The concept of a scientific basis and a research methodic was gradually developed and in several areas the contributions gradually (therefore making them difficult to isolate) turn from generalisations and prescriptive statements to descriptive statements.

Emerging empirical studies in the 90's give us valuable insight into discrepancies between theory and industrial practice. Papers on testing methods and analysing their usefulness in practice are growing in number, and in '99 we suddenly have several papers on design capability and measurement.

At ICED '97 a pamphlet [3] is published on design research, underlining the emerging paradigm of design research and the necessity of a scientific rigour in our research. Figure 1 shows the relevant analysis for the statements above.

From Methodics to Design Science	81	83	85	87	89	91	93	95	97	99	01
Design Methodics (Design process theory and methods)	19	31	30	26	14	44	41	42	44	38	55
Methods in Practice	2	13	7	2	9	16	8	20	15	8	5
Theory Contributions (Design Science, TTS, neighbour theories)		10	4	7	7	14	11	25	34	14	6
Research Methods	1	2			1			1	5	4	9
Testing, usefulness of methodics		2	4	2	2	5	4	4	7	8	6
Capabilities, measuring										13	6
Total	22	58	45	37	33	79	64	92	105	85	87
Relative to total number of papers %	25	50	39	30	34	33	24	26	29	22	27
Total number of papers	86	115	114	123	97	235	267	347	361	390	324

Figure 1. The changing focuses from development of methodics to scientific verification and proper research methodics from 1981 to 2001. DFX is omitted in the methodics balance, see below.

The first ICED '81, in Rome, mirrors clearly the purpose of WDK and the focus on the profession of the engineering designer (German: Konstrukteur). Figure 1 shows the focus on the methodics and the design process, and there is a balance between papers on conceptualisation, layout and detailed design.

In the "Design for X"-area, i.e. the methodics for fitting design to different product life areas and different important product property classes, the majority of papers are related to manufacture and cost in the early ICED conferences. Later the scope broadens to several areas and Design for Environment shows a high increase, see Figure 2.

Design for X	81	83	85	87	89	91	93	95	97	99	01
Manufacture, Cost	3	3	10	9	6	21	24	32	20	11	5
Quality, Reliability, Maintenance	1	8	7	5	6	13	20	15	16	8	11
Environment, ecology						2	14	15	19	16	20
Others				2		3	11	4	4	5	6
Total	4	11	17	16	12	39	69	66	59	40	42
Relative to total number of papers %	4	9	15	13	12	16	26	19	16	10	13

Figure 2. The emergence of DFX areas and growth of papers from the initial focus on manufacture and cost.

In the eighties there is a growing understanding of the role of product development as the context for emerging design and the changing role of the designer from a profession to a more integrated and total role as a team member. This development is mirrored in the paper classes shown in Figure 3.

From Engineering Design to Product Development	81	83	85	87	89	91	93	95	97	99	01
Product Development, concurrency, simultaneity, Life Cycle Design	5	3		4	7	5	18	15	18	17	9
Design Management, planning, strategy	1		2	5	1	6	9	4	6	2	9
Teams and Human Resources	1	1	2	2	3	1	7	6	5	12	20
Coordination and cooperation					2		2	2	6	8	5
Total	7	4	4	11	13	12	36	27	35	39	43
Relative to total number of papers %	8	3	3	9	13	5	13	8	9	10	13

Figure 3. The inclusion of Product Development topics to the ICED conferences.

This understanding of the context of engineering design confronts us with several neighbour research areas, bringing insights to the proper role and conditions of design and leading to a more total and mature understanding of designing.

The role of computers and information technology

As mentioned above I have "reduced" the number of papers on computers by allocating the papers to the design area, which are manipulated by the computer. Therefore the proceedings have many more papers allocated to the computer area than shown here.

Until '93 the use of CAD-systems for engineering design is presented in the papers, but later these papers have to do with more powerful systems for design and product development. Computers play just as important a role for handling knowledge, information and data (KID), and in the late nineties the number of papers on networking grow immensely, see Figure 4.

The role of computers and IT	81	83	85	87	89	91	93	95	97	99	01
Computers for engineering design and product development		14	8	13	1	19	28	25	20	23	8
Computers for KID	3	1	4	7	6	7	17	18	17	17	7
Networking								1	21	16	4
Product Modelling, Data Management							7	8	4	11	7
Knowledge Management										7	4
Modularisation, Configuration, Re-use, Platforms						3	7	10	14	20	21
Total	3	17	12	20	7	29	59	62	76	94	51
Relative to total number of papers %	3	15	10	16	7	2	22	18	21	24	16

Theory of Technical Systems		7	1	1		2	1	1	7	3	
Structuring, Laying out	2	2	10	2	1	9	12	6	1	1	1
Modelling		2	2	4	1	4	1	8	11	34	18

Figure 4. The change from CAD to broader IT application, and the theoretical areas supporting the handling of products as data.

One of the peculiarities and strengths of the ICED conferences is the emphasis on Theory of Technical Systems (TTS), i.e. the artefact theory, which creates the basis for several types of research. The balance of TTS papers on its own (see Figure 4) is not showing the areas' importance, but its role for the areas product modelling, data management, knowledge management, modularisation, configuration and product structuring should not be underestimated. Figure 4 shows the explosive growth of papers on some of the areas, mirroring their importance in industry. In the same way TTS plays a role in modelling area, where property modelling is put in focus.

The delimitations of engineering design research

How do the ICED conferences show the reflections on their own role, validity and power? As mentioned above, the scientific enhancement and research rigour has been in focus in the nineties. Also the societal role and importance and visions of the design area have been treated, see Figure 5.

Most contributions to ICED come from participants with an engineering background. Only a certain part of the design phenomena can be explained and understood from an engineering research viewpoint. Therefore it is crucial, as pointed out in numerous papers on design research and in the mentioned ICED pamphlet from 1997, that design research shall be arranged as multidisciplinary research. Some of the core areas include the understanding of the human designer, of cognition, creativity and attitudes, see Figure 5.

Modern industry's agenda is demanding dynamic change and a stream of competitive products, individually fitted to the customers. Only recently have the ICED conferences shown papers on innovation, understanding customers' needs, and studying change of the way design is organised, see Figure 5.

The delimitations of ICED	81	83	85	87	89	91	93	95	97	99	01
Societal role, importance and visions			2	2	2			2	3	2	
Cognition, Creativity			6	1		5	2	7	4	3	3
Reflective practise											5
Attitude, Ethics, Liability								2	1	4	1
Innovation, Invention							2	5	6	17	3
Needs, customers, competition						1			6	1	3
Study of change										3	2
Total			8	3	2	6	4	16	20	30	17
Relative to total number of papers %			7	2	2	3	1	5	6	8	5

Figure 5. Some of the topics on the borderline of the ICED conference topics.

5 Conclusion

What can we conclude concerning the ICED conferences' relation to industrial agendas? Let us try to identify a main ICED pattern. There seem to be four distinct time periods:

- A machine design, industrial practice, design profession period, where design methodology based upon experienced practice is the main source of insight. Very few are actually carrying out design research.
- A CAD-oriented period with emerging understanding of product development as context of engineering design and the design profession. Integration is carried by different DFXs, reduction of lead time is in focus.
- A period, where the focus turns away from the design to the management of the designing activity and the teams performing design. Product development is now seen as a new profession with focus on team competences. Design is studied with multiple views and empirical methods.
- Today's situation, where very many are performing design research, and a multifold of approaches and views are studied in what seems an explosive expansion. The role of scientific rigour and the necessity of multidisciplinary research are understood, but the theoretical foundation is not strengthened through the papers from this period. IT is setting the agenda for design practice. Efforts are done to understand the human operator.

Is this pattern of four time-periods of overlapping focuses, mirroring industrial agendas? I believe, that the industrial situation has changed radically throughout this period, showing the following steps:

- Product development was engineering oriented, CAD-based and managed by procedures in the early 80's.
- The focus was changing to market-orientation. Demand for dynamics led to integration and concurrency; team based design is evolving. This was the situation around 1990.
- Today, industry's agenda is dynamics and leanness, carried by IT-support, Product Data Management, and Product Modelling, and networking is emerging. Managers focus on human resource management.

The image of ICED conferences seem to be a mix of two things: "a market pull" i.e. interest in areas where needs are articulated (environment, DFX, teamwork, workbenches, data management etc.) and "a technology push", i.e. interest in areas emerging in industry (CAD, PDM, Configuration systems, Networking, etc.). It is only in the market pull area that ICED seems to be setting the agenda.

This analysis may hopefully serve as an inspiration for better understanding the role of ICED and what should be solved outside ICED by networking, discussion groups, workshops, research programmes and cooperation, especially with industry for understanding industry's agenda.

References

[1] The full list of proceedings from these conferences is printed in this proceeding.

[2] WDK introduce itself as follows:

WDK-Workshop Design-Konstruktion is an informally constituted international society, based on common interest in engineering design. Its main goal is to rationalize engineering design work through design science. Members of the WDk-Society are scientists, engineers and educators, who take an active part in the investigations, meetings and publications of the Society.

The WDK-Society is active in research on the design process and on technical systems; it organizes international co-operation in these fields through workshops and conferences (ICED). It is also active in distributing scientific results for rapid exchange of views, e.g. by publishing books and bibliographies (Series WDK) and by organizing seminars for students and practising engineers

[3] Papers for ICED. Information pamphlet to authors from the ICED '97 organisation, Tampere University 1997.

Professor, Tekn dr Mogens Myrup Andreasen
Section of Engineering Design and Product Development
Department of Mechanical Engineering, Technical University of Denmark
Building 358, Akademivej, DK-2800 Kgs. Lyngby
Phone: +45 4525 6258 • Fax: +45 4588 1451 • e-mail: myrup@mek.dtu.dk

Design Theory and Methodology

Including sub-sections:

Nature and Philosophy of Design
Design Theory
Design Theory and Methodology
Complexity
Complexity Management
Research Methodology
Reflective Practice
Models of Design Processes
Creativity and Innovation
Creativity and Innovation in Teams

This page intentionally left blank

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A CONTEMPORARY SURVEY OF SCIENTIFIC RESEARCH INTO ENGINEERING DESIGN

I Horváth

Keywords: Engineering design research, contextual reasoning model, research categories, research domains, research trajectories

1. Introduction

Engineering design is a distinguished discipline since it (i) synthesizes new information for product realization, (ii) establishes quality through defining functionality, materialization and appearance of artifacts, and (iii) influences the technological, economic and marketing aspects of production. By generating knowledge about design and for design, discipline-oriented (scientific) research is instrumental to the development of engineering design. By now, the notional research into engineering design has grown to a significant complexity. For this reason, it is not easy to see the trends of evolution, to identify the landmarks of development, to judge the scientific significance of the various approaches, and to decide on the target fields for investments. Orientation in the jungle of discipline-oriented research of design is difficult not only for specialized researchers, but also for research managers and science policy makers. On top of the managerial aspects, understanding the holism, coherence and robustness of the results, comprehension of the interactions of parts of the aggregated scientific/technological knowledge, and recognition of the status of the theories is important from a gnoseological point of view. What we need in this situation is a comprehensive but, at the same time, sufficiently articulated vision that reflect the current state-of-the-science and project ahead to the near future. Accordingly, the intent of the author in this paper is to collect and study the knowledge and facts about engineering design research. He has been inspired by the fact that these kinds of studies are less frequent than it would be needed.

As it can be seen in the literature, enormous efforts have been and are being made towards scientific understanding, technological underpinning and practical exploitation of engineering design. Research approaches design from several aspects such as governmental, industrial, historical, technological, educational, scientific, sociological, and practical. Researchers usually focus on a target application field (e.g., architectural, mechanical, electrical engineering) and narrow down their scope to problem areas such as conceptualization, detail design, computer support, and product realization. Thus, the number of scientific publications on technical issues of engineering design has been growing exponentially in the last four decades. At the same time, the number of papers addressing philosophical, epistemological and teleological issues of design research does not seem to grow so fast. Some authors have provided outstanding (systematic) surveys about the fields of attention and approaches of academic and/or company research [3], [5], [6] and [7]. In order to arrive at a systematized discussion, they introduced various reasoning schemes. The basis of these schemes ranges from a simple chronological principle, through a phenomenological classification. to a contextual taxonomization.

Rather than dealing with particular technical problems of engineering design, this paper intends to present a classified survey about the status of research that intends to create notional, or even scientific, foundation for engineering design. Like Archer [1] and Beheshti [2], the author applies a content-driven approach. He explores the boundaries of attention in engineering design research, classifies the domains based on their contents, and gives a concise summary on the recent advancement. The main difference between this survey and the other ones following taxonomic cataloging is that the author introduces a *reasoning model*. It enables him not only to define a domain of discourse, but also to explore intrinsic relationships.

2. The reasoning model

Engineering design research manifests as a platform for exploration, description, structuring, rationalization, and application of design knowledge and technologies, in combination with the designed artifacts and processes. This inherent complexity makes it inevitable to apply a multi-level, contextual structuring. Being aware of it, the author proposed a gnoseology-oriented approach [4]. It postulates a conceptual scheme that arranges (and explains) the universe of engineering design research. The fundamental observation is that the global discipline of engineering design is naturally rationalized and directed. The knowledge is transferred from scientific/theoretical inquiry and comprehension to technical/pragmatic application. Hence, the underpinning idea behind the conceptual scheme is the natural stream of knowledge through design. The analogy of the source, pipeline and sink of a material flow has been used to support the idea of a formal model. The advantage of this approach is that it covers the whole discipline of engineering design. The basis of demarcation of the fields of attention in research is the studied subject matter. The postulated scheme throws light upon the highest-level contextual categories, promotes a contextual decomposition (structuring), and brings the contextual relationships (interactions and dependencies) into the limelight. This formal conceptual model lends itself to an integrated view on the observable world of engineering design research and facilitates a comprehensive reasoning about it. For this reason, it has been referred to as a *reasoning model*.

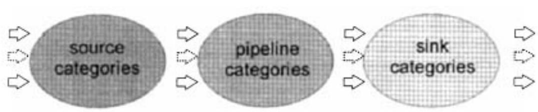


Figure 1 The natural stream of knowledge in engineering design

The conceptual scheme can be represented graphically as a chart of source-pipeline-sink categories, which can be particularized for contents on various levels (Figure 1). The *source categories* of engineering design research are the categories that endow with the fundamental mental capacity for engineering design. The *pipeline categories* establish links between scientific/theoretical knowledge categories and pragmatic/technical knowledge categories by structuring, deriving and dedicating knowledge. The *sink categories* are concerned with eliciting knowledge that is necessary for the ultimate utilization of the entirety of engineering design knowledge. As specific to engineering design research, the author inaugurated nine contextual *research categories* in the reasoning model. Within each contextual category, *research domains* are identified. In order to help distinguish particular research approaches and treatments of research issues in the research domains, *research trajectories* are introduced. They can be seen in Figure 2, which shows the contextual reasoning model so as it is decomposed to research domains level. In certain cases, the author could not find a better solution than to apply the same name to a contextual category and to a research domain that is fundamental to the concerned category. However, a name used to indicate a category has broader meaning than the one used as a marker of a domain. Due to space limitations, the design trajectories level decomposition of the reasoning model cannot be included and discussed in this paper.

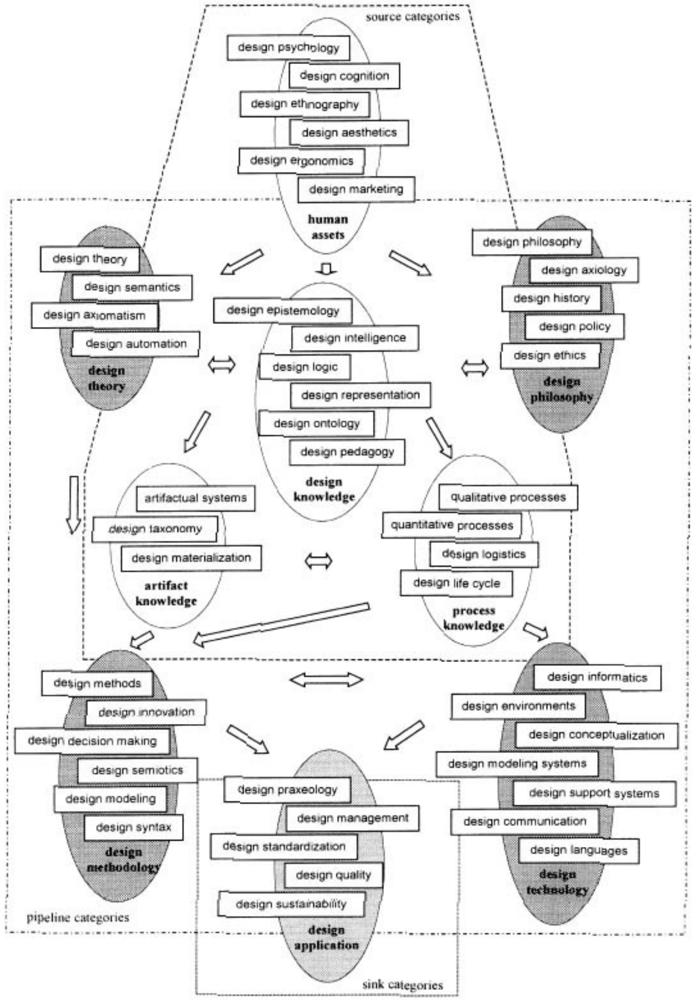


Figure 2 The gnoseology-oriented reasoning model showing the research domains in the contextual categories

3. Research in human assets

In spite of the fact that there have been efforts towards a kind of mechanization (computerization and/or automation) of engineering design, it remains one of the most human-related activities featuring intellectualism, creativity and ingenuity. The research in human assets focuses on the people who come into view as (i) *scholarly originators* of general and specific design knowledge, (ii) design *problem solvers*, and (iii) *target benefiters*. The goal is to study and understand the human aspects of engineering design, investigate all kinds of relationships of humans to design, enrich problem solving knowledge, and enhance the mental potentials. Within this category, *design psychology* research studies the humans who design, and who are affected by designs. In the former case, the mental processes and behavioral characteristics are investigated, in the latter case, the attitude towards designing and having designs are in the focus. *Design cognition* research has made its first steps towards the understanding of knowing, perceiving and conceiving design knowledge, and towards the comprehension of intuitions, hypotheses, feelings and beliefs regarding design. The research in *design ethnography* focuses on distillation of culturally relevant knowledge, and on the development of methodologies for an 'open to cultural differences' product development. *Design aesthetics* research achieved moderate progress with understanding emotional reactions of humans to aesthetic impressions caused by designs and the rules of creating designs with intended appearance through form giving, materialization and decoration. *Design ergonomics* has progressed in accumulating knowledge for optimizing the connection between clusters of humans and products/environments, especially from the aspects of physical and informational ergonomics. Supported by the general knowledge of marketing, *design-marketing* research progressed in understanding customer behavior, product policies, life cycle engineering, customization and globalization under the pressure of competition on product markets.

4. Research in design knowledge

Contemporary research in designing-related knowledge targets three generic issues, (i) the roots and the nature of engineering design knowledge, (ii) the contents and the characteristics of design knowledge, and (iii) the exploitation of design knowledge. Figure 2 shows the pertinent research domains. Having recognized the importance of an epistemological understanding of inquiring, generating, possessing and utilizing design knowledge by humans, researchers brought to existence *design epistemology* as early as the first part of the seventies. Knowledge of engineering design is fundamentally empirical in nature, although efforts have been made to theorize it by structuring, reasoning, abstraction and generalization, and mathematical processing. Descriptive formalisms are used for synthetic design knowledge, and prescriptive or predictive forms for the analytic part. *Design intelligence* research is interested mostly in (i) understanding human design intellect, (ii) study of abilities, creativity, intuition, thinking, common sense and formal reasoning, (iii) apprehension of specific problem solving capabilities, (iv) extension of design knowledge, and (v) reproduction of design knowledge in/by artificial systems. Also studied are (i) generation of synthetic knowledge by means of cognitive senses, and (ii) deriving analytic knowledge by mental reasoning. Closely related to these, *design logic* studies (i) the principles of common-sense, plausible and non-deterministic design reasoning, (ii) the foundations of design thinking, with the aim of deriving principles for procedural inference and criteria for design appreciation, and (iii) handling informational complexity, associations and constraints. *Design representation* research studies the alternative forms of externalization and epitomizing of general, artifact and process knowledge. A relatively new domain, *design ontology* research is directed to the conceptual understanding of all kinds of design 'worlds', providing very high levels of specifications, and finding extracting principles and structures for knowledge. In the future, ontology research may play an important role in implanting knowledge structures to, for instance, knowledge-intensive design systems in the form of very high-level languages. The main interest of *design pedagogy* research is to (i) provide an enabling 'cache' for engineering design, (ii) find up-to-date design education methods, (iii) develop new tutorial means and materials, and, (iv) train for engineering design research.

5. Research for knowledge about artifacts

Knowledge related to *artifacts* (also referred to as technical systems or products) represents a specific subset of design knowledge. Looking back to a long history, the research into artifacts intends to understand the rules, forms and relations of processing substance, energy and information in designs. In close relationship with the technical sciences, it studies physical, functional, morphological, structural, behavioral, realization and use aspects. The current theories understand the artifactual systems as goal-implied, synergetic arrangements of organs and place the emphasis on the laws of transformations, causality of changes, and optimization of operation. The aims of *design taxonomy* research are the teleological study of manifestation of purposeful artifacts in various categories and the exploration of general principles for orderly classifications of designs and their relationships. Having much in common with general system research, part of artifact research is interested in the holism of artifactual systems and their decomposition. *Design materialization* research deals with (i) the interrelation of materialization of artifacts and their manifestation, behavior, implementation and use, (ii) principles and laws of using various materials in artifacts, as well as with (iii) optimization of material use, (iii) ecological design and powering, and (iv) recycling issues.

6. Research for knowledge about processes

Knowledge related to *processes* is the other specific subset of design knowledge. Research concerns the artifacts/systems-related processes, involving (i) mental, virtual and physical creation processes, (ii) operational/behavioral processes, and (iii) processes accompanying the existence of products. It also covers processes of designing from aspects such as (i) defining the contents and organization of design processes, (ii) optimization of transformations in design processes, (iii) use of resources in design processes, and (iv) automation of design processes. Presently, research efforts are spread over both quantitative and qualitative processes. The research in *quantitative processes* to a large extent focuses on observable natural and artificial processes. On the other hand, *qualitative process* research concentrates on mental processes of design and qualitative reasoning with natural (physical) processes in artifacts/systems. An emerging research domain, *design logistics* is engaged with the issues of (i) distributing information in geographically distributed design environments, (ii) optimal organization of design processes, and (iii) managing complexity of design knowledge associated with artifacts and processes. Being a maturing domain, research in *life cycle processes* aims at observation, definition, modeling, and analyzes of product-related processes in a holistic way. It also looks for methods to incorporate life cycle (conceptualization, creation, behavior, use and recycling) knowledge into product models and products.

7. Research in design philosophy

Research in *design philosophy* tries to explain the nature and essence of engineering design based on logical arguments. It pursues a general understanding of design in a speculative, rather than in an observational way. By introducing general hypotheses, conjectures, concepts and ways of thinking about designs and/or designers, it affects the generic theory underlying design thinking. It investigates the scientific nature of design, the validity of design theories, the notions relating design. It also proposes teleological explanations of the relationships of humans and designing. Recently, emphasis was given to the strategic role and societal aspects of engineering design. *Design axiology* research is spontaneously developing to study the nature and the measures of the technical, economic, moral, social and aesthetic values created by design. *Design history* research focuses on (i) chronological developments of design knowledge and the subdisciplines, (ii) advancement of some philosophical/theoretical frameworks (paradigms), (iii) ontological, methodological and technological evolution of design, as well as (iv) political, social, cultural, and economic factors influencing the trends in designing and product development. *Design policy* research concerns the knowledge about (i) utilizing design on society level, (ii) embedding design in production environments, (iii) strategic issues and goals of design, (iv) outsourcing policies for design projects, and (v) planning collaborative design processes. Research in *design ethics* studies the ethical dimension in engineering design, including the man-made changes of nature, principles of a product to be useful for the society as well as the rules of designing considering all moral, social, political, cultural and personal aspects

8. Research in design theory

Research in design theory connects general and specific design knowledge to design methodology and practice. More specifically, it intends to (i) explain, generalize and/or abstract observed design processes, (ii) organize engineering design knowledge beyond the level of craftsmanship, (iii) develop formal design theories for an algebraic representation of designs and processes, (iv) introduce idealized models for the evolutionary design process, and (v) derive theorems, rules and procedures for solving design problems in synthetic environments. Global and local, empirical, descriptive and prescriptive theories have been considered. Research in *design semantics* targets meanings and intentions in design. Among the goals are (i) contextual understanding of the designing and designs, (ii) explicating design intents, (iii) clarification of functions and functional relationships, (iv) comprehension of shape perception and shape morphing, (v) grasping function to form transition, (vi) clarifying relationships of shape and behavior, and (vi) study of design evolution. *Design axiomatism* strives for developing formal reasoning frameworks from self-evident truths (axioms), propositions (conjectures) and/or facts (evidences). Various systems of axioms have been proposed to support global and local design theories. The domains discussed above interweave *design automation* research, which asserts that engineering design is a computable function. It studies computer-based problem solving strategies, methods, heuristics, creativity, learning, and reasoning. The ultimate aim is formal design inference, automated problem solving, and transplantation of design capabilities.

9. Research in design methodology

Current *design methodology* research is involved with (i) methodological systematization of design processes, (ii) exploration of the mechanisms of design decision-making, and (iii) improvement of design modeling, representation, analysis, simulation, evaluation, and/or physical testing techniques. Process monitoring and protocol studies are used to understand the human methods of designing, design knowledge requests and processing, collaborations, use of methods and tools, and design communications. *Design innovation* research creates a scientific basis for rationalizing multidisciplinary product development and facilitates solution finding for design problems. A booting-up research domain, *design decision-making*, investigates the cognitive and logical mechanisms behind finding solutions, selection of tools and methods, and weighted consideration of the factors. It also deals with the dependencies of design decisions on awareness, intents, situations, conditions, and constraints. *Design* *semiotics* studies the symbolism applied in the key functional activities in design, and in the related activities. Being an extremely wide area of interests, research in *design modeling* aims at generating, processing and using mental, cognitive, formal and symbolic models of humans, artifacts, processes and knowledge. It also investigates the role of models in externalization, communication and testing of design ideas. Closely related to design languages research, *design syntax* research specialized itself in the understanding of rules, structures and expressiveness of design grammars.

10. Research in design technology

In the broadest sense, design technology is associated with the use of (i) applied science and design-specific knowledge, (ii) human and informatics resources as well as of (iii) methods, techniques and means to solve well-defined design problems. Some 30 years ago, pulled by the industrial need and pushed by the rapidly evolving computer technology, *design technology* research showed an unexpectedly rapid progress. As indicated in Figure 2, it has developed into a well-articulated set of research domains. The two governing problems have been processing design knowledge by computers, and development of systems supporting design. Research in *design informatics* aims at studying all design-specific aspects of handling data and knowledge related to humans, products and tools. The premier issue has been processing visual and spatial information, which is enabled by the methods and techniques offered by interactive computer graphics and image processing research. Computer internal representation and processing of numerical, textual, symbolic, graphical and geometric information have become the key issue of research in *design environments*. The attention is put on computer processors, peripheral devices and networks that are dedicated to design tasks. It also endeavors to organize design and image catalogues in the context of design activities. Research in *computer-aided conceptualization systems* has become very active in the last two decades and it is still rapidly progressing. The efforts are to support (i) collecting and processing user needs, (ii) exteriorization of human concepts, and (iii) producing initial (possibly incomplete) models of products. The research in *design modeling systems* has focused both on simplified design representations and on true modeling of the geometry, structure, materialization, behavior, appearance and realization of products. For instance, a wide variety of multi-dimensional modeling techniques have been availed for representation of the shape such as wireframe, solid, surface, parametric and constraint-based, feature-based, deformable, fuzzy and vague geometric modeling. Being theoretically well supported, they lend themselves to the development of various types of so-called downstream modeling systems, such as analysis, manufacturing, assembly, and simulation systems. As far as behavioral modeling is concerned, noteworthy results are achieved with finite element modeling/analysis and kinematics simulation systems. Research in *design support systems* intends to discover new principles for product data management systems, design decision support systems, design knowledge- and databases, design optimization and design reengineering systems. The intentions to apply artificial intelligence techniques in design support systems have gradually weakened since the end of eighties, although this paradigm had seemed to be very popular earlier. The emphasis shifted to knowledge-intensive systems without built-in problem solving capabilities. Typical design support systems are the design for manufacturing and assembly systems, following specific methodologies. Research in *design communication systems* focuses on (i) video-conferencing-based collaborative techniques, (ii) network-based collection, organization, classification, transformation, visualization, retrieval and use of design knowledge, and (iii) network-based management of designing. Research in *design languages* targets formal product definition languages, and product description languages of neutral format.

11. Research in design application

Design application research extends to research domains, which make use of design theories, methodologies and technologies in solving concrete design problems. *Design praxeology* studies the nature, manifestation and instruments of design in the practice. Following the ever-increasing industrial expectations towards engineering design, research in *design management* investigates the methods of low-level organization of designing, exploitation of design tools for particular products, and creating efficient design environments. Significant steps have been made during the years in the domain of *design standards* research, which investigates the principles of standardization, normative regulations and qualitative aspects of measurement. Generating design codes, norms and standards is a domain with pronounced relationship to design technology. In the last ten to fifteen years, *design quality* research attained distinguished attention. It orientated itself to creation of reasoning models about quality, and pursued studying all factors that influence the resultant quality of artifacts and related processes, to support optimal execution of product development and production processes. Design *sustainability research* is involved in issues of (i) ecological design, (ii) (de)materialization and (de)powering of products, and (iii) reuse and recycling.

12. Conclusions

A gnoseology-oriented reasoning model is applied in this brief survey, which facilitates identification of contextual research categories, domains and trajectories, and analysis of the interrelations. Future research will focus on getting deeper insight into the definition, intrinsic characteristics, relationships, merits and advancement of the research trajectories.

References

[1] Archer, L. B., A View of the Nature of Design Research, in *Design: Science: Method*, ed. by Jacques, R., Powell, J. A., Westbury House, Guildford, 1981, pp. 30-35.

[2] Beheshti, R., van der Veer, P., Design Science and Technology: Setting the Agenda for the 21st Century, *EuropIA2000 Session Initiation Paper*, Civil Engineering Informatics, TU Delft, web-site: http://cci.ct.tudelft.nl/EIA2000, 1999.

[3] Finger, S., Dixon, J. R., A Review of Research in Mechanical Engineering, Part 1. *Research in Engineering Design*, 1989, Vol. 1, No. 1, pp. 51-68, and Part 2, Vol. 1, No. 2, pp. 121-137.

[4] Horváth, I., Vergeest, J. S. M., Engineering Design Research: Anno 2000, in *Proceedings of the International Design Conference - DESIGN 2000*, Dubrovnik, May 23 - 26, 2000, CD-ROM.

[5] Hubka, V., Eder, W. E., Design Science, Springer Verlag, London, 1996.

[6] Hsu, W., Woon, I. M., 1998, Current Research in the Conceptual Design of Mechanical Products, *Computer-Aided Design*, Vol. 30, No. 5, pp. 377-389.

[7] Pugh, S., Engineering Design - Unscrambling the Research Issues, *Research in Engineering Design*, Vol. 1, No. 1, 1990, pp. 65-72.

Corresponding author's name: Professor Dr Imre Horváth
Corresponding author's address: Faculty of Design, Engineering and Production
Delft University of Technology
Jaffalaan 9, 2628 BX Delft, The Netherlands
Phone: 00+31 15 278 3520, Fax: 00+31 15 278 1839
E-mail: i.horvath@io.tudelft.nl

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

GENETIC ANALOGIES IN DESIGN

T Bercsey, S Vajna, and P Mack

Keywords: evolutionary design, design activities, genetic behaviour

1 Introduction

The genetic analogies in design use the same model like the Darwinian evolution theory, i.e. the law of the "survival of the fittest". In [1] GERO showed the presence of these genetic analogies in design. The fundamental aspects of these analogies are:

- The design description maps onto the phenotype level. The phenotype describes the real existing product.
- Design processes map onto selected genetic operators (mainly crossover and mutation) at the genotype level. The genotype is the representation pattern created for the product.
- The difference of the representation at the genotype level from that of the design description (phenotype) level.
- Design performances (behaviours) map onto fitnesses, which describe the conformance of the product.
- Genetic operations are carried out with populations of individuals (design variants).

Our goal is to show that, based on these theoretical aspects, genetic analogies are present in the basic design activities of process and product modelling. Furthermore, we show examples and concepts of the adaptation and the use of these genetic analogies in order to increase design efficiency.

At first, the chosen design activities are analysed to point out the genetic analogies. The second chapter concentrates on the development of design variants and the evolutionary behaviour of these emerging objects. Afterward the analogies found on the product modelling level are discussed.

2 Process modelling

The design process can be regarded as a set of intuitive shifts between the different design phases. In each design phase, different alternatives "competing" with each other are compared. So the design process model must consider this "mixed" process of searching, adopting of existing knowledge, learning, evaluating, selecting, and combining [2]. These properties can also be found in the natural evolution process. That is why an evolutionary approach could be useful to understand, model and describe the design process.

By using obvious and hidden analogies between natural evolutionary processes and design processes the so-called Autogenetic Design Theory (ADT) has been developed [3, 4]. It is based on the idea that the design process can be described as an evolution of a specific nature in which new or adapted solutions (i.e. individuals) can be developed with the help of evolutionary methodologies as described in chapter 1. Different versions of a solution – which are a product of the symbiosis of evolution and human intuition – compete with each other. Under the pressure of selection (i.e. requirements and boundary conditions) "good" properties (i.e. properties which provide a better fulfilment of the requirements and conditions) of the preceding solutions (parents) are passed on to the succeeding solutions (children). Thus generation by generation the solution is subsequently improved, getting closer and closer to the given requirements and boundary conditions. This autogenetic behaviour can be observed in all variants of a solution as each emerging solution passes through a self-development process. This phenomenon repeatedly occurs within and between the individual design phases.

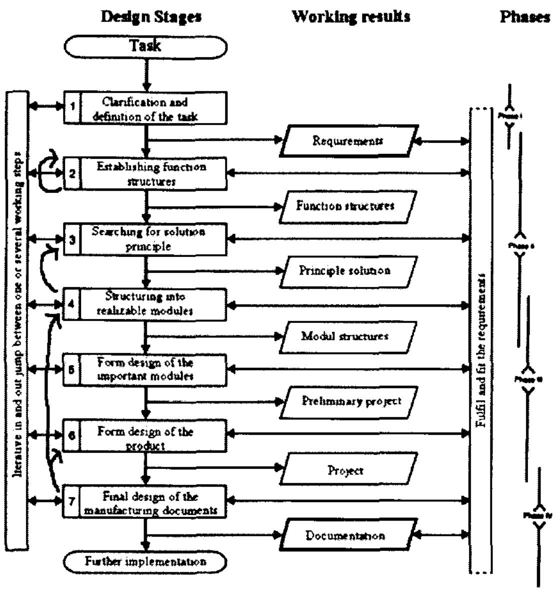


Figure 1. Structure of the VDI 2221 guideline [5]

Figure 1 shows the VDI 2221 guideline that realises a kind of autogenetic behaviour by offering the possibility of iteration cycles between the design phases. Although the possibility to fulfil this genetic analogy is present, the prescriptive layout of the methodology does not force the designer to follow the iteration cycles. The guideline emphasises a sequential process with only a few necessary iteration cycles. Nowadays these types of design approaches are integrated into company development processes as guidelines or standards, which is no doubt a great breakthrough in terms of managing the design process, but they are too abstract or too inflexible on the level of the daily work of most of the designers.

To present the power of ADT, we have a look at the working style conform to it and compare this with the aforementioned quasi-sequential methodology and their working style. Figure 2 shows the working style of two designers; both conform to the VDI 2221 guideline. The practical designer – more conform to the ADT – tests the variants on different abstraction levels in order to choose the most appropriate solution (black and white arrows). The more methodical, sequential working style tests the variants without making too many steps between the different abstraction levels. Naturally there are many influencing factors like design tools, education experience of the designer etc., which influences the efficiency of designing. In short, there seems to be a difference between the conventional design theories and the nature of the practical, real-life engineering design.

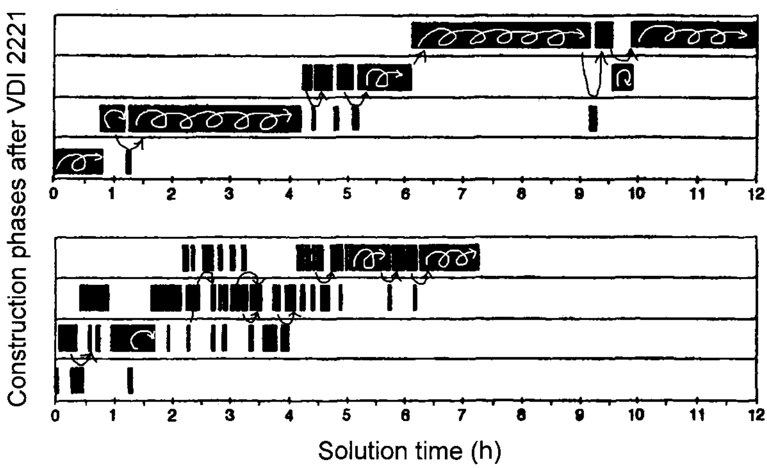


Figure 2. The difference between a "methodical" (upper) and "practical" (lower) designing styles [6]

After analysing GÜNTHER'S work [6], it could be established that a design process with an evolutionary-like layout is more efficient than the methodical quasi-linear layout. The evolutionary type of designing is not so transient like the conventional process but nowadays with e.g. efficient process management tools more complex processes can be monitored to avoid possible errors. VAJNA and WEBER showed in [7] that most of the tools are available today that are already able to manage successfully such a "sequenceless" procedure, even in the early stages of design.

In addition to pointing out the evolutionary analogies we must also point out the significant difference between natural evolution and the ADT. Both are directed processes towards an objective, but designing is a consciously directed process while natural evolutions is an unconsciously directed process.

The tool of a designer for consciously direct the design process is the intuition. By dialectically solving conflicts between product requirements and real product behaviour - which are influenced by design characteristics - the designer creates new design variants which hopefully better fulfil the requirements.

The natural evolution is directed by the fact, that the fittest solution has more chance to survive the evolutionary design process. Natural evolution uses random design changes without any rational directing. On figure 3 are the different working styles (conventional, evolutionary driven and combined) are demonstrated.

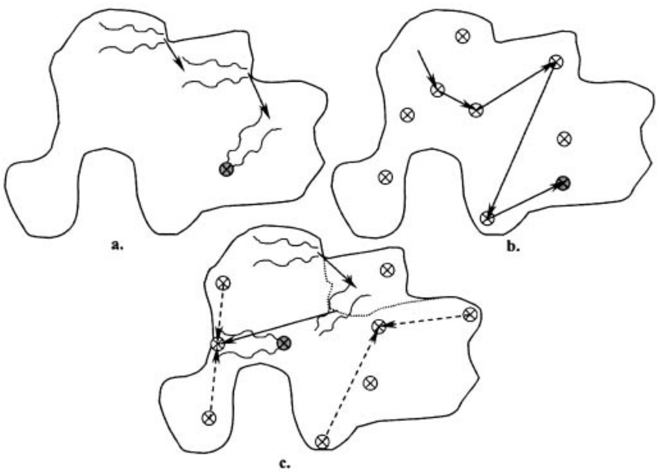


Figure 3. Conventional (a.), evolutionary(b.) and combined working styles (c.)

The conventional working style (figure 3/a) tests many design variants, continuously enlarging and narrowing the actual search space, restricted by the requirements. The evolutionary design style (figure 3/b) creates random variants by taking the best overall variant. A possible combined working style according to the ADT is demonstrated on figure 3/c. The designer dialectically creates variants invoking "wild" variants (random points) created by random changes and combinations of those wild variants (chopped arrows). This could be useful, to widen the search space and to find new alternative variants in the case of changing or new boundary conditions.

Within the ADT the evolutionary approach is realised by evolutionary algorithms, which simulate the natural evolution process by creating artificial populations using mathematical expressions for replication, crossover and mutation. A population consists of versions of a solution competing with each other.

A predefined fitness function (containing the evolution goal) is used to assign a quality value (fitness value) to each individual of the population. This value in conjunction with a fixed population size generates a selection pressure such that "improved" solutions get a greater chance to reproduce themselves. In this way, new generations of populations develop continuously.

With a search and optimisation tool described in the next chapter, we realised an autogenetic designing system conform to the ADT for the later phases of the design process. The future goal is to combine the ADT with modern process management tools in order to enable the utilisation of the genetic analogies in all design phases. In this way we can unite the most powerful designing techniques with a complete process transparency and control.

3 Product modelling

In the second chapter we showed that the product development process is analogous with the evolutionary process. This analogy can also be found on the product modelling level. Genetic analogies in product modelling are formulated, related to the phenotype (which maps to the object itself) and to the genotype level (which maps to the characteristics of the object).

3.1 Problem formulation

To set up the development of a new product the followings are required:

- The formulation of the problem,
- The search target and the appropriate requirements.

The difficulty when formulating the problem is to establish the connection between the real existing product (phenotype) and the representation pattern created for it (genotype). After this set-up, a search-optimisation is performed in order to find solution variants related to the problem. Firstly, with a search process the characteristic features of the problem must be found, and possible solution spaces should be identified. At next the formulation of possible parameters and the restriction of the search spaces, an optimisation process is carried out. After the optimisation is done, new requirements could be invoked or alternative new parameters could be integrated. This cyclic process shows a typical autogenetic behaviour; it evolves new solutions, combining search and optimisation processes.

3.2 Solution technique

To choose a proper search-optimisation strategy or method, the following significant factors must be considered:

- exploration,
- exploitation and
- universality.

Like the ADT, evolutionary algorithms could be used as search and optimisation technologies for product modelling. There are existing solutions using evolutionary algorithms as search and optimisation tools, but almost all of them are isolated solutions to special sub-problems, like special optimisation applications.

Fitting the solution technique to the problem formulation, the fact must be noticed that

- a too deep adaptation to the problem disturbs the universality of the system, i.e. it cannot be applied for other design problems,
- but a too rough adaptation is likely to be ineffective in search.

Evolutionary algorithms are in balance between exploitation methods like Hill-Climbing and exploration methods like Monte-Carlo. They "learn" by stochastically searching in the design space. In order to achieve quick and viable results, the separation of the phenotype and genotype level is very important. In this way the universality of the evolutionary optimisation system combined with the power of the natural evolution process can be ensured.

Based on the above-mentioned points, an optimisation system was developed using evolutionary algorithms with a modular build-up, as shown in figure 4. This system facilitates intuitive variation of geometrical product variants of growing complexity during their development.

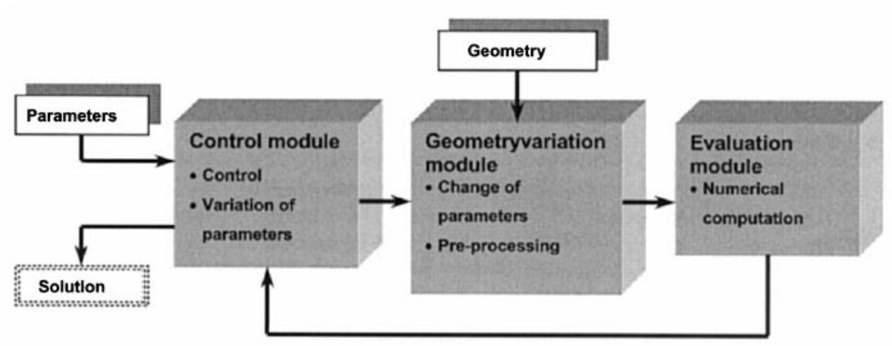


Figure 4. Modular search and optimisation system

3.3 Practical applications

The system was successfully applied to different types of geometrical design and optimisation problems like:

- Spiral spring design (3 optimisation parameters): the helix diameter, the wire diameter, and the spring length were optimised in compliance with the linear spring characteristics.
- Joint mechanism (12 parameters): A general four-joint mechanism was developed for a given trajectory.
- Inlet ports of a passenger car engine (23 parameters): On the basis of a geometrical model of the port channel, a geometrical parameter optimisation was performed using computational fluid dynamics as an evaluation tool [11]. Figure 5 shows the geometrical model of the optimised inlet port.

A big advantage of the genetic approach is the dynamical handling of the requirements. Thus, if the solution advances in time and possible new conclusions about the problem can be made, new requirements can be included in the evaluation procedure.

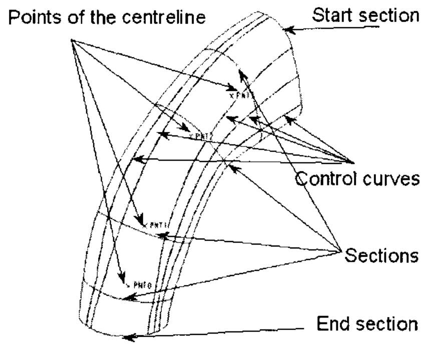


Figure 5. Geometric optimisation model of an inlet port

In the practical application, the inlet port was at first optimised for throughput and tumble scale. After having found an appropriate solution for these starting requirements, new aspects like fuel injection, emission, and temperature conditions can be added into the list of requirements. This means that with the change of the requirements on the phenotype level, the genotype level can be changed dynamically, i.e. new evolution methods and other possible non-geometric parameters can be included.

Further developments of this system (which realises a continuous monitoring of the phenotype level) will allow

- to search for the dominant genes (parameters) beside the variants, i.e. to look for the existence of a super scheme, and
- the semi-automatic generation of conclusions on the basis of the super scheme.

4 Conclusions

Genetic analogies can be found on the design process and on the product modelling level. To raise the effectivity of processes and methods, methodical and practical examples where demonstrated to each topic area. By an extended and improved use of genetic analogies and behaviour in design, new possibilities open up which offer quality improvement for both the design activities and the resulting products.

We also admit that the change to the practical use of genetic analogies is not always economic if one focuses on one special design problem only, but the situation is different considering the whole product life cycle. The employment of a new technique always drags some realisation problems with itself. But for complex non-linear systems with changing boundary conditions, the genetic analogies always provide good solutions.

References

[1] Gero, J., in Parmee, I.C. (editor): "Adaptive Computing in Design and Manufacture", Springer, Berlin New York, 1998

[2] Vajna, S., Bercsey, T., Clement, S., Mack, P.: "Autogenetic Design Theory - A Contribution To An Extended Design Theory", Proceedings of the ASME 2000 Design Engineering Technical Conferences, Baltimore MD, 2000

[3] Bercsey, T., Vajna, S: "Ein autogenetischer Ansatz für die Konstruktionstheorie I.-II.", CAD/CAM-Report 13(1994)2, CAD/CAM-Report 13(1994)3

[4] Wegner, B.: "Autogenetische Konstruktionstheorie - Ein Beitrag für eine erweiterte Konstruktionstheorie auf der Basis Evolutionärer Algorithmen", Thesis, Otto-von-Guericke Universität Magdeburg, 1999

[5] Verein Deutscher Ingenieure: "Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte", Richtlinie VDI 2221, VDI-Verlag, Düsseldorf 1993

[6] Günther, J: "Individuelle Einfluesse auf den Konstruktionsprozess", Shaker Verlag, Aachen, 1998

[7] Vajna, S., Weber, Ch.: "Sequenzarme Konstruktion mit Teilmodellen – Ein Beitrag zur Evolution des Konstruktionsprozesses", Konstruktion 52(2000)5

[8] Nachtigall, W.: "Bionik – Grundlagen und Beispiele für Ingenieure und Naturwissenschaftler", Springer, Berlin - New York, 1998

[9] Goldberg, D.E.: "Genetic Algorithms in search, optimisation, and machine learning", Addison-Wesley Pub Co., 1989

[10] Vajna, S., Freisleben, D., Scheibler, M.: "Knowledge-based Engineering Process Model", in: Riitahuhta, A. (editor): Proceedings of ICED 97 Tampere, Vol. 2

[11] Mack, P., Wegner, B., Bercsey, T., Vajna, S., Kickermann, H., Kellner, P.: "Computer Aided Optimisation of Cylinder Head Inlet Ports Using Genetic Algorithms" in: Lindemann, U., Birkhofer, H., Meerkamm, H., Vajna, S. (editors): Proceedings of ICED 99, Vol. 2

Univ.-Prof. Dr.-Ing. Tibor Bercsey
Institute for Machine Design
Technical and Economical University of Budapest
H-1111 Budapest, Müegyetem rkp. 3.
Tel./Fax.: +36-1463-4081 / +36-1463-3505
bercsey@xenia.gee.bme.hu
http://www.gszi.bme.hu

Univ.-Prof. Dr.-Ing. Sándor Vajna, Dipl.-Ing. Péter Mack
Computer Applications in Mechanical Engineering
Otto-von-Guericke University
Universitätsplatz 2, P.O. Box 4120, D - 39106 Magdeburg
Tel./Fax.: +49-391-67–18793 / +49-391-67-11167
vajna@mb.uni-magdeburg.de, peter.mack@mb.uni-magdeburg.de
http://imk.uni-magdeburg.de/LMI/index.htm

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DESIGN RESEARCH IN PERSPECTIVE – A META-RESEARCH ON ICED 97 AND ICED 99

M Cantamessa

Keywords: Survey, Technological trajectories, Introduction of methods in industry

1 Introduction

If compared to other branches of engineering, engineering design research sets itself apart from a number of perspectives. Among these, one can mention its relative youth, the involvement of researchers with different disciplinary backgrounds and the fact that there is no specific field of the natural sciences of which it can be viewed as a natural offspring, and from which research methods and tools have been inherited. It is therefore no simple matter to define the contents, the research approach or the community behind research in engineering design. Gaining insight upon such matters is useful for the "governance" of research, be it in organizing events, in refereeing papers, or in supervising the work performed by students. This paper aims to provide a first step in this direction by proposing a quantitative analysis of the papers contributed to the two previous editions of the ICED conference.

2 Background

Literature provides a number of attempts to defining the field of design research. Design and science have traditionally been viewed as separate fields, with the latter producing knowledge upon nature and the former using such knowledge to perform actions upon nature itself. Such a perspective is close to the Aristotelian distinction between "episteme" producing "theoria" and "techne" aiming to "poiesis" (i.e., producing new things). Modern students of technology [1-2] showed that, even if technology is indeed related to science, technological knowledge is something different and richer than simply "applied scientific knowledge". Other authors, such Simon [3], suggested the development of a "science of design" as "a body of... analytic, partly formalizable, partly empirical, teachable doctrine about the design process". Simon considered design broadly, as a process defining a course of action "aimed at changing existing situations into preferred ones". During the '80s and '90s, prominent figures of the design research community have tried to relate design to science in a more complete manner. Hubka and Eder [4] viewed "design science" as a comprehensive body of knowledge which includes four underlying categories (i) theory of technical systems, (ii) design theory and theory of design processes, (iii) special technical information and applied knowledge from natural and human sciences, (iv) design methodology. Cross [5] distinguished among (i) "scientific design" (i.e., when design is a subject that uses scientific knowledge), (ii) the "science of design" (i.e., when design is viewed as a phenomenon and is a passive object of scientific analysis), (iii) "design science" (i.e., when one makes design happen in a scientific way through methods and tools, and design is an object to which scientific knowledge is applied). Finally, Finger and Dixon [6,7] survey the many streams of design research.

The panorama offered by design research, as it appears in scientific journals or in conferences such as ICED, or as described by the previously mentioned taxonomies, is therefore extremely varied. A number of factors suggest that such variety is bound to increase in the future. In first instance, there is a trend towards a broader concept of "design" and "technology". While design research has initially been quite close to the area of mechanical engineering, it has progressively widened its attention to products based upon heterogeneous technology, including software. The design of products is now being integrated with that of systems and services (e.g., one priorities of the Growth research Workprogramme of the European Union consists in the development of "product-services"). Second, there is a growing number of design researchers coming from disciplines different from engineering, from management to operations research and from the different streams of Information Technology to psychology. All such variety is not an evil in itself, of course. When dealing with the design of heterogeneous artifacts, researchers are forced to focus upon the more abstract themes of design that are common to different application areas, therefore cutting loose from field-specific aspects. At the same time, the various disciplines enrich the research field by providing multiple perspectives and methodological approaches. So, no one would like design research to be a sort of a monolith. On the other side it must be remarked that, while variety has the potential of delivering value, this is not a certainty . If left to itself, there is a risk that research may end up in a set of unconnected streams and in a sort of methodological anarchy where anyone can come along and claim the scientific validity of his work.

In Kuhnian terms, the objective of this paper is to investigate the extent to which design research may be considered to be an identifiable research paradigm. In [9], a paradigm is broadly defined as the set of research problems, background theory and research methods that become widely accepted within a scientific community in between scientific revolutions. The power of the paradigm is that, by conforming to it, researchers don't have to "go back to fundamentals" every time and can build upon each others' work, thus leading to a rapid sequence of incremental improvements. The process goes on until the limits inherent to the paradigm's theory and methods make it incapable of solving new problems. At this point, the old paradigm is confronted with new competing and "revolutionary" ones, based upon different theory and methodology, and the one that best manages to solve research problems eventually wins. From this perspective, design research may be seen as a relatively young discipline that has indeed emerged out of revolutionary visions and proposals. What should be studied is therefore the extent to which it has moved into a paradigm or whether it still is in a pre-paradigmatic stage. Instead of the usual historical methods, this paper proposes an empirical approach and uses the two past editions of ICED (ICED97 and ICED99) as "cross sections" of the research work being performed by the community.

3 Research approach

The research papers in the ICED proceedings offer a wide variety of objectives, which makes them rather difficult to compare, especially because of the different criteria used at the two events for creating presentation streams. A new classification has therefore been developed, based upon the objectives of research with respect to the design process. Five groups have been formed (i) empirical research (ES in the following), (ii) experimental research (EX), (iii) development of new tools and methods (NT), (iv) implementation studies (IS) and (v) other (OTH). The rationale of such classification is that the first four categories, which cover more than 80% of the papers, may easily be related to one another, as shown in Fig. 1. This helps to stress the idea that, if the design research community executes different forms of research, this should be in view of exploiting their complementary nature. In Fig. 1, the world is split into the realm of industry and that of research. The "end product" is assumed to be the industrial implementation of tools and methods within the design process. Such tools may be derived directly from research efforts or, indirectly, through commercial development. The development of methods and tools depends on knowledge upon the design process and this, in turn, comes from experimentation or empirical observation.

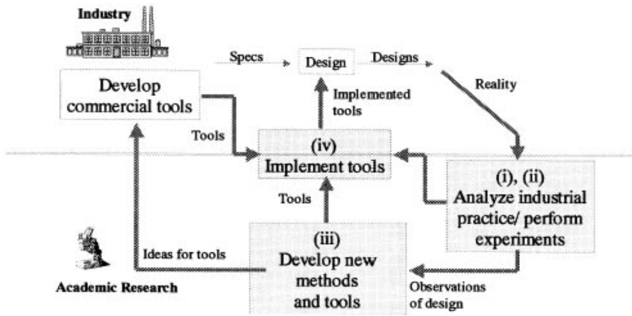


Figure 1. A model of research approaches in engineering design

Data collected from the papers presented at the two editions of ICED include descriptive variables upon authors, their nationality and affiliation, research objectives and methodology, references to previous literature, industrial involvement and the degree to which research is related to other complementary branches of research. Some of these variables are judgmental and related values have been determined by a panel of graduate students operating with a sort of Delphi procedure. When evaluations were unanimous they were not discussed. When conclusions differed they were subject to discussion and, if a consensus could not be achieved, the author would act as referee. This procedure does not totally exclude subjective bias, but it gives confidence that measurement error, though present, is reasonably reduced.

4 Findings

4.1 Descriptive statistics

At both ICED97 and ICED99, the development of new methods and tools appeared to be the dominant research theme. Table 1 shows that little less than half the papers were related to this kind of research, with little change across the two events and no significant effect by presentation type (oral vs. poster).

Table 1. Distribution of papers by research type

	ICED97			ICED99		
	Oral	Poster	Total	Oral	Poster	Total
IS	24	32	56	25	26	51
EX	8	12	20	17	5	22
EM	37	16	53	32	26	58
NT	71	70	141	80	110	190
OTH	22	37	59	37	31	68

Concerning the affiliation of researchers, there is some degree of concentration. As fig. 1 shows, the largest 20 % of institutions produced 60 % of papers. A weight index for research centers has been developed, in which oral papers weight double the amount of poster papers. By observing the histogram of this index, research centers have been classified in small (index < 24.8), medium (24.9 < index < 49.6) and large (index > 49.7). Table 2 shows paper distribution by geography. The fact that ICED97 took place in Finland and ICED99 in Germany causes some concentration, but anyway at a reasonable level.

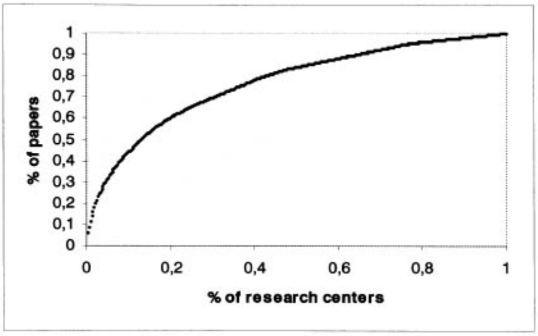


Figure 1. Pareto analysis of research paper production by research centers.

Table 2. Distribution of papers by geographical regions

	D, A, CH	UK, IRL	S, FIN, N, DK	USA, BR	Other
Oral papers	175	129	60	65	114
Poster papers	106	91	67	56	183

Research type shows a statistically significant degree of association with authors' affiliation and geographic origin and, to a minor extent (Table 4). Such evidence isn't particularly revealing, but it does suggest that, to some degree, determinants of "specialization" do exists.

Table 3. Association between research type, research center size and region (figures within parentheses give the number of papers that would be expected if the two categorical variables were statistically independent)

	Weight of research center (G-K Tau p<0.032 w/ research type dependent)			Region (G-K Tau p<0.001 w/ research type dependent)					
	Small	Medium	Large	D, A, CH	UK, IRL	N, DK, FIN, S	USA, BR	Other	Cross region
IS	97 (90.3)	13 (16.8)	3 (5.9)	34 (26.9)	27 (22.7)	12 (12.7)	8 (10.6)	26 (33.1)	6 (7.0)
EX	29 (34.4)	12 (6.4)	2 (2.2)	9 (10.2)	8 (8.6)	2 (4.8)	9 (4.0)	11 (12.6)	4 (2.6)
EM	84 (92.7)	26 (17.2)	6 (6)	15 (27.4)	26 (23.1)	23 (12.9)	6 (10.8)	31 (33.7)	14 (7.1)
NT	274 (272.6)	44 (50.6)	23 (17.8)	92 (81.3)	66 (68.5)	31 (38.3)	36 (32.0)	101 (100.0)	15 (21)
OTH	114 (107.9)	16 (20.0)	5 (7.0)	28 (32.2)	23 (27.1)	16 (15.2)	11 (12.7)	50 (39.6)	7 (8.3)

4.2 Use of previous literature

In science, it is customary for researchers to build upon colleagues' results. Giving reference to previous work is the means with which this relationship is signaled to the community. It is therefore quite common to measure referencing patterns and to use such figures for different applications. Specifically, references may be used as a metric for evaluating the degree with which researchers feel their current project to be related to their own previous work or to that of others. This may be seen as a proxy of the cohesion present in the community, which is an important feature of a research paradigm. ICED papers have been checked for references made to previous work by the same author and for the references made to papers appeared in previous editions of ICED (Table 4). The figures have been compared with the corresponding data obtained from conferences from other fields (three are rather specialist branches of engineering, while a fourth comes from an operations management background and is very close to the domain of ICED). The referencing pattern exhibited by papers in the ICED proceedings is similar to what can be found in work presented by other research communities.

Table 4. Use of previous literature in ICED and other conferences (OMAE = 10th Int. Conf. on Offshore Mech. and Arctic Eng., IEPG = 6th Int. IEE Conf. on "Opportunities and Advances in Int. Electric Power Generation", PCSM = 4th Int. IEE Conf. on Power Control Systems and Mgmt, PDM = 10th EIASM Product Development Mgmt Conf.)

		ICED	OMAE	IEPG	PCSM	PDM
Ref. made to papers appeared in previous editions of same conference	None	68.3 %	45.6 %	65 %	77.8 %	74 %
	At least one	31.7 %	54.4 %	35 %	22.2 %	26 %
Ref. made to previous work by author(s)	None	47.2 %	37.5 %	55 %	59.2 %	51.9 %
	Appeared in other journals or conferences	35.1 %	34.6 %	30 %	27.8 %	33.8 %
	Appeared in previous editions of the same conference	17.7 %	27.9 %	15 %	13 %	14.3 %

Finally, tighter referencing patterns are associated with research type and with authors' affiliation (tables 5-7). This can be viewed as preliminary evidence that shows the existence of different writing "traditions" or habits across regions or disciplinary areas.

Table 5. Association between use of previous literature and research type (figures within parentheses give the number of papers that would be expected if the two categorical variables were statistically independent)

		IS	ER	EX	NT	OTH
Ref. made to previous work by author(s) (G-K Tau $p < 0.034$ w/ reference dependent)	None	46 (52.6)	64 (54.0)	19 (20.0)	143 (158.6)	76 (62.8)
	In other journals or conferences	45 (40.2)	35 (41.3)	16 (15.3)	132 (121.3)	38 (48)
	In previous editions of the same conference	22 (20.2)	17 (20.8)	8 (7.7)	66 (61.1)	21 (24.2)

Table 6. Association between use of previous literature and research center size (figures within parentheses give the number of papers that would be expected if the two categorical variables were statistically independent)

		Small	Medium	Large
Ref. made to papers appeared in previous editions of ICED (G-K Tau p < 0.061 w/ reference dependent)	None	415 (409.3)	77 (76)	20 (26.7)
	At least one	183 (188.7)	34 (35)	19 (12.3)

Table 7. Association between use of previous literature and region (figures within parentheses give the number of papers that would be expected if the two categorical variables were statistically independent)

		D, A, CH	UK, IRL	S, N, FIN, DK	USA, BR	Other	Cross region
Ref. made to papers appeared in previous editions of ICED (G-K Tau p < 0.003 w/ reference dependent)	None	141 (121.8)	86 (102.6)	56 (57.5)	47 (47.9)	149 (149.8)	32 (31.5)
	At least one	37 (56.2)	64 (47.4)	28 (26.5)	23 (22.1)	70 (69.2)	14 (14.5)
Ref. made to previous work by author(s) (G-K Tau p < 0.006 w/ reference dependent)	None	94 (82.9)	61 (69.9)	52 (39.1)	30 (32.6)	97 (102.0)	14 (21.4)
	In other journals or conferences	60 (63.1)	60 (53.2)	18 (29.8)	27 (24.8)	79 (77.7)	21 (16.3)
	In previous editions of ICED	24 (31.9)	29 (26.9)	14 (15.1)	13 (12.6)	43 (39.3)	11 (8.3)

4.3 Issues specific to the four types of research

Empirical research

In ICED papers, empirical research is performed through case studies (46.9 % of papers) and surveys (43.5 % of papers), while in 9.6 % of cases the research approach is not clear. The unit of analysis, when declared, may be the firm, the project or the designer (Table 8). The sample size is declared only in 59 % of papers. Implications of findings, which are assumed to be useful to researchers involved in the development of new design methods and tools, are present in 74.6 % of cases, and absent in the remaining 25.4 %. This can be contrasted with the PDM conference, which has a firmer tradition in empirical research. In the PDM conference the research approach is declared in all papers and sample size in 83 % of them. While sample size, when declared, is not substantially different from ICED, managerial implications are present in 85.9 % of papers.

Table 8. Distribution of empirical research according to research approach and unit of analysis

	Firm	Project	Individual designer	Not clear	Average sample size	Average sample size in PDM
Case studies	19	21	8	6	15.6	50.6
Survey	15	9	13	11	150.6	112.6
Not clear	0	1	3	9	-	-

Experimental research

Experimental research is a numerically small part of research presented at ICED conferences. Units of analysis may be the individual designer (39.55 % of cases), groups of designers (55.8 %) or customers (4.65 %). Conclusions and implications are present in 81.4 % of cases and absent in the remaining 18.6 %.

Development of new tools and methods

It is possible to classify the work performed within this research stream in new methods (45.5 %), software tools (35.5 %) or methods embedded in software (19 %). Authors define precise motivations for their research, eventually through references to previous research, in only 32.6 % of instances. Such motivations are vaguely defined in 20.2 % of cases and absent in 47.2 % of cases. Papers dealing with the development of new methods and tools generally pay little attention to implementation issues within an industrial setting (37.5 % of papers address the topic but 62.5 % do not). They seldom relate the tool to the current state of the art in commercially available solutions (such relationship is discussed in 32.1 % of cases and neglected in 67.9 %), and they generally do not discuss the issues related to integrating the new method or tool within the current commercial offering (40 % do, while 60 % do not). Not surprisingly, the degree of industrial involvement is associated to a statistically significant degree with the above mentioned issues (table 9).

Table 9. Association between industrial involvement and aspects of research in the development of new tools (figures within parentheses give the number of papers that would be expected if the two categorical variables were statistically independent)

		None	Present	One industrial author
Implementation issues (G-K Tau $p < 0.008$ w/ implementation issues dependent)	Neglected	161 (148.7)	24 (30.5)	21 (26.8)
	Considered	78 (90.3)	25 (18.5)	22 (16.2)
Relationship with commercial state of the art (G-K Tau $p < 0.019$ w/ state of the art dependent)	Neglected	93 (89.8)	15 (14.1)	14 (18.1)
	Considered	41 (44.2)	6 (6.9)	13 (8.9)
Issues related to integration within commercial state of the art (G-K Tau $p < 0.063$ w/ integration dependent)	Neglected	82 (76.4)	12 (12.1)	10 (15.5)
	Considered	51 (56.6)	9 (8.9)	17 (11.5)
Industrial needs for new tool or method (G-K Tau $p < 0.005$ w/ industrial needs dependent)	Not specified	116 (111.1)	24 (23.1)	14 (19.8)
	Vaguely specified	55 (49.8)	10 (10.3)	4 (8.9)
	Well identified	70 (80.1)	16 (16.6)	25 (14.3)

Implementation studies

Implementation studies vary widely as far as generality of results is concerned. 21.2 % of such papers contain a generic discussion upon the topic. In 36.3 % of cases they report upon a single implementation project. Only in 14.2 % of cases one can find an attempt to generalize the findings, and in 28.3 % to develop a methodology. Implementation studies are rather evenly split as far as the object of the study is concerned (46 % deal with methods and 54 % with software tools). Reference to relevant empirical research is made by 45.1 % of papers.

5 Conclusions

The conclusions that can be drawn from the evidence thus gathered are "impressionistic" and certainly not conclusive. From a methodological perspective it should be noted that this paper has had the objective of discussing patterns, if present, and has not followed the more rigorous approach of formulating hypothesis and then using data to validate or refute them.

Generally, speaking, research presented at ICED conferences appears to be rather well-founded. Comparisons with other engineering conferences and the stability shown over the two editions of the conference show that, in general terms at least, the identity of design research is by now established. At the same time, a number of weak spots emerge from data, even though this is a typical situation in which one could debate whether the glass is "half empty" or "half full". One problem which is quite apparent is that the different streams of research are rather loosely coupled. Authors are probably not accustomed to studying and referring to the literature from other streams of research that could help them in directing their efforts more effectively. Particularly significant is the relative "isolation" with which authors engaged in the development of new tools appear to work with respect to industrial needs, to the current state of the art, and to implementation issues. In such cases, industrial involvement appears to be beneficial, and should probably be actively backed in order to ensure a closer connection to reality. Other weak aspects occasionally emerge concerning research methodology within individual streams of research, as in the tendency not to declare the research approach explicitly in empirical projects. This fundamental solidity, coupled with a few weak aspects should not be seen as a negative fact but, rather, as the sign that design research still has room for improvement.

References

[1] Vincente, T., "What Engineers know and how they know it", John Hopkins University Press, Baltimore, 1990.

[2] Staudenmaier, J.M., "Technology's Storytellers : Reweaving the Human Fabric", MIT Press, Cambridge, MA, 1989.

[3] Simon, H., "The Sciences of the Artificial", MIT press, Cambridge, MA, 1996.

[4] Hubka, V. and Eder, W.E., "Design Science", Springer-Verlag, London, 1996.

[5] Cross, N., "Science and Design Methodology: a Review", Research in Engineering Design, 5, 1993, pp.63-69.

[6] Finger, S. and Dixon, J.R., "A Review of Research in Mechanical Engineering Design, Part I: Descriptive, Prescriptive, and Computer-Based Models of Design Processes". Research in Engineering Design, 1, 1989, pp.51-68.

[7] Finger, S. and Dixon, J.R., "A Review of Research in Mechanical Engineering Design, Part II: Representation, Analysis, and Design for the Life Cycle" Research in Engineering Design, 1, 1989, pp. 121-137.

[8] Kuhn, T., The structure of scientific revolutions, Chicago Univ. Press, Chicago, 1962.

Marco Cantamessa
Politecnico di Torino, Dipartimento di Sistemi di Produzione ed Economia dell'Azienda, Corso Duca degli Abruzzi 24, 10129 Italy, Tel: +39-0115647238, Fax: +39-0115647299, E-mail cantames@athena.polito.it

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

CURIOSITY-ORIENTED RESEACH IN ENGINEERING [CORE] (DESIGN)

A Samuel and W Lewis

Keywords: Research, science, hypotheses, conceptual frameworks, research strategy

1 Introduction

In his book on "Sciences of the Artificial"[1], Herbert Simon notes:

"Historically and traditionally, it has been the task of the science disciplines to teach about natural things: how they are and how they work. It has been the task of engineering schools to teach about artificial things: how to make artifacts that have desired properties and how to design."

In this chapter on *"The Science of Design"* Simon deplores the way in which teaching institutions progressively moved away from the study of sciences of the artificial (design) towards the natural sciences. Simon also notes that while the natural sciences were perceived to be *"intellectually tough, analytic, formalizable and teachable"*, while design was perceived as *"intellectually soft, intuitive, informal, and cookbooky".*

Simon then goes on to define the core activity of design as:

"The artificial world is centred precisely on this interface between the inner and outer environments; it is concerned with attaining goals by adapting the former to the latter. The proper study of those who are concerned with the artificial is the way in which that adaptation of means to environments is brought about- and central to that is the process of design itself."

The object of this paper is to open discussion about the formal study of the design activity in its various forms. Research and doctoral programmes in engineering design have received some attention in the past (see for example [2], [3], [4], [5]), but we would argue with undue emphasis on application to industry and on industry take-up of research results [6], rather than research driven by a need to generate fundamental design knowledge.

2 Research in the sciences and design

Research in the sciences is generally associated with the observation and explication of physical phenomena, also human behaviour if people are part of the system being observed. This raises the question as to the types of knowledge gained by research. As set out below in *Table 1*, knowledge exists at varying levels of abstraction and engineering design research may be conducted at one or more of these levels.

Table 1 - Broad comparison of research in the sciences and design

Research deliverables in the sciences	**Research in engineering design**
Laws, theories	none hypothesised
Principles	some proposals have been made, [7]
Models, simulations	some severely constrained work on encapsulated design examples;
Empirical relations, rules, guidelines	"received wisdoms" as noted by [7], need compilation and categorisation
Methods and procedures	considerable body of work on prescriptive and descripti methods;
Concepts	some identified – refer Table 2
Facts	subsumed in above

A variety of research techniques is available for advancing knowledge about engineering design, a situation very similar to that in management science, and in the past it has not been uncommon for significant design research papers to be published in management science journals.

3 Conceptual frameworks for design research

There are various conceptual frameworks in which engineering design research may be conducted. We give some examples, together with citations of papers relevant to the established field of enquiry nominated.

Table 2 Conceptual frameworks for design, with examples

Conceptual framework	**Examples of studies conducted**
Psychology and cognitive science	Design strategies, styles of problem solving, [8]; Intellectual skills of designers, [9]; Creativity [10]
Human information processing	general –[11] ;specific– [12]
Study of design methods	context free– [13] in context– [14]
Computer science	Collaborative design via computer networks, [15]
Social psychology	Interactions between designer(s) and client(s), [16]
Decision making	Incremental innovation, [17] Extrapolatory design, [18] Judgement, subjective probabilities, [19]
Management science	Design process, [20] Concurrent engineering ,[211 Product development, [22]

This table is not intended to be comprehensive but to indicate the wide-ranging nature of engineering design research.

4 Engineering Design in Context

Engineering design is not an isolated phenomenon. It operates in an industrial environment as part of the processes of mediation between engineering expertise and the needs of society at large. Research in engineering design therefore encompasses matters such as:

Innovation and policies for encouraging innovation [23]

Technology transfer, mechanisms for its facilitation. [24]

Engineering design research should help governments formulate appropriate policies for industry and industrial development. If engineering design research does not encompass these matters, the result will be inappropriate or non-existent government policy.

The social and environmental context for engineering design includes the following factors, among others.

- If a product is being designed, whether that product is made to stock or to order. [25]
- If the agency or organisation responsible is a company, then the history of the company - whether it is a start-up or emerging firm or a mature organisation, its position in a supply chain, its association with other related companies in a formal or informal industry cluster [26]

Research papers relying on empirical evidence of industry practices should be careful to identify all relevant factors. In our observation the context of design research is not always adequately described in the research literature. This is a distraction to readers and other researchers who find it difficult to assess the relevance of the work reported to their own programmes.

5 Curiosity Driven Research in Design

There are many drivers and barriers to research in general. In the natural sciences research is driven by a need to understand the "status quo", or why nature behaves the way it does. The results of this type of research are embodied in *hypotheses* about natural phenomena. Two unalienable laws of scientific research as well as some clear guidelines bind these hypotheses to their generation. The laws are expressed in general terms as:

• to test the validity of an hypothesis it must be clearly refutable. Well known examples are hypotheses about conservation of mass or energy;

• of all the proposed hypotheses about natural phenomena, the one relying on the least number of untested conjectures (assumptions) should prevail (Occam's razor or the "law of parsimony").

The most valuable *guideline* to developing hypotheses is that they should be, as far as possible, all embracing, or portable to the widest range of applications.

We need to establish the basic difference between research in the natural sciences and research in design. In natural science we are concerned with existing phenomena and our curiosity about the laws that govern these phenomena and how to predict such phenomena drive our research effort. In the practice of design we are most commonly faced with the need to achieve a goal. Our concern is to find the best or perhaps the least worst way to reach our goal. We are not driven by curiosity, but by our goal. This "goal-seeking" behaviour is the one studied most intently by design researchers. While the process of "best design" is a valid research objective, it is not a subject that will easily yield testable hypotheses..

In design, as in any human operation based discipline, we face the need to study the most complex of natural phenomena, that of human behaviour and human cognition. While there are many guidelines and conjectures in these two sciences, design has yet to formulate any clearly refutable hypotheses. Yet, complex as these phenomena are, they present a fertile field of study for testable hypotheses. Gunther and Ehrenspiel (1999)[27] propose serval hypotheses about design behaviour, but they offer no strategy for testing these.

The overwhelming feature of research papers presented in design journals or conference proceedings is that research in design:

• is many stranded - the keywords suggested for ICED 2001 list under the notional heading of "Design theory and research methodology" 24 items ranging from the clearly understandable *descriptive models of the design process*, to the confusing, such as *design philosophy*[1]. Unfortunately there is no keyword that deals with hypotheses, conceptual frames of reference or the most basic of research planning issues, namely research strategy;

• lacks focused streams of activity- apart from descriptive models of design methods and design education there are few recurring themes in design research publications. This is not a criticism, as much as an observation about the nature of our emerging discipline. The distinctive feature of mature disciplines is that there are streams of research activity that lead to the in-depth study of phenomena. Typical examples are the study of turbulence in fluid mechanics, the study of particulate emissions in environmental research and the study of just noticeable differences in research on haptics[2].

• lacks clear organisation - We conjecture that design researchers are yet to properly grapple with the overwhelming complexity of the discipline. Consequently the approach of dedicated researchers has been to simply "chip away" at the parts of the problem that are accessible. In his keynote address at EDC1998[3], Andreasen remarked that the internal thinking processes that lead to the creation of new products is inaccessible to us. However, the observable responses of designers, novice and expert, to well organised stimuli are clearly worthy of study ([27], [28], [29]).

• often lacks any performance metrics-there have been several attempts at defining such metrics (see for example [30], [31])

It is the presence of these features in the study of design that has made it almost impossible for research funding bodies to calibrate the value of design research.

Yet, as Simon has pointed out, the study of design involves many unanswered questions including the general ones of *"how to make artifacts that have desired properties"* and *"how to design"*. There are many articles dealing with the procedural aspects of designing. This represents one of the recurrent themes in design publications. However these articles invariably deal with ipso facto descriptions after the design event has taken place. Human behaviour suffers from the additional complexity of what may be termed the "Heisenberg's Uncertainty Principle of Behaviour". This uncertainty is generated by the influence of human observation on behaviour we try to observe. This complexity has been dealt with adequately in many behavioural science studies, but has been largely ignored in engineering design research.

One of the most recent articles on design research is that by Culley [5]. Culley's article he explores the various dimensions of design research, but essentially as a product generating

---

[1] philosophy is etymologically the love of thinking
[2] haptics is the study of touch and feel in humans and primates
[3] Engineering Design Conference held at Brunel University

process. The issue of human cognition and human behaviour is not addressed in that paper. Clarkson and Hamilton [9], study human cognition applied to design, but they do not formulate hypotheses in a clearly refutable way.

6 A Strategy for Research in Design

Our theses in this article are that :

- formulating hypotheses is a distinctive feature of mature scientific disciplines;
- we can address "pre-hypothesis" research by studying and identifying patterns of behaviour and factors influencing such behaviour; Some of this work has already been addressed ([9]; [27]; [32], [33]);
- there are many interesting phenomena in design cognition and behaviour that should be addressed in our research programmes;

Some questions that might engender some curiosity driven experimental or theoretical studies follow:

- what type of background or training might enhance the construction of three-dimensional spatial objects, a skill so important to design visualisation?
- what are the received wisdoms (rules of thumb, heuristics) that govern various knowledge domains where design has an application?- can they be catalogued?
- what cognitive content is provided by project based design learning?; how are changes in cognitive behaviour influenced?
- what cognitive enhancement takes place in moving from novice to experienced designer? is this transportable between knowledge domains?
- can cognitive mapping (concept mapping) of design enhance the design process?
- what aspects of design cognition can be modelled in a computer?
- can negotiating behaviour be modelled in the computer? (see for example [34], [35] on semantic information processing)
- can the social dynamic of group design be modelled adequately?; Can we identify appropriate criteria for a "successful" model?

As an example hypothesis we might consider :

"experienced designers are more sensitive to sub-problems than novice designers".

Hypotheses could be considered in the context of Guilford's "cognition of implication", that formed part of his studies of human intellectual capabilities [36].

Human behaviour and cognition have received substantial study in psychology, education and cognitive science. We strongly advocate that design researchers adapt these research studies to our discipline (see [28]).

7 Long-term Impact of Design Research

An essential part of research of any kind is its ultimate impact on the knowledge base it seeks to advance. In the natural sciences the most common metric used to measure impact is citation index of archival publications. In engineering design research a citation is not an appropriate metric of performance, because archival publications in design appear in such a wide range of journals and conference proceedings, that citation measures are either not available or are unreliable. Journals such as Journal of Engineering Design, Research in Engineering Design, Artificial Intelligence in Engineering Design and Manufacturing and Design Studies are relatively recent phenomena (approximately one decade). The lack of a consistent archival build-up of design research knowledge over a longer period has added to the rather confused state of design research. Perhaps a database of design research results of wide applicability over a longer time-frame could be generated as a means of addressing this difficult issue.

8 Conclusion

Recent influential reports and studies of engineering design research have emphasised industry take-up and the generation of results which can be useful in specific applications. To redress the apparent imbalance we have argued in this article for an all-embracing approach to engineering design research covering fundamental knowledge and policy issues as well as industrial applications. To conclude we summarise the points made.

- Engineering design research is multi-faceted. It draws on the conceptual frameworks of a variety of established scientific disciplines and delivers results at varying levels of generality and specificity. [sections 2 and 3]
- Engineering design research is not confined to studies of technical problem solving *per se*; it encompasses the political, social and economic environments in which engineering design researchers work the needs of clients and through them society at large. [section 4]
- As engineering design matures as a discipline, the nature of the associated research activity will shift from a pre-hypothesis stage of organised data collection to a higher stage requiring enformulation and verification of hypotheses. Proposals for advancing engineering design research in this way have been put forward. [sections 5 and 6]
- To ensure the impact of the results of engineering design research in the long term, a comprehensive, archival database needs to be established, cataloguing research results in an accessible form. [section 7]

The danger for us, the engineering design research community, in concentrating our efforts on the practical and professional training aspects of design is that we will continue to be perceived by the natural science community and granting bodies as a "soft discipline" with no clearly testable hypotheses, or clearly identifiable deliverables. This article is intended to engender earnest discussion of research programmes and research streams that are driven by curiosity about the social and cognitive behaviour of engineering designers and the contexts in which they work. In the process we hope to encourage the deployment of substantial bodies of knowledge available from the behavioural and cognitive sciences.

References

[1] Simon, H. A. (1998) The Sciences of the Artificial, Third Edition, Cambridge:MIT press

[2] Finger, S. and Dixon J. R. (1989) A Review of Research in Mechanical Engineering Design. Part I: Descriptive, Prescriptive and Computer-based Models of the Design Process, Research in Engineering Design, (1):51-67

[3] Finger, S. and Dixon J. R. (1989) A Review of Research in Mechanical Engineering Design. Part II: Representations, Analysis and Design for the Life Cycle, Research in Engineering Design, (1):121-137

[4] Dixon, J. R. (1992) Why we Need Doctoral Programs in Design, Mechanical Engineering, February, 75-79

[5] Culley, S. J. (1999) Future Issues in Design Research, Final Report of DTI and EPSRC Workshop, University of Bath.

[6] Royal Academy of Engineering, Invitation to Contribute to Study of Engineering Design Research, Personal Communication, London , 2000

[7] French, M. J. A Collection of Design Principles, Ch. 15 in Proceedings of the International Workshop on Engineering Design and Manufacturing Management [Samuel, A.E.,Ed. 1990], Elsevier Science Publishers, Amsterdam.

[8] Eisentraut, R. (1999) Styles of Problem Solving and their Influence on the Design Process. Design Studies, 20: 431-437.

[9] Clarkson, P.J. and Hamilton, J.R. (1999) Signposting the Design Process, Proceedings of ICED 99 Munich, (1):1017-112

[10] Owens, W.A.(1969) Cognitive, Noncognitive, and Environmental Correlates of Mechanical Ingenuity. Journal of Applied Psychology (53): 199-208.

[11] Miller, G.A (1956) The Magical Number Seven, Plus or Minus Two : Some Limits on our Capacity for Processing Information. Psychological Review, (63: 81-97.

[12] Lewis, W. P. The Role of Intelligence in the Design of Mechanical Components. Chapter 4 in T. Sata and E. Warman (eds.), Man-Machine Communication in CAD/CAM, North-Holland, Amsterdam, 1981.

[13] Samuel, A.E. and Weir J.G.(1997) A Parametric Approach to Problem Enformulation in Engineering Design, Proceedings of ICED97, Tampere Finland, August, (3): 83-90

[14] Kusiak, A.and Larson, N. (1995) Decomposition and Representation Methods in Mechanical Design, Journal of Mechanical Design, Trans. ASME (117B): 17-24 June ASME New York NY USA 1050-0472 JMDEEC

[15] Rodgers, P. A. and Huxor, A. P. (eds, 2000.) Web-based AI Design Tools, Design Studies, (21):119-220.

[16] Mitroff, I. I. (1968) Simulating Engineering Design - a Case Study on the Interface Between the Technology and Social Psychology of Design. I.E.E.E. Transactions on Engineering Management, (EM-15):178-187.

[17] Marples, D. L. (1961) The Decisions of Engineering Design. I.E.E.E. Transactions on Engineering Management, (EM-8): 55-71.

[18] Dollin, F. (1963) Some Design Problems in the Development of Very Large, High Speed Turbines. Proceedings, Institution of Mechanical Engineers, (177): 221-237.

[19] Lewis, W. P. (1985) Subjective Assessment of Probabilities in the Design of Mechanical Systems. Proc. 3rd Int. Conf. on Engineering Design, Hamburg, pp. 555-564.

[20] Hales, C. Analysis of the Engineering Design Process in an Industrial Context. Gants Hill Publications, Eastleigh, U.K., 1991.

[21] Backhouse, C. J. and Brookes, N. J. (eds.). Concurrent Engineering : What's Working Where. Gower Publishing, Aldershot, U.K., 1996.

[22] Griffin, A. (1993) Metrics for Measuring Product Development Cycle Time. Journal of Product Innovation Management, (10): 112-125.

[23] Lewis, W. and Borg, G. (1983) The Diffusion of New CAD/CAM Technology in Australian Manufacturing Industry, Proc. 7th Int'l Conference. on Prod. Research, Windsor, Ontario, 358-365

[24] Burvill C. R. and Samuel, A. E. (1998) Engineering Centres as Targets for Industrial Innovation, Industry and Higher Education, 13(1):54-60

[25] Maffin, D., Alderman, N. ,Braiden, P., Hills, W. and Thwaites, A. (1995) Company Classification: A new perspective on Modelling the Engineering and Product Development Process, Journal of Engineering Design, (6): 275-289

[26] Porter, M. E. The Competitive Advantage of Nations. Macmillan, London, 1990

[27] Günther, J. Ehrlenspiel, E. (1999) Comparing Designers From Practice and Designers with Systematic Design Education, Design Studies, 20(5):439-452

[28] Gero, J. S. (2000a) Computational Models of Innovative and Creative Design Processes, Technological Forecasting and Social Change, 64:183-196.

[29] Gero, J. S. (ed.) (2000b) Artificial Intelligence in Design'00, Kluwer, Dordrecht , 719pp

[30] Smith R.P., Morrow, J. A.(1999), Product Development Process Modelling, Design Studies, 20(3):237-262

[31] Stone, R.B., Wood, K. L., Crawford, R. H.(1999) Using Quantitative Functional Models to Develop Product Architectures, Design Studies 21(3):239-260

[32] Badke-Schaub, P. Wallmeier, S., Dörner, D.(1999) Training for Designers: A Way to Reflect Design Processes and Cope with Critical Situations in Order to Increase Efficiency, Proceedings of ICED 99 Munich, (1):205-216

[33] Wallmeier, S., Badke-Schaub, P., Birkhofer, H.(1999) Training for Designers: Empirical Results of Trained Designers in Selected Design Processes, Proceedings of ICED 99 Munich, (1):211-216

[34] Minsky, M. (1975) A framework for Representing Knowledge. In The Psychology of Computer Vision, Ed. P. Winston. New York: McGraw Hill

[35] Minsky, M. (1985) The Society of the Mind, New York: Simon &Schuster, Inc.

[36] Guilford, J. P. (1967) The Nature of Human Intelligence, New York: McGraw Hill

Professor Andrew Samuel,

Department of Mechanical and Manufacturing Engineering,

The University of Melbourne

Grattan St. Parkville Vic. 3010

+61-3-8344-6752

+61 - 3-9347-8784

aes@mame.mu.oz.au

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A MEASURE FOR DESIGN COUPLING

G Arcidiacono, G Campatelli, and H Lipson

Keywords: Axiomatic Design, optimisation techniques, design understanding

1 Abstract

The degree of coupling between design modules affects design quality: designs with high coupling are generally more complex to adapt, while uncoupled designs are simpler and their adaptation requires fewer iterations or even a single pass. The design process of products involving a high degree of coupling is consequently inefficient and often many iterations are required to complete it. In order to assess and compare the level (degree) of design coupling it is necessary to specify a general quantitative measure. Two such measures – Reangularity and Semangularity [Rinderle, 1982 (10)] – are already being used for this task [Suh, 1990 (13)]. This paper proposes a new method based on measuring the direct consequences of coupling.

The analysis is based on the Axiomatic Design framework which assumes that a design can be described formally as a matrix representing the relationship between Functional Requirements and adjustable Design Parameters. This paper also intends to demonstrate that the degree of coupling approach to the choice of a possible design solution is faster than the classical method based on the information content.

We will analyse and optimise the design of a water pump to show how coupling functions are to be used

2 Axiomatic Design

The primary goal of Axiomatic Design (AD) is to establish a systematic foundation for the design process by means of two fundamental axioms and a set of implementation methods [Suh, 1990, 1999 (13) (14)]. The axioms are:

Axiom 1: The Independence Axiom: maintain the independence of Functional Requirements (FRs).

Axiom 2: The Information Axiom: minimise the information content in design.

Guided by the design axioms, AD maps the Functional Requirements FR_i to the Design Parameters DP_i. The FR_i and DP_i sets can be interpreted as two vectors FR and DP whereas the mapping instructions form a matrix called Design Matrix. The possible matrix structures (diagonal, triangular or full) produce three types of design: uncoupled, decoupled and coupled, represented in order of the extent to which coupling is present. In the first case the

FRs are independent of one another, in the second only if the FRs are in a certain order they are independent, while in the third case they are dependent. The information content of the proposed design has to be computed to enable the search for the best design to be carried out.

3 Coupling Functions

The degree of coupling criterion measures to what extent the design is coupled. Two functions are developed: one for decoupled designs and another for coupled designs; in fact, as will be explained later, the coupling for these two design types is defined in different ways. We seek an alternative for Reangularity and Semangularity functions because these measures do not capture some aspects of coupled designs [Campatelli, 2001 (3)]. With the introduction of the proposed coupling functions it is possible to pass from a crude and discrete scale of coupling to a continuous scale, as shown in Figure 1 and Figure 2.

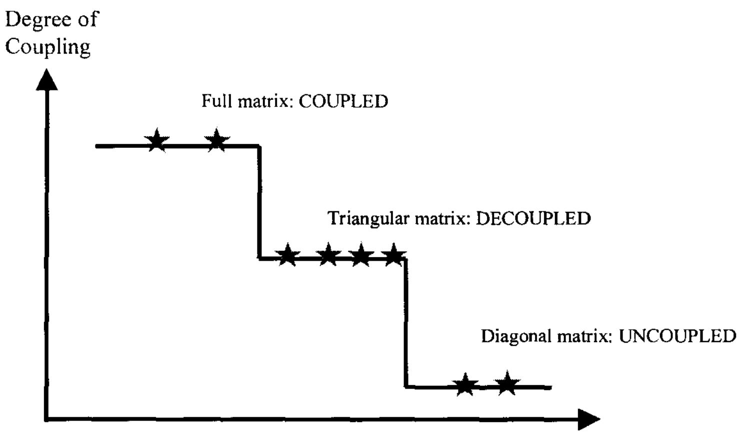


Figure 1. Representation of the coupling for a set of proposed designs; each proposed technical solution that satisfy the design is represented by a star.

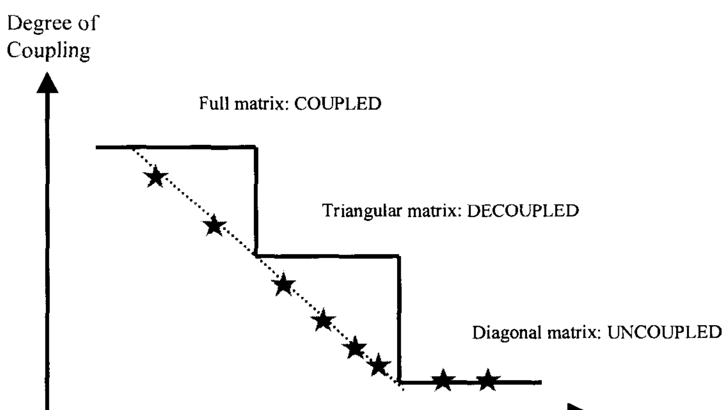


Figure 2. Degree of coupling scale using the coupling functions.

The proposed functions also have the advantage that they require less computing time than the information content does; in fact, as will be explained later, they require less data.

4 Coupling function for coupled designs

In the case of coupled designs the system is represented by a full matrix (Figure 3). Since elements of the matrix all have different physical units, DP values cannot be found in a single calculation step using linear algebra, but require an iterative process.

$$\delta\begin{Bmatrix} FR_1 \\ FR_2 \\ FR_3 \\ FR_4 \end{Bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \delta\begin{Bmatrix} DP_1 \\ DP_2 \\ DP_3 \\ DP_4 \end{Bmatrix}$$

Figure 3. Representation of a coupled design

Elements of the design matrix represent sensitivities of DPs to FRs, according the following formula

$$a_{ij} = \frac{\partial FR_i}{\partial DP_j} \tag{1}$$

In the case of coupled designs the definition of the degree of coupling is defined by the difficulty of adjusting the DP values to meet the FR specifications. This difficulty is measured as a function of the number of iterations required to solve the system. Smith (1997) has applied a similar approach to calculating functional coupling in the DSM (Design Structure Matrix) method [Smith, 1997 (12)]. The calculation of the system's rate of convergence is simpler as it is inversely proportional to the number of interactions. Firstly the calculation of the rate of convergence requires the best iterative method to be chosen from among the possible alternatives. Jacobi's method, which is single stepped and not over-relaxed, was chosen for its simplicity.

The system can be represented with the following notation:

$$b = A \cdot u \tag{2}$$

where b is the vector of the FRs and u is the vector of the DPs. Equation (3) updates vector u at every calculation level:

$$u^{(n+1)} = G \cdot u^{(n)} + c \tag{3}$$

where G and c are expressed by the following formulae

$$G = \begin{cases} 0 & diagonal \\ -\dfrac{a_{ij}}{a_{ii}} & extra-diagonal \end{cases} \quad \text{and} \quad c = \frac{b_i}{a_{ii}} \tag{4}$$

It is therefore possible to obtain the rate of convergence using the following equation [Young, 1971 (8)]

$$Rc = -\log_n(S_G) \tag{5}$$

where S_G is the spectral radius of the G matrix i.e. its maximum eigenvalue.

$$S_G = \max(eig(G)) \tag{6}$$

To evaluate this function, the design matrix [A] (Figure 3) must be expressed numerically. However, it is not always possible to resolve equation (1) numerically especially for the first levels of the decomposition. It is therefore necessary to ascertain the extent of the interactions between the FRs and the DPs in another way, for example using Functional Analysis [Campatelli, 2001 (3)]. In this example the normalised value (=1) was assigned to the diagonal elements. The off-diagonal elements were calculated comparing them with the previous ones. The best possible matrix is usually the one with the dominant diagonal. In the following figure four different cases of coupled matrices are analysed. As mentioned before the fourth case is the worst, because the iterative loop involves all the DPs. In this case it is essential that the numerical values of all the matrices considered are known. In the second case the worst rate of convergence of the two sub-matrices[1] has to be considered.

$$\begin{bmatrix} x & 0 & 0 & 0 & 0 \\ x & 1 & 0 & 0{,}5 & 0 \\ x & 0{,}2 & 1 & 0 & 0 \\ x & 0{,}1 & 0{,}1 & 1 & 0 \\ x & x & x & x & x \end{bmatrix}$$

Case 1

Rc = 1,236

$$\begin{bmatrix} x & 0 & 0 & 0 & 0 \\ x & 1 & 0{,}4 & 0 & 0 \\ x & 0{,}2 & 1 & 0 & 0 \\ x & x & x & 1 & 0{,}1 \\ x & x & x & 0{,}7 & 1 \end{bmatrix}$$

Case 2

Rc = 1,263

$$\begin{bmatrix} 1 & 0 & 0{,}3 & 0 & 0 \\ 0{,}9 & 1 & 0 & 0{,}5 & 0 \\ 0 & 0{,}2 & 1 & 0 & 0 \\ 0{,}3 & 0{,}1 & 0{,}1 & 1 & 0 \\ x & x & x & x & x \end{bmatrix}$$

Case 3

Rc = 0.745

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0{,}5 \\ 0.9 & 1 & 0 & 0 & 0 \\ 0 & 0{,}2 & 1 & 0 & 0 \\ 0{,}3 & 0{,}1 & 0{,}1 & 1 & 0 \\ 0{,}2 & 0{,}5 & 0{,}3 & 0{,}7 & 1 \end{bmatrix}$$

Case 4

Rc = 0.359

Figure 4. Possible design matrices

[1] The best coupled design is the one with the greatest possible rate of convergence. It can be easily obtained by following these general rules:

- Reduce the number of off-triangular elements
- Reduce the ratio of the value of the off-triangular and diagonal elements in the same line
- Reduce the length of the calculation loops, i.e. move the off-triangular elements closer to the diagonal.

5 Coupling function for decoupled designs

In the case of decoupled designs the system is represented by a matrix which is at most triangular (Figure 5).

$$\delta\begin{Bmatrix} FR_1 \\ FR_2 \\ FR_3 \\ FR_4 \end{Bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & 0 & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \delta\begin{Bmatrix} DP_1 \\ DP_2 \\ DP_3 \\ DP_4 \end{Bmatrix}$$

Figure 5. Representation of a decoupled system

Unlike the previous case iterations are not required to find the DP values which meet the requirements of the FRs of the system. The definition of coupling will therefore be different from that applied in the coupled case. The extent to which each FR is influenced by an individual DP will be adopted as the degree of coupling. This characteristic of the system can be quantified by numerically calculating the angles between the axes of the FRs and the axes of the respective DPs. It is intuitive to think that the smaller the angle between an FR and a DP (zero in the most favourable case) the more significant the influence of that DP on the FR. Consequently, the other DPs will have less influence on that FR and the system will therefore result less coupled. The coupling function[2] of a system containing n FRs will therefore be as follows:

$$f_{COUPL} = \frac{\sum_{i=1}^{n} FR_i \angle DP_i}{n} \tag{7}$$

The vectors of the DPs axes are made up of the rows of the design matrix while the FRs axes are the rows of an identity matrix[3]. The angle between two vectors, each composed of n elements, can be calculated using the geometric formula:

$$FR_i \angle DP_i = a\cos\left(FR_i^T \cdot DP_i\right) \tag{8}$$

6 Case study: a water pump for an automotive engine

An automotive engine water pump is considered as a case study (Figure 6), with the object of demonstrating and applying the coupling functions proposed.

---

[2] The general rules to decrease the coupling are the following:

- Reduce the number of off-diagonal elements
- Reduce the value of the off-diagonal elements

[3] By definition the FRs have to be independent

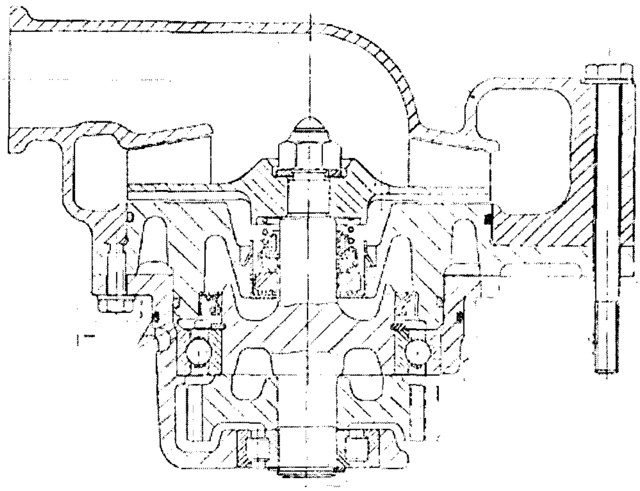


Figure 6. Section of the pump unit being considered

For the sake of simplicity only the first AD break-down level of the pump unit is considered (Table 1).

FR1 Guarantee performance	1	0	0.3	0	0.7	DP1 Centrifugal impeller
FR2 Guarantee sealing	0	1	0.6	0	0	DP2 O-ring and gaskets
FR3 Guarantee pump life	0	0	1	0	0	DP3 Materials
FR4 Allow installation	0	0	0	1	0.2	DP4 Positions of the screws
FR5 Allow maintenance	0	0	0	0	1	DP5 Removing components

Table 1. First AD breakdown level of the pump unit

In this case the design is decoupled, so the equation to be applied is the (7). This provide the following outcome: f_{COUPL}=0.2440. As mentioned in note 2 a decoupled design can be improved by reducing the degree of coupling so eliminating (or at least reducing) the off-diagonal elements. If the design would be coupled the equation to be applied would be the (5) and the improvement would be reducing the off-triangular elements. So the next step will be the analysis of which off-diagonal element could be affected by some different technical solution. These are suggested by the AD representation of the product, in fact that clarify the problems that are present in a design, so it's easier to produce a solution that solve them. For example the coupling element between FR_1 and DP_5 can be eliminated if the duct through which the water passes is shaped differently. From cross analysis between the AD decomposition and Functional Analysis it is possible to trace the causes which determine the existence of each off-diagonal coupling element [Campatelli, 2001 (3)]. This analysis in fact is able to identify the causes for the coupling of the upper levels starting from the analysis of the lower ones. In this way we have a very focused description of the reason that produce the coupling with all the components related to the specific coupling element that we want to eliminate. In this particular case the element came into being from the contrasting requirements of avoiding cavitation, i.e. having the water entry canal as rectilinear as possible, and the need for reduced over-all dimensions. The problem in fact arise from the sharp edge near the vertical part of the canal that arrive to the rotor. This solution is proposed in Figure 7.

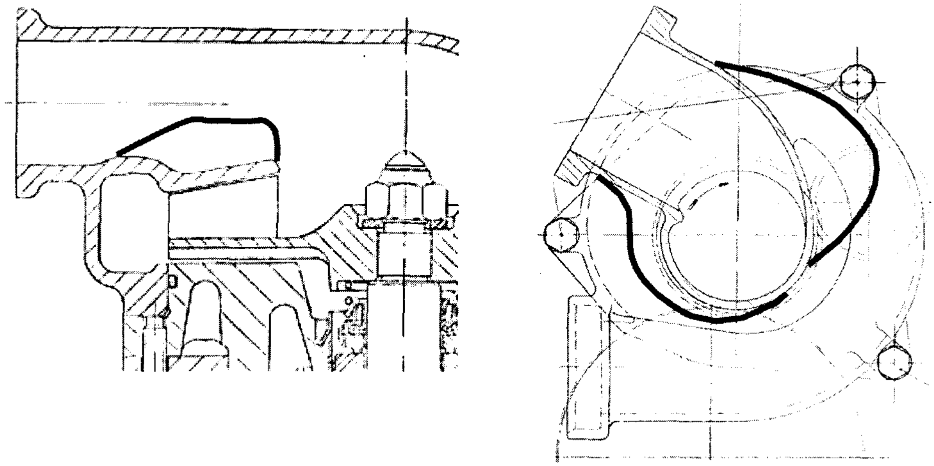


Figure 7. The design solution to improve the project

This solution provided the following design matrix:

FR1 Guarantee performance	1	0	0.3	0	0	DP1 Centrifugal impeller
FR2 Guarantee sealing	0	1	0.6	0	0	DP2 O-ring and gaskets
FR3 Guarantee pump life	0	0	1	0	0	DP3 Materials
FR4 Allow installation	0	0	0	1	0.2	DP4 Positions of the screws
FR5 Allow maintenance	0	0	0	0	1	DP5 Removing components

Table 2. Design matrix of the improved solution

The value of equation (7) becomes $f_{COUPL}=0.1577$, lower than the previous situation, that means a less coupled design. During this analysis more solutions have been proposed with the result to reduce the general coupling of the design and consequently to improve its quality.

7 Conclusions

The direct consequence of coupling in a design is the introduction of an iterative adjustment process. But the degree of coupling may be different in alternative decoupled and coupled designs. It can be computed from the angle between the subspace (decoupled cases) or from the rate of convergence (coupled ones). This measure of coupling may be dependent not only on the diagonal or triangular form of the matrix, but also on such issues as the relative magnitude of elements, dominance of the diagonal, location of non-zero elements within the matrix, and the size and nesting of loops created by the iterative process. While exact prediction of the degree of coupling of a general design process can be difficult, this paper has proposed several factors that may assist in initial quantification and provide guidelines for design practice. The process of reducing the coupling for a design, using the given measures and guidelines, is shown in the application to the automotive water pump. The technical solutions proposed (for paper length constraint only one of these is shown) have reduced the coupling of the design increasing so the general Quality level of the product.

8 References

[1] AA.VV., "Journal of computation", SIAM, 1976-1990

[2] Arcidiacono G., "Axiomatic Design for Reliability", ATA Motor car engineering Journal, Vol.53 n.9/10, pp.309-315, 2000

[3] Campatelli G., "Sviluppo di una misura per l'accoppiamento funzionale mediante l'Axiomatic Design e applicazione al gruppo pompa per motori a V di IVECO", University of Florence Master Thesis, 2000

[4] Cover T. M., "Elements of information theory", Wiley, 1991

[5] Engelhardt F., "Improving products and system by combining Axiomatic Design, quality control tools and Designed Experiments", proceedings, ICAD 2000

[6] Frey D., "Computing the information content of de-coupled designs", proceedings, ICAD 2000

[7] Hinton M., "Numerical methods for coupled problems", Pineridge Press, 1981

[8] Young D. M., "Iterative solution of large linear systems", Academic Press, 1971

[9] Lipson, H., Suh. N.P., "Towards A Universal Knowledge Database For Design Automation", proceedings, ICAD2000

[10] Rinderle J. R., "Measure of functional coupling in design", MIT Ph.D. Thesis, 1990

[11] Sahlin M., "A systematic approach for decision making in a Concurrent Engineering environment", proceedings, ICAD 2000

[12] Smith R., "Identifying controlling features of engineering design iteration", 1997

[13] Suh N. P., "Principles of Design", Oxford Press, 1990

[14] Suh N. P., "Axiomatic Design: Advance and Applications", Oxford Press, 2000

Gabriele Arcidiacono
University of Florence, DMTI, Via di S.Marta 3, Italy, Tel: 0039-055-4796396, Fax: 0039-055-4796394, E-mail: g.arcidiacono@dmti.unifi.it

Gianni Campatelli
University of Florence, DMTI, Via di S.Marta 3, Italy, Tel: 0039-055-4796291, Fax: 0039-055-4796394, E-mail: campatellig@dmti.unifi.it

Hod Lipson
Mechanical Engineering Dept., Massachusetts Institute of Technology, Room 31-261E, 77 Massachusetts Ave., Cambridge MA 02139-4307, USA, Tel: 001-617-258-0264, Email: hlipson@MIT.edu

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

ANALYSIS OF CONCEPTUAL DESIGN CHAINS FOR THE UNKNOWN INPUT/KNOWN OUTPUT PATTERN

R Žavbi and J Duhovnik

Keywords: Prescriptive models of the design, chaining of physical laws, algorithm, combinatorial explosion.

1 Introduction

The concept of use of physical laws is based on the fact that no technical system can function contrary to physical laws. The principle of chaining is illustrated in Figure 1. At first glance this may appear trivial, since a chain of physical laws can be created by analysing the existing technical systems. However, it is our assumption that chaining enables the synthesis of conceptual design chains for technical systems.

This paper has two objectives. First we would like to explain the reasons for assuming that a combinatorial explosion may occur, and then we will show that chaining of physical laws does not produce a combinatorial explosion, even though it would be reasonable to expect it, taking into account the algorithm design, and its occurrence would mean that the existing algorithm of chaining is generally inappropriate for the conceptual design of technical systems.

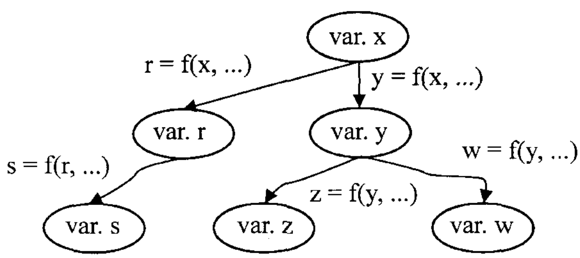


Figure 1. Illustration of chaining of physical laws (var.-variable)

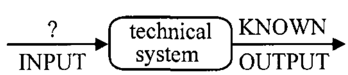


Figure 2. Schematic of the unknown input / known output pattern

In the second part, an example of a conceptual design chain for a voltage generator will be illustrated, which is generated by chaining with the use of the unknown input/known output pattern. The schematic of this pattern is shown in Figure 2. This will be followed by a partial analysis of the generated conceptual design chains and a description of the advantages of such an approach in designing technical systems.

1.1 Review of basic terms

The purpose of this subsection is to review and briefly explain terms used in the paper.

Function: The purpose of the future technical system expressed in a neutral manner, i.e., without indicating any solutions. Examples: sound detection, torsional moment transfer, maintenance of a certain liquid level, etc.

Physical law: The term is here used in its widest sense; it represents the functional relation between variables, geometrical parameters, material and basic constants.

Base variable: The term is taken from physics, in which it is postulated that all parameters (with the exception of the basic ones) can be defined by the basic ones, which are: length, time, mass, electrical current, temperature, amount of substance and luminous intensity [1].

Binding variable: Variable common to a physical law and its successor in the conceptual design chain.

Chaining: Searching for a successor physical law in the conceptual chain using the binding variable. The binding variable may be used to find a physical law that contains such a variable, but it needs to be of the opposite type, e.g., variable-cause can find variable-effect or vice versa.

Conceptual design chain: A chain of physical laws that fulfil the required function.

2 Algorithm

The algorithm is based on the idea of binding physical laws (in fact: their abstractions, presented only by parameters, without operators) and their complementary basic schematics through the binding variables (Figure 3), assuming that it is possible to design technical systems by chaining physical laws [2]. The physical laws in the chain and the corresponding complementary basic schematics represent the conceptual design of a technical system [3].

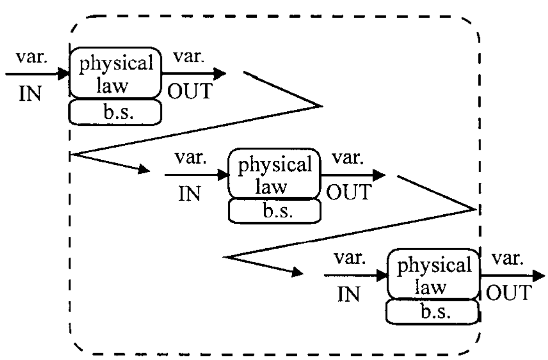


Figure 3. Idea of binding physical laws and their complementary basic schematics (var.: variable, b.s.: basic schematic)

Similar ideas are dealt with in work of Ulrich and Seering [4] where synthesis of single-input single-output dynamical systems are treated, Chakrabarti et al. [5] where basic approach to automated synthesis of solution principles for micro-sensor designs is presented, Welch and Dixon [6] where behavioral reasoning for guiding conceptual design is used, Bracewell et.al [7] where design aid for the conceptual design is described, Sushkov et al. [8] where innovative design based on sharable physical knowledge is treated and in work of Bratko [9] where an idea for innovative design based on a function that is given through examples of behaviour of a future technical system, and a description of the functioning of technologically available components is described.

The basic algorithm is as follows [3]:

Step

1

Deduce the characteristic variable from the function of the technical system to be designed;

2

Search for all abstractions of physical laws that contain this characteristic variable, and then use them to generate the successors of the root node such that they contain the remaining variable from the found abstraction of the physical law.

CONDITION:

IF
the generated node contains a variable from the sets of geometric, material and base variables

THEN
STOP the generation of successors of this node;

3

For other nodes that do not fulfil the CONDITION, search for all abstractions of physical laws that contain the variable of an individual node and generate their successors such that they contain the remaining variable from the found abstraction;

4

Repeat step 3 until all leaf nodes fulfil the CONDITION.

The following limitations need to be considered:

- Only variables of the opposite type can be used to search abstractions (variable-cause can be used to find the physical law abstraction containing variable-effect, and vice versa). The variables in the database of abstractions of physical laws have type designations (cause or effect), which serve to indicate causal relations.

2.1 Combinatorial explosion

One of the basic challenges encountered in researching various approaches to the conceptual design of technical systems is to prove their wide applicability. Combinatorial explosion generally occurs in chaining (on which the presented algorithm is based), so many researchers have doubts regarding the appropriateness of any approach to the designing of technical systems which is based on chaining. When the problem of chaining is discussed, search-related problems are mainly addressed, because the problem of generating conceptual design chains is generally a problem of searching for a path from the initial node to the goal node, where the initial node is the input/output variable and the goal node is the output/input variable of a technical system. Physical laws (i.e. their abstractions) represent the rules for connecting two nodes, while the path from the initial node to the goal node represents the conceptual design chain of a technical system.

Two basic search strategies are available in the AI domain: depth-first search and breadth-first search. Both belong to the group of blind searches, since in the case of several successors the path is not continued in the direction of the best one (whatever this may mean in a particular case; here also lies an opportunity to improve the conceptual design algorithm for technical systems in order to be able to take certain specific design requirements into account and use them as heuristic guidance (one of possibilities to improve the algorithm is Analytic Hierarchy Process-AHP)), but all the possible paths permitted by the rules are generated.

A typical problem associated with searches is a problem of combinatorial complexity. For non-trivial problems, the number of alternatives is so high that the problem of complexity is frequently critical – let us see why. If each node has b successors (i.e. branching factor b), then the number of paths with length l from the initial node is b^l. The set of paths thus grows exponentially with path length, which leads to a combinatorial explosion. Naturally, the previous explanation is based on a simple case of a uniform branching factor. If abstractions of physical laws are chained, the branching factor varies between the nodes (each input/output variable can generate several output/input variables with several physical laws, Figure 4) and the length of conceptual design chains also varies. However, the risk of a combinatorial explosion remains.

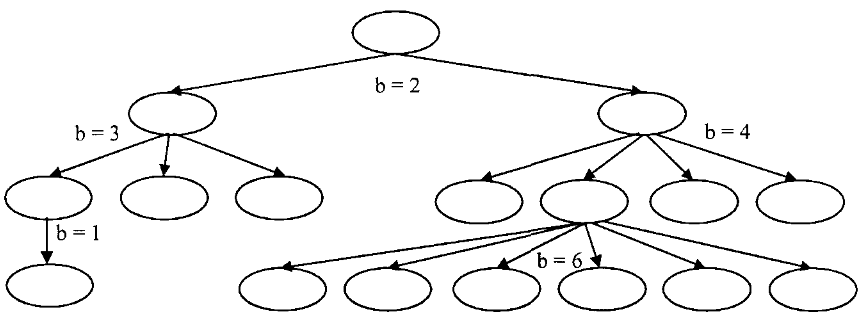


Figure 4. An example of non-uniform branching factor for chaining of conceptual design chains

Considering the actual possibility of the occurrence of a combinatorial explosion, it was decided that the generation of conceptual design chains would be tested using three sizes of sets of physical law abstractions. With a computer program based on the algorithm, conceptual design chains were generated for the available output variables in the unknown input/known output pattern.

A set of 30 abstractions were used for the first test, 60 for the second and 139 for the third. The set of physical laws was made on the basis of Koller's catalogue [10] and certain other sources of physical laws [1]. The results, i.e. the number of generated conceptual design chains depending on the size of sets of physical law abstractions for a basic and a supplemented algorithm, are shown in Figure 5. The number of conceptual design chains for some of the available variables (the result for force is shown in Figure 5 – force-basic) is actually large (only if a set of 139 abstractions is used; for sets of 30 and 60 this number is small), but one cannot speak of a combinatorial explosion here. Results show that branching factor is non-uniform (as expected) and it varies from 1-18, while length varies from 1-19.

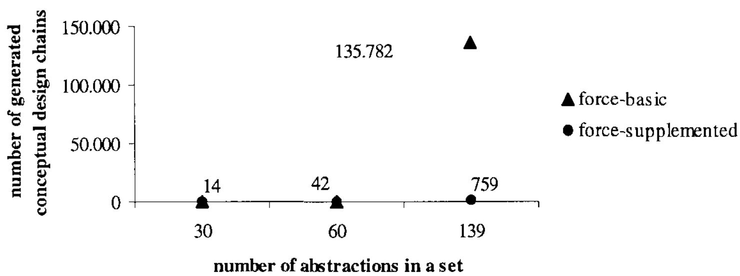


Figure 5. Number of generated conceptual design chains vs. set size for force as known variable in the unknown input / known output pattern

A more detailed examination of the generated conceptual design chains shows that in many chains individual variables are repeated several times. This means repeated fulfilment of a part of the function, which is completely unnecessary. An additional condition was therefore introduced in order to prevent this:

- *No physical law may be repeated in any individual conceptual design chain (i.e. generated path).*

The basic algorithm was supplemented (it resulted in supplemented algorithm) with the above limitation and the program was corrected accordingly. This condition considerably reduced the number of chains designed (again, only for a set of 139 abstractions, while for sets of 30 or 60 abstractions the number of generated chains remained the same), as can be seen in Figure 5 for force as the known output variable (force-supplemented). Results also show that length varies now from 1-9, while branching factor variation remains the same. Such a reduction in the number of generated conceptual design chains is characteristic of all the available output variables (voltage, for example - see the revealing Figures 7 and 8), which is an empirical proof that the designing of technical systems with the presented algorithm is not problematic from the viewpoint of combinatorial explosion. A non-uniform branching factor and length make theoretical proving more difficult. The authors have not found such theoretical proof, yet. Since empirical proving is commonly used in artificial intelligence, we accept the approach (i.e. empirical proof) as appropriate.

3 Conceptual design chains for unknown input/known output pattern

As far as the input and output variables of technical systems are concerned, we distinguish between three general patterns [3]. The most frequently used is the *known input/known output* pattern, in which the designer is required to determine the input and the output variable of a technical system in advance. However, this method also narrows down the range of alternatives, because not all physically possible solutions and combinations of the input and output variables are known in advance. The two other possible approaches are the *known input/unknown output* pattern and the *unknown input/known output* pattern. In contrast with the first pattern, the latter two offer greater possibilities for innovations. If one of the variables (input or output) of a technical system is unknown in advance, the generation of a larger set of conceptual design chains is possible. Subsection 2.1 also shows that there is no real risk of a combinatorial explosion.

Figure 6 shows one of the examples of a conceptual design chain for a voltage generator with the complementary basic schematics, generated for voltage as known output variable in the unknown input/known output pattern.

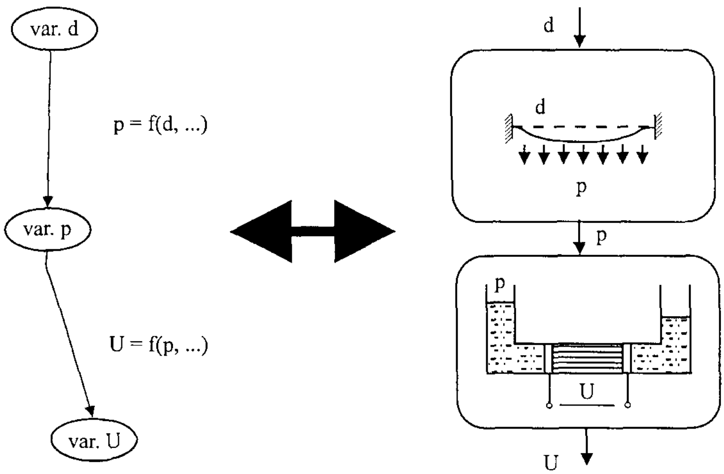


Figure 6. Schematic of chaining a conceptual design chain for a voltage generator (variables (d - deflection, p - pressure, U - voltage), abstractions of physical laws (y = f(x, ...)) and complementary basic schematics)

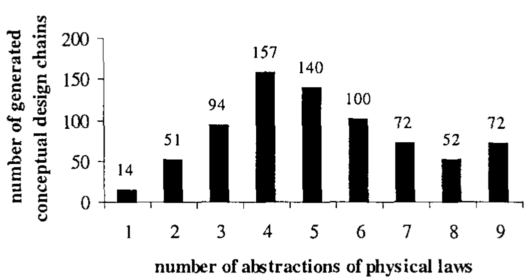


Figure 7. Histogram of a family of conceptual design chains (sum =752) for voltage as output variable (unknown input/known output pattern; supplemented algorithm)

Figure 7 is a histogram presenting the distribution of conceptual design chains with respect to the number of contained abstractions of physical laws for voltage as the known output variable (*unknown input/known output* pattern). The histogram was created on the basis of the results obtained with a supplemented algorithm of chaining physical law abstractions (with a set size of 139). With a view to checking whether there is a possibility of a combinatorial explosion, Figure 8 shows a histogram created using the results obtained with a basic chaining algorithm.

A comparison with the results presented in Figures 7 and 8 shows that the supplemented algorithm generates chains with a maximum length of l=9 (i.e. the number of abstractions per chain), while the basic algorithm generates chains with a maximum length of l=18. The number of conceptual design chains of certain lengths is also smaller. The same pattern of reduction (shorter maximum lengths, smaller number of conceptual design chains of certain lengths) is characteristic of all the available output variables.

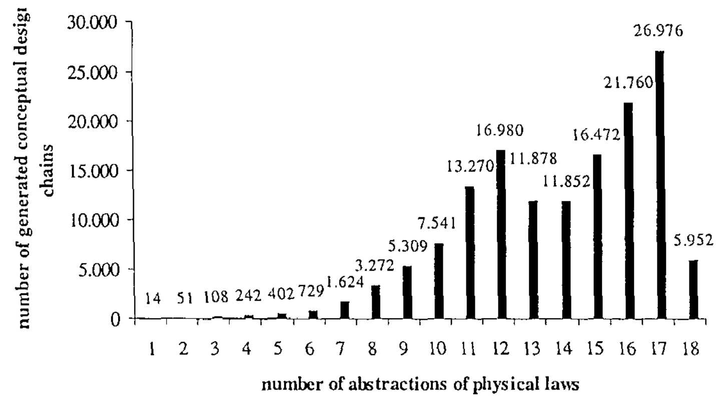


Figure 8. Histogram of a family of design chains (sum = 144.732) for voltage as output variable (unknown input/known output pattern; basic algorithm)

The database of abstractions of physical laws is created on the basis of a set of known physical laws (and complementary basic schematics), and not only those extracted from the technologically available components – the majority of known systems is also limited to so extracted physical laws and complementary schematics. Complementary basic schematics are essentially schematics of components, which enable the embodiment of individual physical laws also with the use of natural components, such as light ray, water drop, liquid level etc. The same applies to components whose planned use is only one of several physically possible uses (e.g. metal spring (planned use: accumulation of potential energy) also conducts electricity (physically possible)). These two facts enable increased innovativity of designed technical systems. Designing at the level of physical laws also prevents a designer's fixation on adaptations of the existing solutions or composition of solutions only from the existing components.

4 Conclusions

The basic algorithm for chaining physical laws was supplemented with an additional requirement, which prevents repeated use of the same physical laws. This limitation makes sense, since repeated use of the same physical law would mean that one and the same function is fulfilled several times. Using a computer program through which the algorithm is implemented, and various sizes of sets of physical law abstractions (30–139 physical laws), it was empirically proved that the combinatorial explosion does not occur. However, the assumption of a risk of a combinatorial explosion occurring is reasonable, since chaining is in essence a search problem, which is characterised by combinatorial complexity. It was also established that a supplemented algorithm reduces the number of generated chains, which strongly facilitates later evaluation of the generated conceptual design chains.

Conceptual design at the level of physical laws enables greater innovativity, which is a result of a rapid generation of new combinations due to the use of an algorithm, especially if known input/unknown output and unknown input/known output patterns are used. Greater innovativity is also enabled by the use of physical laws (and the complementary basic schematics), which are not extracted merely from technologically available components.

On the basis of our results it can be concluded that the algorithm provides an appropriate basis for the conceptual design of technical systems (taking into account the Koller's reference set of physical laws).

References

[1] Giancoli, D.C., "Physics for Scientists and Engineers with Modern Physics", Second Edition, Prentice Hall International, New Jersey, 1988.

[2] Žavbi, R., Duhovnik, J., "Prescriptive model with explicit use of physical laws", Proc. ICED '97, Vol. 2, Tampere, 1997, pp.37-44.

[3] Žavbi, R., Duhovnik, J., "Conceptual design of technical systems using functions and physical laws", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, AI EDAM, Vol. 14, 2000, p69-83.

[4] Ulrich, K. T., Seering, W.P., "Synthesis of Schematic Descriptions in Mechanical Design", Research in Engineering Design, Vol. 1, No. 1, 1989, p3-18.

[5] Chakrabarti, A., Johnson, A., Kiriyama, T., "An approach to automated synthesis of solution principles for micro-sensor designs", Proc. ICED '97, Vol. 2, Tampere, 1997, pp. 125-128.

[6] Welch, R.V., Dixon, J.R., "Guiding Conceptual Design Through Behavioral Reasoning, Research in Engineering Design, Vol. 6, No.3, 1994, p169-188.

[7] Bracewell, R. H., Bradley, D. A., Chaplin, R. V., Langdon, P. M. and Sharpe, J. E. E., "Schemebuilder: A design aid for the conceptual design stages of product design", Proc. ICED '93, Den Haag, 1993, pp. 1311-1318.

[8] Sushkov, V.V., Alberts, L.K., Mars, N.J.I., "Innovative design based on sharable physical knowledge", Proc. AID'96, Stanford, 1996, pp. 723-742.

[9] Bratko, I., Innovative Design as Learning from Examples, Proc. DMMI 93', Bled, pp. 355-362.

[10] Koller, R., Kastrup, N., "Prinziploesungen zur Konstruktion technischer Produkte", Springer-Verlag, Berlin, 1994.

Dr. Roman Žavbi
University of Ljubljana, Faculty of Mechanical Engineering, CAD Laboratory, Aškerčeva 6, SI-1000 Ljubljana, Slovenia, Tel: + 386 1 4771 436, Fax: + 386 1 2527 232, E-mail: roman.zavbi@lecad.uni-lj.si, URL: http://www.lecad.uni-lj.si

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

QUANTITATIVE VISUALIZATION AND EVALUATION OF THOUGHT PROCESSES BASED ON IMPORTANCE FACTORS CONNECTIVITY MAP IN CHECKING A DETAIL DRAWING

S Warisawa, R M Willings, and Y Ito

Keywords: Process modelling, descriptive models of the design process, behaviour of designers, connectivity of importance factors in checking a drawing

1 Introduction

In line with diversification and complexity of user needs and also with very short product life cycle, automatisation of design and manufacturing has been advanced in recent years. It is a fact, however, that the vast majority of techniques and tools developed and used in design research are only to be as a computer aided drawing or a computer aided analysis, not to be as a system supporting design "processes" in relation with designer's thought processes.

Research into human behaviour, both as individuals and as groups has developed in recent years from its foundations in psychology and sociology. Notable among efforts from the engineering community has been work at Cambridge [1] and Munich [2] into the classification of the behaviour of designers in the early stages of creative design. Work at the Tokyo Institute of Technology has made use of graphical representations of design thought process [3][4] and statistical techniques in the evaluation of process planning methods.

However, it has not previously been attempted to put these in the context of designer characteristics and hence understand the variation in approach to design [5]. Furthermore, based on the context, there haven't seen any proposals to visualise the design thought processes, evaluate them in a quantified fashion and also remodel representative processes in terms of designer's classification such as experience, nationality and others in consideration of the variation.

In this paper, three main objectives are hung out. One is to relate the characteristics of a designer to the characteristics of their design process. The influence of experience and nationality are to be sought in this process based on structured interview style investigation conducted machine tool companies in Japan, Germany and the United Kingdom. The second is to propose a method to visualise and estimate the thought processes of designers in a quantitative fashion. A newly proposed concept, "Importance Factors Connectivity Map" is to display connectivity between importance factors in checking a detail drawing in a three-dimensional space graphically and quantitatively. The last is to reveal similarities and differences of thought processes developed based on the proposed method in terms of experience and nationality.

2 Experimental method

The investigation took components of a machine tool spindle system as a case study and focused on engineers who had design experience specifically related to the machine tool industry. The exercise was conducted with a total of 38 designers from 13 companies in Japan, Germany and the United Kingdom. Of these, 7 had less than 5 years experience, 10 had between 5 and 10 years experience and 21 had over 10 years. 20 are Japanese, 10 are Germany and 8 are the United Kingdom. Instructions were provided that in the native language of the subject, but the same drawings, containing Japanese text, were used in all cases. The part drawings used are (i) internal spindle, (ii) external spindle bearing collar, (iii) pulley fixing nut, (iv) quill and (v) external spindle bearing retainer.

The exercise consists of the following activities.

1. Proposal and ordering of important points in the checking procedure without viewing drawings.
2. Presentation in the form of *Directed Graphs* of checking procedures when viewing detail drawings.

A *Directed Graph* is a graphical method used to represent concepts (vertices) and directional links (arrows) between them and is used here as a format, in which the designers are asked to visualise their thought processes. The time frame and details of the response format were left open so as to impose the minimum of artificial constraints and to encourage the designer to behave as naturally as possible. In practice the exercise lasted between one and two hours in the majority of cases. Notes were made of the behaviour of each of the subjects during the exercise.

The results were received in the form of hand-written notes on the prepared response blanks. In order to provide meaningful comparisons, different phrases with the same meaning were identified and all the data were translated into English, and presented on common sheets coded for easy identification of designer characteristics.

3 Characteristics of the response data without viewing a drawing

3.1 Difference of the most important factor by the designer's experience

As a result of aggregating the response data by the designer's experience without viewing a drawing, the most important factors in checking a drawing are clearly observed to be different by experience.

- Low experienced designers: Drawing indication, Size
- Middle experienced designers: Tolerance and fitting
- High experienced designers: Function and purpose, Material

It can be said that low experienced designers are apt to pay much attention to the information which may be found written directly on the drawing. On the contrary, high experienced designers are apt to pay much attention to the information which is not specifically contained within the drawing but is implied by the drawing.

3.2 Difference of the most important factor by the designer's nationality

Despite the fact that the exercise sample was unbalanced in terms of nationality and also that some notifications and identifications on the provided drawings are written in Japanese, we may make some observations as a result of aggregating the response data by the designer's nationality without viewing a drawing.

- Japanese designers: Tolerance and fitting
- English designers: Tolerance and fitting
- German designers: Function and purpose

Nothing can be concluded here regarding the difference.

3.3 Difference of the thought process pattern by the designer's experience

The directed graphs given by the designers are analysed by shape. Our study of the shape of the directed graphs suggests that those can be classified into six patterns (Fig.1): (a) linear, (b) simple branching, (c) simple converging, (d) branching and converging, (e) branching, converging and looping, (f) distributed. It is observed that the patterns become simpler and more linear with experience. Furthermore, the number of importance factors to be checked by low and high experienced designers is found to be smaller by observing each answer sheet. It suggests that low experienced designers have generally few importance factors because of their less knowledge. It is also explained that high experienced designers synthesise their more knowledge into compact one.

One category of designers is the linear pattern that suggests a structured but linear flow of ideas. This is the characteristic of more experienced designers. Another category of designers is the distributed pattern that demonstrates no consistency in approach. We differentiate this category, the flair-based designer who truly uses no structure, from the highly experienced designer who knows subliminally the likely problems to search for without any apparent procedure.

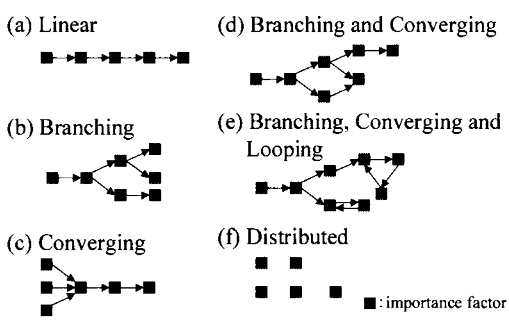


Figure 1. Patterns of thought processes

4 Characteristics of the response data with viewing a drawing

4.1 A concept of "Importance factors connectivity map"

Directed graphs provided by the designers generally have so complicated patterns and so many nodes that it is very difficult to identify their characteristics and similarity. Then it is hypothesised that a connection between importance factors would represent (1) a technological relation between factors and also (2) a local pattern of a designer's thought process. Supposing that there are two connections, A to B and B to C, a local thought process could be structured as A -> B -> C. And thus, when all connections between two importance factors are statistically processed and the frequency of each connection is estimated as a connectivity strength between the two factors, a representative connection pattern could be extracted.

In the Fig.2, X and Y-axis represent importance factors and Z-axis represents connectivity ratio or connection frequency between two factors. We call the three-dimensional space as "Importance Factors Connectivity Map". Differences of connectivity represented by its contour map can be made clearer than those by the directed graph representations.

When selecting connections with a large connectivity and tracing them on the contour map, a representative thought process could be modelled as shown in Fig.2. Each connectivity has a value representing a connectivity strength. So the designer's thought processes can be handled quantitatively as well as qualitatively. Visualisation of the thought process can be realised easily by some computer algorithm and their similarity can be also proposed easily.

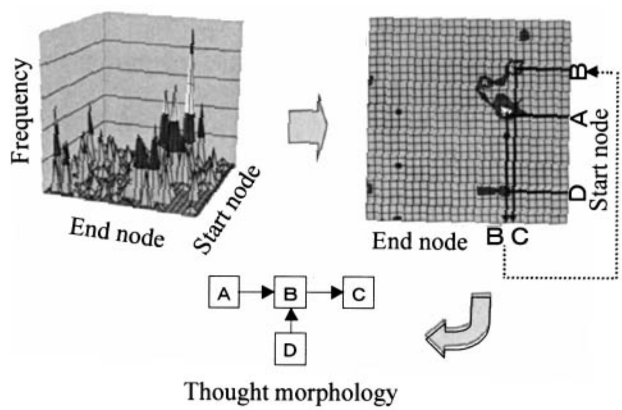


Figure 2. A concept of "Importance factors connectivity map" and representation of thought processes

Table 1. Importance factors

1. Drawing indication	2. Drawing name	3. Cross section view	4. Detailed drawing	5. Size	
6. Overall length	7. Detailed length	8. Overall shape	9. Detailed shape	10. Taper shape	
11. Key way shape	12. Hole shape	13. Screw shape	14. Chamfer	15. Function	16. Machining
17. Precision	18. Finish	19. Tolerance	20. Heat treatment	21. Material	22. Rigidity
23. Weight	24. Cost	25. Assembly	26. Notification	27. Others	

4.2 Visualisation of the thought processes between three nationalities

Figure 3 shows an importance factors connectivity map among three nationalities. The numbers on X and Y-axes indicate importance factors as listed in Table 1. The thought patterns between three nationalities can be visualised based on the connectivity map concept as shown in Fig.4. Despite the different nationalities, there can be found a common thought pattern as from "material" to "heat treatment". The differences and characteristics between the three nationalities are depicted as follows:

- [Germany] : It is obvious that German thought process has a clear linearity. A strong connectivity from "tolerance" to "machining" can be observed. Such link cannot be found

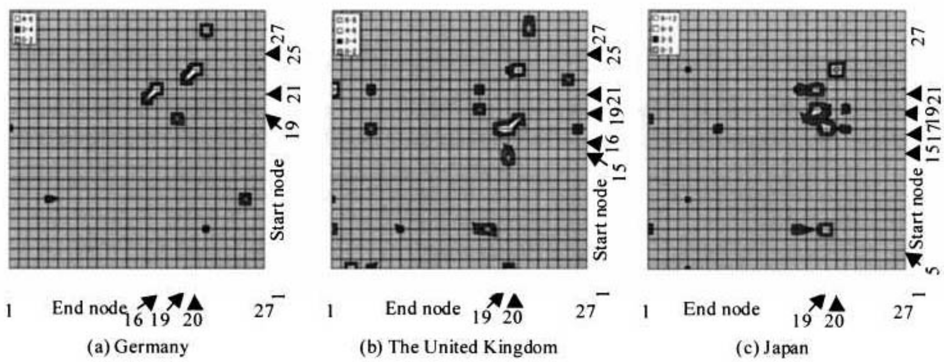


Figure 3. Importance factors connectivity map between three nationalities

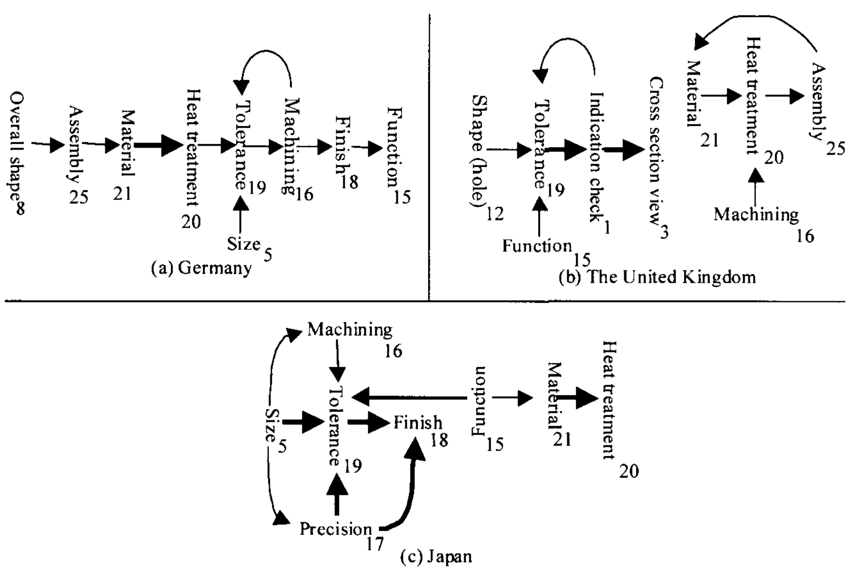


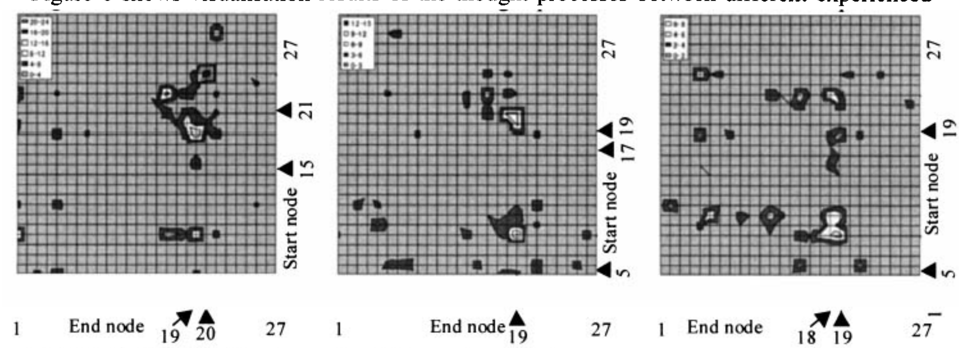
Figure 4. Thought pattern between three nationalities

in Japanese and English designers' thought processes.

- [The United Kingdom] : There can be found two main thought flows. One is to check "tolerance" and is based on the information shown on the drawing. The other is to check "machining" and "assembly" which are not explicitly shown on the drawing. There is a strong connectivity from "tolerance" to "indication check" or "cross section view".
- [Japan] : There are many patterns having "finish" as an end node. It is observed that Japanese designers pay much attention to factors such as "tolerance" and "precision" factors related to "finish".

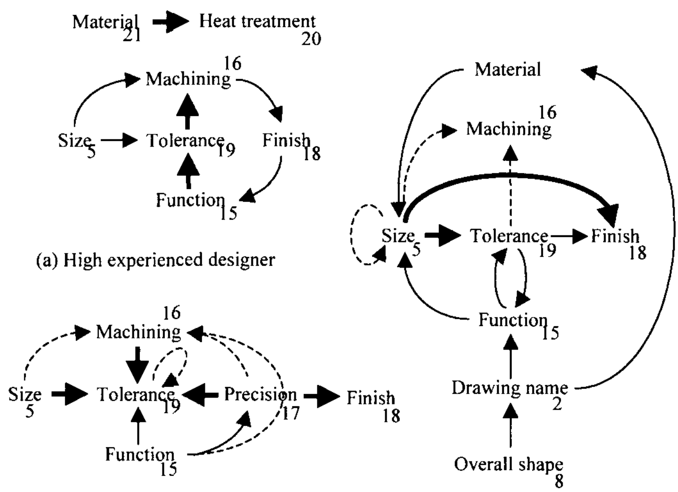
4.3 Visualisation of the thought processes between different experience levels

Figure 5 shows importance factors connectivity maps between different experienced levels. Figure 6 shows visualisation results of the thought processes between different experienced



(a) High experienced designer (b) Middle experienced designer (c) Low experienced designer

Figure 5 Importance factors connectivity map among three experience levels



(a) High experienced designer

(b) Middle experienced designer (c) Low experienced designer

Figure 6. Thought pattern between different skill levels

levels based on Fig.5. A thick line means a strong connectivity between nodes and a dashed line means a weak connectivity.

First of all, there can be found a similarity between the thought processes with viewing a drawing and that without a drawing. Namely, low experienced designers are apt to pay much attention to the information explicitly written on the drawing as mentioned in 3.1. In Fig. 6(c), main factors are "size", "tolerance" and "finish". Furthermore, many factors are linking with "size" factor. On the contrary, high experienced designers are checking factors not shown explicitly on the drawing such as "heat treatment" and "machining" which are linked strongly with other factors.

Second, although the same factor is listed as important, designers with different experience level link different factors with it. For example, in Figure 6, important factors linking with "tolerance" are as follows:

- [Low experienced] : "size" (explicitly shown on a drawing)
- [Middle experienced] : "size" , "precision" and "machining"
- [High experienced] : "function" (not explicitly shown on a drawing)

Third, because of the drawings for components of a machine tool spindle system, "tolerance" is a common main factor in the thought processes of all designers.

By the way, in the section 3.2, it was mentioned the number of importance factors to be checked by low experienced designers is smaller. However, as shown in Figs.5(c) and 6(c), there are more factors and nodes listed by low experienced designers than those checked by middle or high experienced designers. To explain this inconsistency, we propose a concept named as "Design Perception". Without viewing a drawing, low experienced designers cannot touch their memory in their head successfully. With viewing a drawing, however, they can associate their design knowledge by perceiving more design information on the drawing.

5 Conclusions

In this paper, to give the cue for development of a design tool to support design processes which are different among designers who may have different thought patterns,

]In order to model a designer's thought process, "Importance Factors Connectivity Map" has been newly proposed. Based on the concept, designer's thought processes can be visualised and modelled in a quantitative fashion. Taking a checking procedure of a detail drawing of spindle system as a case study, the designers' thought processes have been visualised and estimated so that some findings have been acquired very clearly.

Extensions of the research based on the proposed concept should reveal designer's thought process and contribute to development of "real" computer aided design environments. In this sense, the understanding of the thought processes involved in design and the behavioural characteristics of designers that has been investigated in this paper is very valuable. Because it should give the cue for development of a design tool to support design processes which are different among designers who may have different thought patterns.

For example, the design tool will be able to guess how the designer will behave at the next step corresponding to his experience, nationality and others. Then, the design tool can provide necessary information to the designer during design processes. That should improve the ease of use corresponding to the designer's characteristics. For another example, if the design tool tracks the designer's behaviour by means of eye mark tracking system, audio and visual recording system as well as capturing the actions on the design tool, then the design tool can learn how the designer's thought pattern in the connectivity map proposed.

The proposed concept and other concrete results regarding designer's thought process should contribute to opening the door to our final research goal of giving the cue for the design tool development,.

References

[1] Dwarakanath, S. and Wallace, K. M., "Decision-making in engineering design: Observation from design experiments", Journal of Engineering Design, 6-3, 1995, p191-206

[2] Ehrlenspiel, K. and Dylla, N., "Experimiental vinvestigation of designer's thinking methods and design procedures", Journal of Engineering Design, Journal of Engineering Design, 4-3, 1993, p201-212

[3] Ito, Y., Shinno, H. and Nakanishi, S., "Designer's thinking pattern in the basic layout design procedure of machine tools - Validity evaluation of thinking block", Annals of the CIRP, 38-1, 1989, p141-144

[4] Ihara, T., Ogawa, M. and Ito, Y., "An evaluation method of complexity of difficult-to-machine portions for CAPP based on flair", Journal of Japan Society for Precision Engineering, 59-3, 1993, p435-440

[5] Ullman, D.G., Dietterich, T.G. and Stauffer, L.A., "A Model of the Mechanical Design Process Based on Empirical Data", AI EDAM, 2-1, 1988, p33-52.

Dr. Eng. Shin'ichi WARISWAWA
Department of Engineering Synthesis, Graduate School of Engineering The University of Tokyo, Hongo 7-3-1 Bunkyo-ku Tokyo 113-8656, Japan, Tel: +81-3-5841-6014, Fax: +81-3-5841-6014, E-mail: warisawa@nml.t.u-tokyo.ac.jp, URL http://www.nml.t.u-tokyo.ac.jp

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

FUNCTIONS REVISITED

V Hubka and W E Eder

Keywords: theory of technical systems, design science, design process, function structure

1. Introduction

Many forms of methods-based systematic designing employ functions and function structures. Nevertheless, there is much confusion and misunderstanding about definitions, usage and utility of functions and function structures. Clarification should help to make designing more rational.

According to Design Science [1–8], systems may be divided into *process* systems and *technical* (real) systems (TS). The technical (real) systems are normally the main operators of the process systems. In designing a technical system (mechanical, electrical, chemical, architectural, societal, etc.), whether it is a radical new design or a redesign (variant, revision, etc.), at some critical stage in the process the functions of that system will normally be considered.

The primary condition for obtaining a near-optimal solution to any design problem is that the solution field is searched as widely as possible (within the constraints of time and cost for designing) to establish the alternative candidates. If this search can be accomplished at various levels of abstraction, a closer approach to optimality can usually be made. The minimum number of levels proposed for novel design problems in Design Science [1–8] is: (1) requirements and constraints in functional, economic, societal and other aspects, functionality, (2) transformation processes and operations, (3) technology of operating a process, (4) TS-internal functions, (5) organs or solutions-in-principle, (6) preliminary constructional parts in layout, (7) definitive/dimensional constructional parts in layout, (8) details and parts lists. This may be compared to the notional two levels of 'functional requirements' (FRs) and 'design parameters (DPs) proposed by Suh [9]. Each of the steps in designing should allow generation of alternative solution proposals, a sequence of *goals–means* searches. They are, in this form, an intermediate step in designing that allow physical means – constructional parts, connection regions, action zones, assembly groups, etc. – to be allocated and arranged. They cannot be traversed in such an apparently linear fashion, but must be *iteratively* established.

Searching for alternatives in this way would soon lead to *combinatorial complexity*, the numbers of possible combinations would become far too large for human comprehension. Control of combinatorial complexity is achievable by evaluating all the alternatives at each level of abstraction, selecting only the (one or two) most promising, and continuing the search at the next more concrete level. The discarded alternatives should be kept in the records, in case (a) difficulties require back-tracking to a different alternative, or (b) a technical development renders a sufficient improvement to a different alternative to change the order of preference.

2. Functions

Functions (as defined in Design Science [1–8]) describe the required or desired (internal and cross-boundary) *capabilities* of a (future) real system that (will) make it possible for that system to perform its (external) duties, its intended goal tasks. Some of these functions will describe inputs and outputs of the real system, the tasks of the receptors and effectors. Other functions will describe how the desired and secondary inputs may be (or are) transformed within the real system into the desired and secondary outputs. Functions are normally formulated in words, as a combination of a verb (or verb phrase) and a noun (or noun phrase) – similar to the usage in

Value Engineering, but with a different, more precise definition of 'function'. The formulation is normally chosen by the designer to allow several candidate solutions to be proposed – searched for by reviewing existing systems and literature, and inventive imagination using the human capability of mental association. Functions in these formulations can be combined (and abstracted) or sub-divided (decomposed and concretized), which may lead to further alternative candidate solutions to the design problem. As an example, a spindle with 6 degrees of freedom is brought into the state of one degree of freedom (rotation) by the function 'hold the spindle in bearings which can react radial and axial forces'.

The function structure is defined by a set of its elements (functions) and a set of relationships of these functions to one another. The function structure gives the engineer a means to evaluate the operational states of a (future or existing) technical system (TS).

The *capability of using a function* is a property of the technical system, class 1 in the classes of properties defined in Design Science [1–8], figure 1. A function describes the ability of a TS to fulfill a purpose, namely to convert an input measure into a required output measure under precisely given conditions. The TS function can be understood as a unique (but usually not 1:1) coupling of the elements of a set of *independent* input measure to the elements of a set of *output* measure. This interpretation shows some similarity to the concept of a mathematical *transfer function* as defined for dynamic systems.

Functions in this definition, as one set of the essential properties of technical systems, are absolutely dependent on the design characteristics of the system being designed, listed in the properties of technical systems as class 12 in the Design Science scheme, figure 1. Functions are used as *elements*, and together with their *relationships* (mainly connections, couplings among

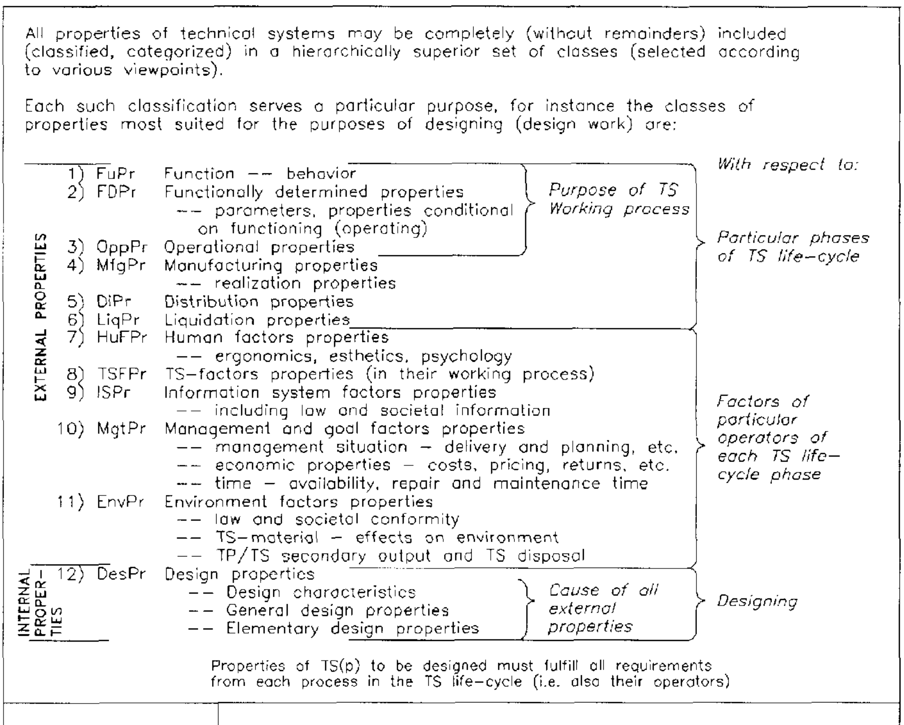


Figure 1 Classes of Properties of Technical Systems

individual functions) form a *structure* that should describe (at an abstract level) the capabilities for action of a technical system. The functions as formulated should 'scan', i.e. the input to any one function should be equivalent to the output from its preceding function(s), and should form a 'chaining' with a reasonably logical flow of materials, energy and information from TS-inputs (receptors) to TS-outputs (effectors). Functions may be arranged in series or in parallel, and may show branching or combining of paths.

As a solution proposal is developed through these (abstract to concrete) steps, several additional problems arise – they in turn need to be solved through (for that design problem) newly discovered (*evoked*) functions, organs and constructional parts.

The number of steps, iterations, sub-divisions, etc. that are used in an actual design sequence depends on the complexity and difficulty of the task. A design task may be sub-divided into smaller, more easily solved problems, and solution candidates re-combined at a more concrete level – a *recursive* working mode. The different sub-problems are usually at different states of concretization – which leads to an apparent confusion of steps to a casual observer. Only by searching for alternatives at each (abstract or concrete) step can designers be sufficiently certain of obtaining a near-optimal solution – which may need to be (mathematically) optimized for its performance.

There are two main ways to work out a function structure. One way starts by concretizing from a 'TS black box', and implies that the functions and their relationships are to be found or *synthesized*, as the means by which the output effects that will drive the technical process (as the aims) can be realized. This path leads through establishing the inputs and modes of action of the TS, and can take place at the various levels of abstraction listed above, and achieve differing levels of completeness, as explained above.

The second way starts by abstracting from an existing constructional or organ structure, and is *analytical*. This way is more useful for redesigning an existing system. It results in either a very comprehensive function structure with many types of function (resolution or decomposition), or, by neglecting those functions that are not essential for the direct transformation process, a function structure similar to the one established from the 'black box' can be obtained.

Good and complete consideration of functions (and consequently of organs and of constructional parts) during designing is frequently decisive for the quality of the resulting system. Therefore the types of possible functions should be defined, to provide a check-list for designers to use in verifying that they have considered all necessary functions in their design process.

3. Classification of Functions

Many different kinds and classifications of functions appear in the literature. Here we will only define a few basically important types (as used in Design Science [1–8]) which will be used in the further discussions. Of the many classifying viewpoints, three are particularly meaningful:

(a) ***Complexity of the function***. Each function may be assigned to a certain degree of complexity in a hierarchy of complexities (several levels from 'most complex' to 'simplest'). The lowest degree is occupied by the elementary functions, those that cannot (usefully) be resolved into more limited functions.

(b) ***Degree of abstraction of the function***. Each function may be described at various levels between 'concrete' and 'abstract'. This in turn influences the number of possible or available organs (function carriers) that can be found (as means) to realize the function (as the goal). If, for instance, the given description refers to the function 'change motion of ...', then the range of available means is very broad. With an increased number of additional data about the function, selected from the ranges of effects, conditions, operations or working means, the range of available means of fulfilling a function is progressively narrowed, until a single concrete TS remains. The additional data mentioned here are the design properties, properties class 12 in the Design Science scheme, figure 1.

The degree of abstraction is derived from the design properties for the given ranges. Functions may be designated as 'functions with ***i*** conditions'. The order in which these conditions appear is not determinate, the behavior of a real TS can also be attained by a different ordering. For systematic work in engineering design it is essential that a certain ordering arrangement is selected and agreed, and used consistently.

The connection between the degree of abstraction and the degree of complexity should also be noted. Resolving the functions into partial functions (i.e. functions of lower degree of complexity) is only possible and sensible when a certain more concrete level of abstraction has been obtained. In our recommended procedural scheme this is true only when an action principle and/or a mode of action has been established. Designers who are familiar with systematic design methods will recognize an analogy to the morphological chart [4,10], in which the path of progress leads from a function (preferably starting with the most important), by considering one or more action principles and/or mode of action, to the organs (function carriers) able to fulfill that function. Others of these characteristics are unconsciously neglected, either because they are implicit in the statement of the task, or they are assumed by tradition to be 'fixed variables'.

(c) ***Categories of purpose of the functions***. The various sorts (classes and categories) of functions that can be useful in designing a system include (see figure 2):

(1) the *purpose function* (PuFu) of the proposed system, its essential capabilities, which include those outputs that are needed from the technical system, the effects it applies, the chosen technology, and the technical process (changes in the operands in the transformation process under the influence of the technical systems) – this is a composite. For systematic designing of a novel system, it is preferable to separate these elements, and to consider each on its own: first the transformation process, then the technologies, then the needed effects, etc., with possibilities for variation and search for alternatives at each step;

(2) the *working effects* (WEf) as intended output via the technology to the operand;

(3) the *working functions* (WFu) internal to the TS as means to fulfill the purpose;

The working functions and working effects, with the selected mode of action, that fulfill the purpose of the TS are necessarily accompanied by a series of additional functions. These are essential, and serve to ensure that the purpose function can be realized, or its operation supported. These additional functions are:

(4) *assisting functions* (AssFu) that allow the working functions to fulfill their tasks, which may further be sub-divided into:

(4a) the *auxiliary functions* (AuFu) that deliver assisting materials within the technical system;

(4b) the *propelling functions* (Pfu) that deliver energy;

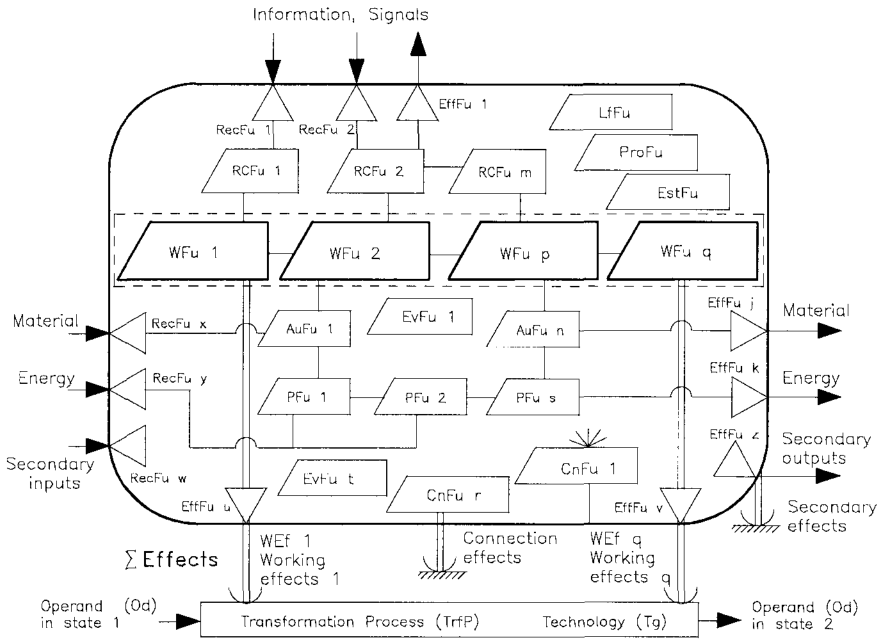
(4c) the *regulating and controlling* (RCFu) functions that deliver information (data, commands, etc.), perform measurements and comparisons with standard quantities, and give feedback for use within the system; and

(4d) the *connecting and supporting functions* (CnFu) that keep the technical system together in one unit and connect it to the fixed system;

(5) functions that provide the connections across the boundary – *receptor functions* (RecFu) and *effector functions* (EffFu)

(6) the *life functions* (LfFu), properties-generating functions, e.g. those that permit life-cycle actions and achieve reliability or stability of the system;

(7) *production functions* (ProFu) that permit manufacture and assembly (and concurrent engineering of the product and its manufacturing processes), but also disassembly for maintenance and disposal (e.g. recycling) of the system – these are mostly evoked



PuFu ... Purpose functions (active and passive) describe the combined working effects, the main technical process (transforming the operands) and the technology to transform the operand
WEf ... Working effects needed to implement, drive, operate the technical processes
WFu ... Working functions (active and passive) needed to generate the working effects
Assisting Functions (active and passive) needed to directly support the working functions, including:
- AuFu ... Auxiliary functions
- PFu ... Propelling functions (energy converting and delivering)
- RCFu ... Regulating and controlling functions
- CnFu ... Supporting and connecting functions, TS–internal and to external (fixed or environment) systems

Trans–boundary Functions:
- EffFu ... Effector functions
- RecFu ... Receptor functions

Evoked Functions, needs for properties recognized during designing:
- LfFu ... Life functions needed to support cleaning, storing, transporting, safety, reliability, maintainability, etc. of the technical system
- ProFu ... Constructional functions needed to enable manufacturing, assembling, testing, adjusting, servicing, commissioning, etc.

EstFu ... Esteem and other functions to enhance human, social, economic and legal acceptance.

Figure 2	Complete Model of Function Structure of a Technical System

functions that are discovered during designing in the more concrete structures (organ, or constructional).

(8) esteem functions (EstFu) that enhance human, societal, cultural, economic, legal and other acceptance of the completed TS.

These categories of function can act as check-lists to verify that the considerations during designing have been as complete as possible, leading to 'right-first-time' designing.

During designing, the primary considerations are establishing the means to fulfill the working functions, as main means to achieve the working effects, and thus the purpose functions. Whilst these considerations progress, especially during the process of establishing the organ and constructional structures (i.e. in arrangement and preliminary layout) the assisting and other functions are successively brought into the considerations to bring the solution proposals towards completion.

If for a particular family of technical systems the formulations of functions can be standardized, computer assistance (e.g. in the form of petri-nets) can be developed to search for known solutions. Analysis for redesigning an existing system may not need to reach the abstraction of the function structure, unless a substantial degree of innovation is required.

Figure 2 shows a representation of the function structure as a 'block schematic' that shows the important relationships as connecting lines, and conforms to the general arrangement of elements of the transformation system. A typical arrangement of the functions within such a function structure should reflect the general model of function structures, see figure 2 (and figure 8–4 in [4]) and should include the assisting (auxiliary, propelling, regulating, supporting and other evoked) functions, and the receptor and effector functions at the TS boundary.

In designing, it is probably best to only include the more important functions, i.e. the working functions and the main assisting functions, completeness may not be necessary.

This representation is by no means the only possible form. Many examples of other forms are described in the literature [11,12,13,14,15]. Especially noteworthy is the form used for the 'general function structure' according to Roth [12]. It uses the 'general functions' as defined below and can be regarded as a form of logic flow or switching schematic.

Another form that is similar to the function structure is that of the 'hierarchical function tree' [13,14,15]. It shows the functions with only their immediate dependent relationships, but cannot easily show all other relationships, especially between functions in different branches of the tree structure. Nevertheless, because it is a clear and simple representation, it finds frequent use. Examples are given in the literature, for instance figure 8–5 in [4].

4. Other Definitions

Apart from the three types of functions discussed in sub-sections (a), (b) and (c), a few other terms that frequently occur in the literature should be defined. They are not a part of the Theory of Technical Systems [2], but show a connection to other theories and methodologies of engineering design.

(d) *Logical function*. This type transforms one or more independent variables into a single dependent variable that can only take on two measures (e.g. 1 and 0, + and -, or true and false).

(e) *General function* [12]. This is an elementary function that results from coupling a general operation (storing, conducting, transforming, translating, and summative or distributive associating) with a general 'value type' (material, energy, information).

(f) *Basic function* [16] is an elementary function (combining, dividing, conducting).

(g) *Physical elementary function* (or basic operation) [17]. This term has been applied to 12 elementary functions: emitting, conducting, collecting, guiding, transforming, enlarging, direction-changing, directing, coupling, connecting, adding and storing.

The functions defined in (e) and (f) serve to establish the general function structure that contains the elementary functions standardized for that particular method. These elementary functions are based on the idea that only the three flows of material, energy and information can be present in technical systems, and that only certain operations (basic operations) are permitted.

Note that the term 'total function' is not included in this list, it is more related to an organ structure. The VDI Guidelines [13,14,15] quote mainly the purpose functions (with all their ambiguity), but also use the terms 'primary/main functions' and 'secondary functions'. These latter may be regarded as loosely equivalent to the Design Science forms of 'purpose, working effects, technology and working functions', and 'assisting, evoked, life and constructional functions'.

In a hierarchy of functions, the ways in which a higher function of a technical system can be resolved or decomposed into its partial functions should now be investigated, compare [11]. This resolution is not as easy or obvious as examples quoted in the literature might suggest. The classes of partial functions into which higher functions can be resolved are of three types:

(A) partial functions that are determined from, and because of, the selected mode of action;

(B) partial functions that are necessarily evoked by the given function (see categories of purpose of functions);

(C) partial functions that serve to realize other necessary properties of a TS – manufacturability, transportability, safety of humans, etc. (i.e. additional tasks) – such as a 'connection between sub-assemblies' made necessary by reasons of final erection or transportation, or a 'guarding' of moving parts needed to protect humans from danger.

Classes (A) and (B) are established while working out the function structure and/or organ structure. Class (C) is particularly useful during the transition from the organ structure to the constructional structure.

Usually, even at the beginning of a project, a number of the characteristics and properties of the technical system are already firmly established and fixed, typically:

- the effects (as output from the TS) needed to drive an established technology of transformation;
- the mode of action, defined by action sites, action motions and conditions;
- the degree of mechanization or automation in executing the technical process;
- even at times the actual basic mode of action of the TS and its inputs.

The number of these design characteristics that are established in this stage of engineering design, when considering the function structure, depends on the degree of abstraction, and determines the degree of completeness of description for the TS.

5. Closure

Functions are useful additions to any designer's armory. They allow a further stage for searching for alternative solution principles and executions. Identifying the sorts of functions can help to check how far the design task has been completed. The lists and figures in this paper are tools towards that end.

References:

[1] Hubka, V. and Eder, W.E. (1996) **Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge** (London, Springer-Verlag); completely

revised edition of Hubka, V. and Eder, W.E. (1992) **Einführung in die Konstruktionswissenschaft** (Introduction to Design Science) (Berlin, Springer-Verlag)

[2] Hubka, V. and Eder, W.E. (1988) **Theory of Technical Systems** (Berlin/Heidelberg & New York, Springer-Verlag); revised edition of Hubka, V. (1984) **Theorie technischer Systeme** (Theory of Technical Systems) (Berlin, Springer-Verlag) – 2nd edition of Hubka, V. (1974) **Theorie der Maschinensysteme** (Theory of Machine Systems) (Berlin, Springer-Verlag)

[3] Hubka, V. (1976) **Theorie der Konstruktionsprozesse** (Theory of Design Processes) (Berlin, Springer-Verlag)

[4] Hubka, V. and Eder, W.E. (1992) **Engineering Design** (Zürich, Heurista); completely revised 2nd edition of Hubka, V. (1982) **Principles of Engineering Design** (London, Butterworths) translated and edited by W.E. Eder from Hubka, V. (1980) **WDK 1 – Allgemeines Vorgehensmodell des Konstruierens** (General Procedural Model of Designing) (Zürich, Heurista)

[5] Hubka, V., Andreasen, M.M. and Eder, W.E. (1988) **Practical Studies in Systematic Design** (London, Butterworths) – English edition of (1980) **WDK 4 – Fallbeispiele**

[6] Eder, W.E. (ed) (1996) **WDK 24 – EDC – Engineering Design and Creativity** – Proceedings of Workshop EDC (Zürich, Heurista)

[7] Eder, W.E., Hubka, V., Melezinek, A. & Hosnedl, S. (1992) **WDK 21 – ED – Engineering Design Education – Ausbildung der Konstrukteure – Reading** (Zürich: Heurista)

[8] Hubka, V. (1978) **Konstruktionsunterricht an Technischen Hochschulen** (Design Education in Universities) (Konstanz: Leuchtturm Verlag)

[9] Suh, N.P. (1989) **Principles of Design** (Oxford, University Press)

[10] Zwicky, F. (1948) **The Morphological Method of Analysis and Construction**, Courant Anniv. Vol. (New York: Wiley-Interscience)

[11] Pahl, G., & Beitz, W. (1995) **Engineering Design** (2 ed.) (London: Springer-Verlag) (Edited and translated by K. Wallace, L. Blessing and F. Bauert); revised from Pahl, G., & Beitz, W. (1984) **Engineering Design** (London: The Design Council, & Berlin/Heidelberg: Springer); translated by A.J. Pomerans and K.M. Wallace, and edited by K.M. Wallace from Pahl. G. und Beitz, W. (1988) **Konstruktionslehre** (2. Auflage) (Berlin/Heidelberg: Springer-Verlag)

[12] Roth, K.-H. (1994) **Konstruieren mit Konstruktionskatalogen** (Designing with Design Catalogs, 2 ed., 2vols.) (Berlin/Heidelberg: Springer)

[13] – (1993) VDI Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte (Methodology for Develpoing and Designing of Technical Systems and Products) (Düsseldorf: VDI)

[14] – (1997) VDI Richtlinie 2222 (1): Konstruktionsmethodik: Methodisches Entwickeln von Lösungsprinzipien (Design Methodology: Methodical Development of Solution Principles) (Düsseldorf: VDI)

[15] – (1999) VDI Richtlinie 2223 (Entwurf): Methodisches Entwerfen technischer Produkte (Methodical Layout of Technical Products) (Düsseldorf: VDI)

[16] Rodenacker, W.G. (1970) **Methodisches Konstruieren** (Methodical Designing) (Berlin/Heidelberg: Springer)

[17] Koller, R. (1985) **Konstruktionslehre für den Maschinenbau** (Study of Designing for Mechanical Engineering) (Berlin/Heidelberg: Springer-Verlag); 2nd edition of Koller, R. (1979) **Konstruktionsmethode für Maschinen-, Geräte- und Apparatebau** (Design Method for Machines, Appliances and Devices) (Berlin/Heidelberg: Springer)

W. Ernst Eder
Department of Mechanical Engineering, Royal Military College of Canada, P.O. Box 17000 Stn Forces, Kingston, Ontario, Canada K7K 7B4, tel & fax 1-613-382-3810, eder-e@rmc.ca

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

THE APPLICATION OF SEMIOTICS IN THE UNDERSTANDING OF SYNTHESIZING PRODUCT QUALITY IN DESIGN

M W Lange

Keywords: Design for Quality, Perceptual Knowledge, Product Synthesis, Design Semiosis

1 Introduction

Products are designed by somebody to be perceived by somebody. We use our senses of sight, touch, sound, taste and smell to perceive the physical nature of these products. When designers engineer products for or in a corporation they are provided with a quantified description of the product, as a specification. This description includes explicit knowledge in the form of statements and values about how the product will fulfill needs, provide functions and have parameters.

These two states represent two distinct domains of knowledge, the one hand referring to perceptual knowledge and on the other hand relaying explicit knowledge. These two states represent the endpoints of a knowledge range that a designer must be capable of traversing in their effort of product creation. Bridging these endpoints in a structured manner so that the job is performed in an effective manner with the right product quality as the outcome, is an ad hoc process best performed by designers with extensive experience.

This paper will introduce a theoretical development that intends to bridge these two knowledge domains, a theory in which design is treated as an act of semiosis [1]. The objective is to provide a descriptive and prescriptive model of the process of product design, with a particular interest in designing an identity in a product brand. In turn it will be argued that this theory can be applied to the understanding of product.

2 Scope

To limit the theoretical discourse, only the conceptual design of a physical product will be considered in particular the incremental development of existing products. This phase of the product development process is of interest because many of the problem-solving situations that occur in this phase are concerned with the synthesis of a quality that will be perceived by an end-user, particularly in cases where there is a misfit in form[1] [2].

[1] In a treatment of design, C. Alexander considers the synthesis of solutions as responses to misfits in a property or a set of properties, as discussed in [2].

3 Approach

This approach addresses a problem found in the definition of design; the transformation of a set of stated needs into an artifact [3]. This infers that design is process of abstraction from the qualitative to the quantitative. Furthermore, this also implies that there is some form of emergence of the levels of representation in the design process, such as the abstract to the concrete, the general to the specific, etc. Within the domain of design research, there is lacking a philosophical foundation, or theoretical view, which describes the emergence of levels of human-based perceptual knowledge with levels of corporate-based explicit knowledge. Being that all knowledge begins with an understanding engendered through our senses, this would appear to be a logical starting point for a discourse on a theory of design by perception.

3.1 Semiotics and Semiosis

The first point to consider in such a discourse is what is the nature of a mechanism through which perception engenders understanding? An answer to this question can be found in the sign semiotics of Peirce [4] and sign semiosis of Morris [5]. From semiotic viewpoint, Peirce has proposed that the basic mechanism of understanding is the "sign". This sign is a triadic element composed first of a sense-able sign vehicle (the **R**epresentamen), secondly an objectification made of the sense-able sign vehicle (thus making it an **O**bject), and thirdly a sign in the mind of the user of the sign (as the **I**nterpretant).

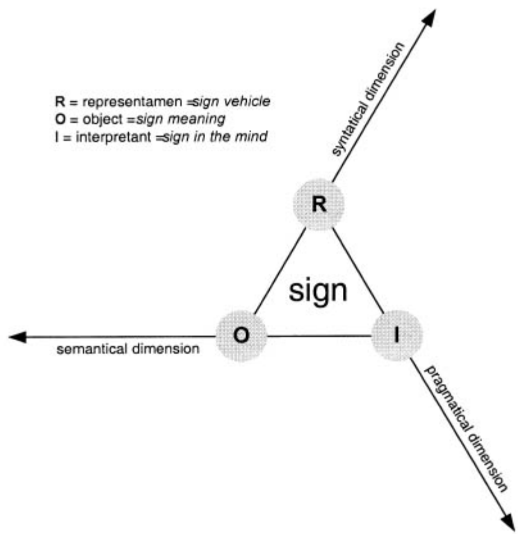


Figure 1. The Space of Sign Semiosis, as a model of a theoretical foundation for studying product design and synthesis.

Morris further developed this concept so that the three aspects corresponded to three dimensions of semiosis, which is the synthesis of signs in the communication of understanding. The first dimension is syntactic and is concerned with the relations between a given sign vehicle and other sign vehicles. The second dimension is the semantic and addresses the use of different sign vehicles in a specific context. The third dimension is the pragmatic and reflects the state of understanding obtained through the use of a sign vehicle.

Within the domain of research in product design, semiotics as a foundation for investigation is first found in the discipline of Industrial design. An example of this application is found in Monö's definition of the *current product sign*' where the gestalt of the product's form is the means of objectifying the function of a product [6]. This construct is be further developed by Warell into a generic set of industrial design elements [7]. This research intends to further extend the application of semiotics into the domain of machine design with a perception on product synthesis.

3.2 Signs and Sign Theory

This necessitates the next question, what is a sign in design? In general a sign is simply something that means something to someone. A sign then mediates between what is perceived and what is understood [6]. In this sense, noises, signals, or sign elements are not signs, until they are used in some context to communicate an understanding. A "⊛"or a "" or for that matter the letters H-A-N-D-L-E or P-A-N-H-E-A-D-S-C-R-E-W have no meaning until they are expressed by someone in a particular context for the purpose of communicating something about something.

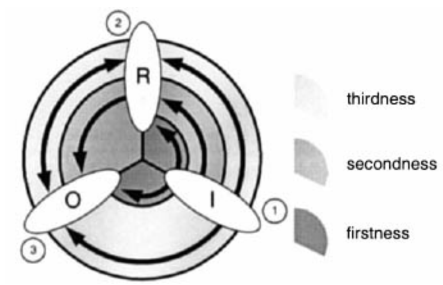


Figure 2. The cyclic nature of semiosis moves through distinct emerging content levels. These levels of sign making are based on Peirce's generic and iconical level of thirdness through to the level of firstnes [8].

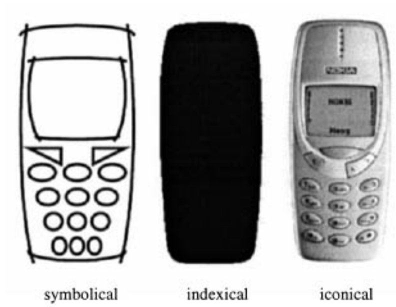


Figure 3. Visual sign vehicles (representamen) of a **mobile telephone** (Object) *concept* (Interpretant), represented as a sketch (symbol), a silhouette (index) and a photograph (icon). Note, these sign vehicles are not the "carriers" of product information, but the stimuli for understanding the objectification **mobile telephone**.

This gives rise to the next question, how is a sign constructed? In order to understand the response to the question, it must be understood that signs have three general form levels that emerge from each other. The first level is the symbolic level (called the level of firstness by Peirce). This is a level in which the intrinsic is perceived, no reference is made to any "second" thing. The next level is the indexical level (the level of secondness). Here are properties that distinguish one thing from another and involves opposition or other-ness [6]. Finally there is the level of the iconic (the level of thirdness). This is the generic semiotic level of a sign, *anything which determines something else to refer to an object to which itself refers in the same way, the interpretant becoming in turn a sign, and so on ad infinitum* [4].

For example the "⊛" is a graphical symbolic element of a plan view of a Phillips head screw. The "" is an image of a button on a mobile telephone keypad. But these objectification's are arbitrary, "⊛" could be correctly objectified as a button on an overcoat and "" could be correctly objectified as a number key on an instrument. This makes symbols difficult to use with any exactness and requires that a succession of other, often symbols, must be applied to enhance understanding.

3.3 Properties Engendering Signs

The approach concerns the design of physical products. Thus the next question is, what is being *objectified* during design? To begin a response, it is the quality of a product that is perceived by a customer and the quality of a product is perceived through its properties [9]. As described by Mørup, quality is built into a product when choices are made about its properties and functions.

The quality properties of a product are then what is being objectified in a process of sign semiosis. It is these properties, or the objectification of a sign vehicle, that a designer must first consider during the synthesis of a solution. This must be done correctly so that the customer correctly perceives the solution that is created. When considering functions, these can not be perceived as signs. What is perceived here is the effect of a function.

So, although a product can be described quantitatively in terms of its transformation processes, functions, organs, and components these same quantified terms can not be qualitatively perceived. In a sense, the qualitative abstraction of the creation of a product serves as an objectification of sense-able sign vehicle. The more abstract the quantification, the more symbolic and consequently arbitrary the meaning communicated of what is to be perceived in the product.

3.4 Specifications as Symbolic Documents

The situation becomes even more complex when a designer is provided with a task specification. Although, annotated with sketches and drawings, the body of such a document is often filled with textual statements that are about the desired product. These statements alone do not constitute a product. There is nothing there that can be perceived.

Although a specification is intended to serve as a guide in the design effort, it is never complete enough to apply without the addition of complementary information, often in symbolic forms. As such a specification never reaches a state of completeness and instead remains often in continuous revision. Each successive revision is updated with further textual descriptions until the product is complete and appears as a reference in the final archived version.

Properly applying a complete sign and not a free standing sign vehicle is the most important issue in a semiotic approach to the design of a product. Even if the quality of a product is perceived through the collection of a variety of different properties. Sign vehicles can easily be misconstrued as sign elements, or signals, neither is a sign unless it means something about something to someone.

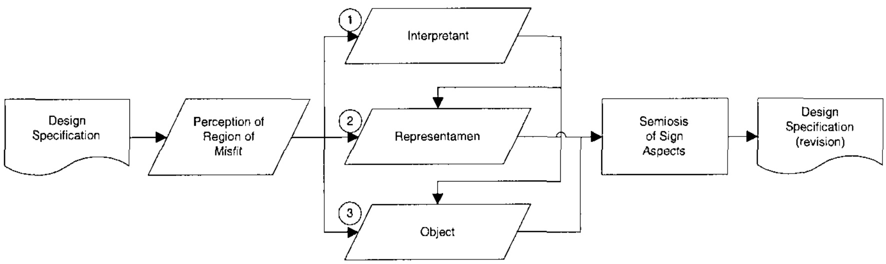


Figure 4 . The Flow of Design Sign Semiosis, in an approach to a specification review.

4 Design Semiosis

The product upon which this approach is applied is a piece of telecommunications equipment. This class of equipment has evolved considerably over the last 15 years. Originally, this equipment was conceived to be handled by personnel educated to understand the technology contained with its workings. As such the interface and usability of the equipment was considered to be secondary to the functional operation. But now with the shortage of trained personnel and the increased proliferation of telecom networks, these secondary considerations are now becoming primary considerations in an effort to reduce the time-for-installation and time-for-service factors associated with telecom equipment installation.

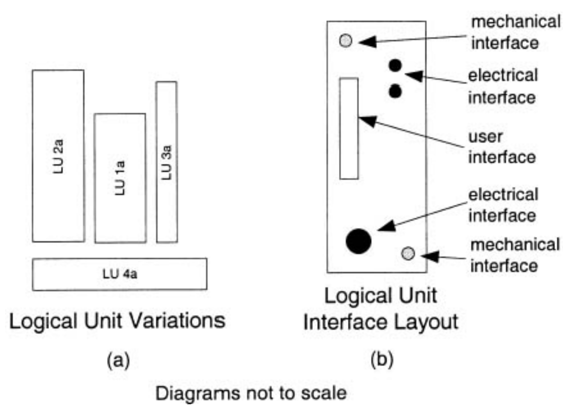


Figure 5. Diagram of Telecommunications Product treated during a Concept Design Activity. From (a) the designer considers intrinsic properties of the logical unit variations such as orientation, height, width, and depth, which are the facts that can be perceived in the gestalt of the artifacts. From (b) each of these units has a generic interface layout that contains different forms of information about the nature of the interface.

This problem background forms the interpretant for the objective of the design task. The objective of the design effort is to distinctly improve on the end-users perception that the product is user friendly. In other words, enhance the interpretant, which is symbolically stated as “user friendly”. This statement, when presented as a requirement, is very pragmatic and fulfillment can be arbitrary and qualitative depending on the level of experience of the individual who is interfacing with the equipment and the individual who is designing the equipment.

The problem under consideration is shown in Figure 5. This figure is a conceptual abstraction of a metaphor (a sign vehicle) of how a designer perceives the region of misfit [2], which is the user interface on the logical units that are installed in the telecommunications equipment cabinet.

This user interface, in its generalized form (shown on the right) appears on a variety of different logical units (shown on the left). In the generalized form, there appear symbolic representations in the form of clarifications in the English language, which are provided to regions in reference. These referenced regions serve as indexes that are related to the symbols through the use of an arrow. This generalized layout can then be referenced to the arrangements of the layout variations.

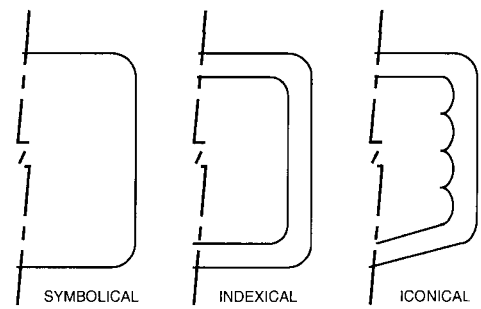


Figure 6. Classes of "Sign Vehicles" for a Handle

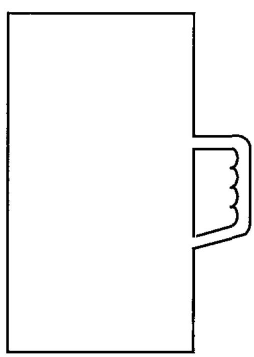


Figure 7. The Symbolical Sign Vehicle for the Designate nature of the "handle" attachment, integrating the handle in the logical unit casting.

Collectively the figure represents the "concept" of the region of misfit. The "factual" content of the region of misfit is placement and identification of the different symbolic interfaces, the mechanical, user, and electrical interfaces. Two of these interfaces were of particular interest to the design team, the user interface and the mechanical interface. In previous generations of the product, the mechanical interface was embodied as a screw, requiring a screwdriver to release the logical unit from the equipment rack. The unit was then removed using the user interface embodied as a handle.

Within the design team, discussions moved from the symbolical to the indexical to the iconical returning again to the symbolical, once the understanding of the iconical was established. For example in the case of the handle design the discussion began with the symbolic sign vehicle shown in Figure 6. When questions were asked about how the solution would be embodied (i.e. how it would be manufactured) the representation moved to the indexical. The visualization of the indexical was based on the previous experience of purchasing a bent-wire handle typically found on electrical cabinets.

Now, a handle could be discerned from the visual sign vehicle. But, it's similarity to the current design was striking, the only difference being that the "handle" was visually separated from the body of the logical unit only permitting the suggestion of an alternative means of manufacturing the component. Enhancing this representation with additional symbolical visualization for "finger grips" and "lifting support" the iconical level of understanding could be reached in the sign vehicle. But nothing could be known about the intrinsic nature of the "handle", such as material, or the "designate" nature, such as attachment solution, or the "sense-able" nature, i.e. attractiveness. Answers to these questions required additional symbolic statements, such as material $\Rightarrow$ aluminum.

The synthesis of signs, with their sign vehicles, interpretants and objectification's continues endlessly until the group is in agreement about how well the proposed concept fulfills the statements found in the design specification. The outcome is then prepared and placed in an additional specification for detail design to be executed at a later stage.

5 Conclusions

It is understood that perception is often considered to be highly subjective and qualitative in nature. How one perceives a physical product is dependent on personal, educational and professional experience. What has been highlighted here is that the design of a solution draws on the experiences engendered by perception. This process is semiotic in nature and moves from the symbolic to the iconic and back again, accounting for a highly iterative nature of design [10]. What also must be considered is that the sign vehicles used in the process of design change form depending on the sense of perception; sight, sound, smell, taste and touch.

Although this perspective on product creation is prescriptive in nature, a more substantial and quantitative description is required in order for the method to be applied in a stringent and structured manner. For this purpose, ten discrete generic levels of semiotic signs have been derived. These levels and their derivation are to be presented in a later publication.

6 Acknowledgements

The author would like to thank the Swedish Engineering Design Research Education Agenda, ENDREA, and Nokia Svenska AB, and the Dept. Of Machine Design at the Royal Institute of Technology, Stockholm Sweden for their support in the research reported in this article.

References

[1] Lange, M.W., *CONCAD Bridging; Supporting Communication between Concept and Detail Design Activities in Product Development and Design*, Licentiate Thesis, Dept. Production Systems, Royal Institute of Technology, Stockholm, Sweden, 2000.

[2] Alexander, C., *Note on the Synthesis of Form*, MIT Press, Cambridge, MA, 1964.

[3] Konda, S., Monarch, I., Sargent, P. and Subrahmanian, E., "Shared Memory in Design: a Unifying Theme for Research and Practice", Research in Engineering Design, 4, 1992, pp. 23-42.

[4] Peirce, C.S., *The Collected Papers of Charles Sanders Peirce*, Harvard University Press, Cambridge, MA, 1931-58.

[5] Morris, C.W., *Foundations of the Theory of Signs*, University Press, Chicago, Il, 1938 (1971).

[6] Monö, R., *Design for Product Understanding: The Aesthetics of Design from a Semiotic Approach*, Liber, Stockholm, 1997.

[7] Warell, A., *Industrial Design Elements; A Theoretical Foundation for Industrial Design Based on a Design Science Perspective.* Licentiate Thesis. Div. Of Machine Design, Dept. of Mechanical Engineering, Linöpings University, Linköping, Sweden. ISBN 91-7219-627-0. 1999.

[8] Ransdell, J., *On Peirce's Conception of the Iconic Sign*, http://www.door.ner/ARISBE/menu/library/aboutcsp/ransdell/iconic.htm, 1997.

[9] Mørup, M., Design For Quality. Doctoral Thesis, Institute for Engineering Design, Technical University of Denmark, Lungby, Denmark, 1993.

[10] Ahmed, S., Blessing, L. and Wallace, K., "The Relationships between Data, Information and Knowledge Based on a Preliminary Study of Engineering Designers", *Proc. of the ASME 1999 Design Engineering Technical Conferences*, Los Vegas, Nevada, 1999.

Mark W. Lange
Nokia Svenska AB
Nokia Networks, R&D
Box 1113
SE – 164 22 Kista
Sweden
Tel: +46 (0)709 - 93 8669
FAX: +46 (0)8 - 632 05 91
E-mail: mark.lange@nokia.com
URL: www.psm.kth.se/~mla

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DESIGN SYNTACTICS – A CONTRIBUTION TOWARDS A THEORETICAL FRAMEWORK FOR FORM DESIGN

A V Warell

Keywords: Aesthetics, industrial design, domain theory, design syntactics, form functionality.

1 Introduction

Products can be analyzed and evaluated from a variety of viewpoints and disciplines. From the viewpoint of industrial design, literature abounds with studies from the perspectives of ergonomics and human factors, economics, design management and marketing, cognitive psychology, ecology, and society and culture. The lack of "its own science" seems to be evident in industrial design. Compared to other disciplines in product development, such as engineering, the lack of a theoretical basis for form design has often led styling activities to become a discussion based on opinion and subjectivity. In product development, engineering design proposals are more easily evaluated and justified due to its causal nature and the ability to "validate" a solution with scientifically based argumentation. The increasing importance of styling as a competitive edge for market success has lead to a need to be able to handle design development of the product's aesthetic form in a manner more communicable and stringent than is possible today. The situation is improving, however. Through the advances in, e.g., product semantics, we are achieving an understanding of the product with regards to how we as human beings interpret its appearance, its use and its context. Our ability to interpret the meaning of form elements – the "representative viewpoint" – is thus provided by product semantics. Yet, however, we have no tool or language for "spelling" visual composition and ingredients of form– the "formal viewpoint" – a theory modeling and explaining the content and structure of the visual form design.

In this paper, a first step is taken towards developing a theoretical framework for visual form design, compatible with models and theories found in industrial design and in engineering design science. The overall objective is to merge aesthetic and technical issues in form development, thereby bridging the gap between disciplines. The uniting key word is functionality; not only technical engineering functionality, but aiming at a wider functional definition incorporating "subjectively" determined functional aspects such as semantics, syntactics, and ergonomics. The framework is denoted "design syntactics", referring to a modeling approach aiming at capturing the contents – form entities and form elements – and structure – the compositional principles – of visual product form.

2 Related work

Efforts at understanding styling from different outsets have been carried out by many researchers. In the industrial design field, product semantics has emerged and grown into one of the most promising approaches for describing form design from the communicative perspective. Vihma [1, 2] applies semiotic and aesthetic theory in developing a model for evaluating the representative qualities of modern design products and presents a semantic analysis of product form relating to type, expression, use and identity. Product semantics as a growing discipline has been treated by a large number of authors, including Klöcker [3], Butter and Krippendorff [4], and Gotzsch [5]. Monö [6] examines product understanding from the perspectives of aesthetics, semiotics, and perception, and proposes four semantic functions as a way to introduce product semantics into active design work and for use in analysis of products. Wikström [7] builds on Monö's work and proposes a method for evaluating the four semantic functions. Akner-Koler [8] studies the structure of three-dimensional form and proposes an approach for formal analysis of compositional principles and specific form elements, "a descriptive anatomy of products". In engineering design, product styling has yet had minor influence. An influential exception is the work of Tjalve [9] who developed a theory and methodology for form development, based on research in engineering design science and aesthetics.

3 Introducing design syntactics

While product semantics tries to explain what a product represents or expresses as a conveyor of a subjectively interpretable message from the designer to the user, design syntactics aims at describing and explaining the ingredients of the visual form composition, i.e. the shapes and their arrangement, in an objective manner, see Figure 1. Sonesson [10] divides semiotic analyses into a plastic and an iconic level, resembling the division into syntactic and semantic dimensions. Vihma [2] defines the syntactic dimension as including the analysis of the product's technical construction as well as the analysis of visual details (e.g., joints, openings, holes, form crossings, texture, graphics, etc.) of the design. Since the technical and behavioral structure of products is extensively treated in engineering design science, the definition of design syntactics applied here is narrower, encompassing visual form aspects only.

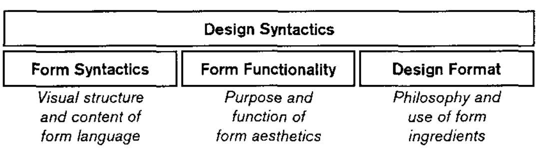


Figure 1. The framework of design syntactics.

Product semantics and design syntactics represent the two main ingredients of a "form language". In a simplified manner, a language requires four grammatical elements: alphabet, syntax, semantics, and phonology, according to Klaus [11]. Related to form design, the alphabet is represented by form elements, the building blocks of the physical shape, used for certain purposes. Form elements build up the visual brand identity, which is different from one brand to the other, and in some cases from one product (or product family) to the other. Syntax is the rules governing the composition of the form elements into "meanings" or gestalts, i.e. how the visual ingredients are arranged in the design. As proposed by Warell and Nåbo [12], syntax can be described as part of a 'design format'; a manual containing guidelines for form ingredients and rules for their use in form design work. Semantics is concerned with the meaning of the visual ingredients, i.e. what they signify as signs. The meaning is only apparent when a human perceiver is present, interpreting the signs into a message. Phonology, the accentuation, is not relevant in this case, since we are dealing with a visual, and not a spoken, language.

Both the semantic and the syntactic aspects are necessary components for describing qualitative and quantitative communicative aspects, respectively, of form design. Product semantics is finding its way to industrial use, mostly thanks to the introduction of theories and methods in design programs during recent years. Large product developing companies are beginning to adopt the thinking of the product design in semantic terms for "charging" the product with desired expression and communicative values, an important ability in the battle for attracting target consumers. However, the knowledge and application of product semantics is of no value, if the awareness of a well-formulated form language, and its relation to company image and identity, is lacking. This regards the message conveyed by the entire product system, including the product, packaging, advertising, sales campaigns, etc. The visual structure and contents (e.g., identity) of the design, its syntactics, must work in synergy with the semantics of the design, in order to "speak the same language" of the visual product brand. If not successfully correlated, the semantics may send one message of the product's properties and qualities to the market, while the syntactics of the product conveys another. This situation is illustrated in a study by Opperud [13], which indicated that the Ericsson A2618s mobile phone is not perceived by consumers to convey the signals suggested by marketing efforts and advertising, while the Nokia 8210 had been more successful in that respect.

4 The definition of form

A materialized product consists of parts. All parts have external surfaces, arranged according to a 'structural skeleton', Mortensen [14]. The skeleton spatially defines the surfaces in relation to each other. Furthermore, material fields constitute the "body" of the parts and connect the surfaces to each other. All surfaces are built up by shapes, the outer skin, defined by geometrical characteristics. The term "form" thus describes the characteristics of the external surfaces of a design, whether it is a whole product, a part, or a part of a part. Form is defined as consisting of shape (i.e. geometry and size) and configuration (i.e. spatial arrangement of shapes). Compared to the definition of form provided by Jensen [16], the term as used here does not include material attributes, but is concerned with external characteristics only.

From the standpoint of industrial design, the purpose of form is to enhance understanding of the product and to create appeal on part of the observer. As such, form is part of the quality Q of a product, i.e. what a customer or user experiences of a product's properties, Mørup [15]. Form is thus a subjective qualitative experience, which can be appreciated through various senses, e.g. sight (i.e. the properties of the form can be categorized as belonging to, e.g., aesthetics or product semantics), and the haptile and tactile senses (if the shape is perceived by touch).

5 The nature of form elements

A product's form is built up of form elements, defined as the constituent parts of a form. Form element is a recursive term, applicable on all levels of form, whether on a whole product, a part, or a part of a part, Warell [19]. Form elements define the appearance of all visible surfaces of a product. For example, the 'catwalk' running along the side of contemporary Volvo cars is a form element, shared by (distributed across) several parts of the car body, Warell and Nåbo [12]. Likewise, the grooves on the cap of a Magic Marker constitute a form element. Moreover, each and every groove is a form element in itself.

Thus the term form element, like form, is related to the characteristics of the external surfaces of a design and not explicitly with the internal material. The use of the term form element here thus differs from that provided by Jensen [16], stating that a part is decomposed into form elements, being structural elements having one or more wirk elements from a behavioral point of view. Here, a form is decomposed into form elements, which are not necessarily constrained to a single part but can be allocated across several parts as constituents of the outer visual form of a product.

6 Form Functionality

Form functionality in design syntactics relates to interactive functionality of a product, see Figure 2. Here, semantic as well as syntactic functionality is included as constituents of the communicative functions, functions that are related to form language. While the semantic functions, Monö [6], deal with the representational qualities of the product form, syntactic functions are related to the constituent form elements and their compositional structure. Syntactic functions may be forms that refer to each other by shape, or are related in terms of compositional principles, e.g., visually connecting or discerning, Warell and Nåbo [12]. The syntactic properties of a product form are largely determined by visual gestalt principles, as discussed in section 8.

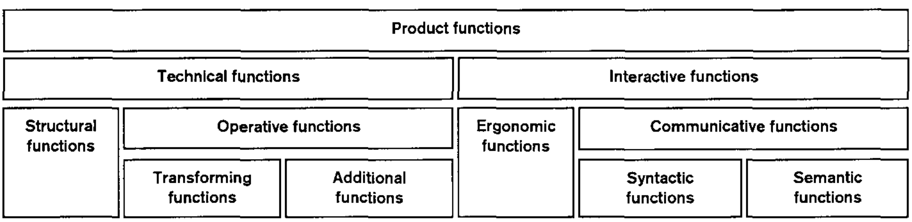


Figure 2. Function classes of a product.

Interactive functions are compatible with the function concept in design science, stated according to the following: "a function is what an element of a product or human actively or passively *does*, in order to contribute to a certain purpose", Warell [17]. The nature of the communicative functions, i.e. their mode of action, is determined by information theory, and their functionality is carried by signals, a type of data transmission from object to human. Shannon and Weaver [18] define a signal as "an action (gesture) or processed object (artifact) that provides a directive in a particular direction". For semantic functions, the sign intended by the designer is carried by the form elements of the product having properties that represent a sign. The sign is transmitted by a signal, which is interpreted by the observer. Given that the perceiver, in a suitable context, is able to receive and translate the signal into the meaning intended by the source (the designers or the company), the desired message can be communicated. The syntactic functions, i.e. the form elements used and the way they are arranged, are likewise communicated by signals which are carried by form elements, experienced by the observer in the composition and contents of the form language.

The encounter with a product with an appealing form may give a direct aesthetic, emotional experience, appreciated as a pure aesthetic experience, defined by syntactic functionality. In addition to the direct excitement of the form, the product gestalt (the totality of form, color, surface structure and so on) may also be interpreted as representing something else, that is, the function (in the sense of purpose), as noted by Monö [6]. Monö defines a sign as "any phenomenon which has significance that is independent in relation to its material form". Thus, a sign and its semantic function refer to something else, e.g., a practical product function. The product's appearance is thus a sign for its function. However, this does not mean that everything in the product gestalt can be (or is intended to be) interpreted as a sign. For example, the forward-pointing arrowhead shape of a modern high speed train may express high speed, the aluminum body state-of-the-art technology, and the clean side surfaces express weight and massiveness, a confidence creating factor for the potential passenger (semantic functions). On the other hand, the sweeping, clean lines, uncluttered shapes and horizontal graphic surface treatment may also simply appear to the onlooker as a very exciting and appealing form (syntactic functions). The latter may in many cases be a stronger incentive for finally buying or using a product than the representative aspect of the product form may be.

The domain theory, Andreasen [20], is used as a basic structural design model to capture and describe part-to-human functionality, denoted 'interactive functionality'. According to the domain theory, organs in the organ domain deliver the desired functions of the product through its parts, belonging to the part domain. Previous theory has not been capable of capturing the effect of product form, apart from the form of surfaces that have a direct contribution to technical functionality (such as structural or operative functions, see Figure 2). Tjalve [9] introduced 'functional surfaces', designating the surfaces of parts having an active function during use. Mortensen [14] denoted such surfaces, which contribute to the realization of an organ and thereby its function, 'wirk surfaces'. The remaining, "in-active", surfaces of the parts, the 'cover surfaces', are according to Mortensen "free" in the sense that they do not directly have functional contribution.

For design syntactics theory, however, it is essential to state that *all* visible external surfaces have functions, and thus contribute to organ functionality, as proposed by Warell [19]. For example, the purpose of a car body exterior is, except from having, e.g., technically determined structural and aerodynamic functionality, to create a desire in consumers to want to own that car, to inform us about the make of the car, its performance and qualities, and to relate the car to previous designs and brand values. The visual form of the car body thus certainly has the ability to deliver a purposeful effect to the observer, i.e. the form has a function. The meaning of effect, Andreasen [20], is thus extended from a pure technical transforming, part-to-part sense, to a concept capable of explaining purpose of part-to-human relations as well, Warell [17].

7 The nature of form entities

Jensen [16] presents a revised organ based structural model and recognizes wirk elements as the structural element of an organ from a behavioral point of view. That is; when subjected to a stimulus, i.e. an effect causing a behavior (a behavior being a transition of state due to stimulus), a wirk element is active regarding behavior, i.e. it has the ability to realize a function. Consequently, when not subjected to stimulus, the wirk element is passive. With the model of Jensen, functionality of a part is dictated by the existence of form elements that become wirk elements due to the structure's transition of state.

However, when considering communicative functionality, no transition of state occurs. The structure remains unaffected; the functional effect is only subjectively perceived and interpretable by an observer (by means of signaling). To be able to handle such functionality, another type of organ element is called for. This element is denoted form entity, Warell and Nåbo [12]. Hence, organs can be decomposed into wirk elements as well as form entities. The functionality of form entities is dictated by the presence of an observer. When form entities are perceived, they are functionally "active", serving either syntactic or semantic functions. Form entities are inherent to all designed objects. All shapes are perceived and reacted to, consciously or not, by vision or touch. Likewise, other signs appreciated by our senses, e.g., smell and hearing, are also important for our impression and understanding of products. Thus, awareness of syntactic and semantic functionality, allocated on the product form through the use of form entities, is beneficial for the aesthetic appreciation of the product. Thus, organs decomposed into form entities, or fulfilling communicative functionality, may be denoted aesthetic organs.

Like a wirk element, a single form entity contributes to the function of an aesthetic organ, but may not be sufficient for realizing the whole function in itself. For realizing an interactive function, an aesthetic organ may be composed of several form entities. In the same way as it is possible to describe machine organs as any type of technology that delivers the desired technical functions, an aesthetic organ can be described as any form solution that creates the desired communicative functions. Aesthetic organs are thus a special class of organs fulfilling syntactic and semantic functionality, existing in a superimposed manner along with technically determined organs. Thus, the functionality of a certain part may be described by several organ structures, realizing different types of functionality, and having wirk elements and form entities as the 'active units' (functional regions, Warell [19]). While a wirk element is a point, line, surface, or space of continuous geometry and uniform material, Jensen [16], a form entity consists of a one-, two- or three-dimensional shape (i.e. a point, line, surface or body), a spatial configuration of such shapes, or a relation between such shapes. Thus, form entities lack any internal material attributes. Furthermore, form entities are not constrained to belonging to a single part of continuous material or geometry, but may be distributed across a form.

For example, the door handle of a car door has a multitude of functions and organs. Just mentioning a few, the door handle should make it possible to open the door (transforming function, delivered by wirk elements). The handle should withstand the force applied to it during opening (structural function, wirk element). The user must understand that the handle is in fact a device for opening the door, and he must furthermore understand how to use it, and he must want to use it (semantic functions, form entities). The handle must give the impression of belonging to the car; i.e. its form must be in accordance with form elements of the handle and of other form elements of the car, all forms and the whole gestalt must create a harmony and a balance in order not to stick out as unmotivated (syntactic functions, form entities).

8 Form entities and visual design aesthetics

In a finished product design, form elements interact to create a system of visual relations, or gestalts, in the form. A gestalt is a typical realization of a form entity; a number of form elements interact, creating a visual entity of "higher order". It is the creation of gestalt configurations that enables us as human perceivers to "read" a product's design, too see its form. Monö [6] defines a gestalt as "a discernible whole; an arrangement of parts so that they appear and function as a whole which is more than the sum of the parts". The form, color and material structure are not merely isolated factors in the wholeness of the design, but they influence each other, and in high quality designs, create synergetic effects.

Form relations, i.e. couplings between form elements, can also constitute form entities. Examples of this is the creation of proximity, similarity, harmony, contrast, dynamism, symmetry, balance, rhythm, orientation, proportion, etc., by conscious arrangement of form elements. Such relations are part of the 'gestalt factors', Monö [6], certain factors that create and help us discern gestalts during visual perception. Important research into this field has been done by Akner-Koler [8], who provided a systematic categorization and classification of such factors, and Klöcker [3], who called such interacting features of visual composition the 'mathematical qualities of form', and further distinguished between arithmetic, geometric, and topological qualities of form.

9 Conclusions

The main contribution of design syntactic theory is the integration of models and concepts of the industrial design and engineering design fields. From industrial design, concepts like product semantics and gestalt theory are adopted. Engineering design contributes with design science, including functional theory, domain theory and organ modeling. The sharing of theoretical concepts with those found in engineering design science makes it possible to ascribe functional properties to aesthetic aspects of form design. Thereby, the basis for creating a common theoretical model, capable of explaining and relating all aspects of the product's design (technical as well as aesthetic), has been established. The emerging, composite theory is denoted "design syntactics".

10 Acknowledgement

This work was financially supported by the Swedish Foundation for Strategic Research through the ENDREA research program. This support is gratefully acknowledged.

References

[1] Vihma, S., "Defining Form in Industrial Design", Proceedings of UIAH'87 Conference, University of Industrial Arts, Helsinki, 1987

[2] Vihma, S., "Products As Representations", University of Art and Design, Helsinki, 1995

[3] Klöcker, I., "Die Produkt-Form und menschliches Verhalten", (Diss.), Technische Universität Hannover, 1980

[4] Butter, R. and Krippendorff, K., "Product Semantics – Exploring the Symbolic Qualities of Form", The Journal of the Industrial Designers Society of America, Spring, 1984, pp. 4-9

[5] Gotzsch, J., "Beautiful and Meaningful Products", Proceedings of Design Plus Research, Politecnico di Milano, Milan, 2000, pp. 146-154

[6] Monö, R., "Design for Product Understanding", Liber AB, Stockholm, 1997

[7] Wikström, L., "Produktens budskap", Licentiate Thesis, Department of Consumer Technology, Chalmers University of Technology, Göteborg, 1996

[8] Akner-Koler, C., "Three-Dimensional Visual Analysis", Department of Industrial Design, University College of Art Crafts and Design, Stockholm, 1994

[9] Tjalve, E., "A Short course in Industrial Design", Newnes-Butterworth, London, 1979

[10] Sonesson, G., "Pictorial Concepts", Lund University Press, Lund, 1989

[11] Klaus, G., "Wörterbuch der Kybernetik", Dietz Verlag, Berlin, 1968

[12] Warell, A. and Nåbo, M., "A Model for Visual Design Aesthetics Based on Form Entities", Proceedings of NordDesign 2000, Technical University of Denmark, Lyngby, 2000, pp. 159-168

[13] Opperud, A., "Is the Beauty in the Eye of the Beholder - en semiotisk studie av mobiltelefondesign", M.Sc. Thesis in Psychology, Göteborgs universitet, Institution of Psychology, 2001

[14] Mortensen, N. H., "Design Characteristics as Basis for Design languages", Proceedings of ICED 97, Tampere, 1997, pp. 26-30

[15] Mørup, M., "A New Design for Quality Paradigm", Journal of Engineering Design, Vol. 3, No. 1, 1992, pp. 63-80

[16] Jensen, T., "Functional Modelling in a Design Support System", Diss., Department of Control and Engineering Design, Technical University of Denmark, Lyngby, 1999

[17] Warell, A., "Extending the Concept of Functionality – A Proposal for a Use-Oriented Modelling Approach for Product Design", submitted to the Journal of Mechanical Design, 2001

[18] Shannon and Weaver, 1949:95

[19] Warell, A., "Industrial Design Elements – A Theoretical Foundation for Industrial Design Based on a Design Science Perspective", Licentiate Thesis, Division of Machine Design, Linköping University, Linköping, 1999

[20] Andreasen, M. M., "Machine Design Methods Based on a Systematic Approach – A Contribution to Design Theory", Diss. (in Danish), Department of Machine Design, Lund Institute of Technology, Lund, 1980

Anders Warell
Chalmers University of Technology, Industrial Design Engineering, SE-412 96 Göteborg, Sweden, Tel: +46 31 772 8289, Fax: +46 31 772 5858, E-mail: warell@mvd.chalmers.se, URL: http://www.mvd.chalmers.se/~warell

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

ELEMENTARY LEVEL IN THE FUNCTIONAL-TECHNOLOGICAL DESIGN

C Opran and A Armeanu

Keywords: functional design, technological design, distinctive features

1 Introduction

The design process has already established successive stages with included steps. Functional aspects are recognised as determinative for the whole design process evolution but their presence is usually localised at the beginning of it. On the other side, technological aspects are recognised as determinative too, but are generally considered at the end of the design process. This is true for the traditional approach, which has proved his limits in time. New methodological approaches were developed in order to solve the inconvenience of the traditional one.

There are very few approaches that reveal the necessary connection between the functional and the technological aspects. Each of them has tried to convince on there utility and was rallied to a new concept of the engineering approach named concurrent or simultaneous engineering.

The design process is often separated into product design and technology design because of great differences of the life-cycle stages they are referring on. In fact, the two processes cannot be separated and is the merit of concurrent engineering to reconsider and propose solutions for solving the conventional approach's contradictions.

The paper introduces some results of the authors in this field, having as goal to demonstrate, first of all, that functional and technological aspects contains the essence of the engineering approach, after, the necessary connection between them and finally, their relevance until ultimate level called elementary level.

2 Related work

Many researchers have discussed the functional issues. Some of them, like Hubka or Pahl and Beitz, are already classics of the field but subsequent approaches have demonstrated that implications of these problems are far from solution. The most difficult problem has proved to be the formalisation of function concept and related data in a computer integrated solving environment. Many papers are referring on function classification [3] but this level of approach is insufficient for the following design concreteness. Function taxonomy is very important but could be applied only to a quite general level were relation with technological features cannot be revealed.

Other approaches have introduces theoretical formalisation [1] which seems to offer a solution but deeper approach proved to be necessary. Essentially remark is that on last level should be considered elementary ports or elementary distinctive features, simple or complex.

3 Theoretical basis and structure of Functional-Technological Engineering (FTE) method

A new proposed methodology, Functional-Technological Engineering (FTE), and a Knowledge and Operation Computer Based Environment (KOCBE) have been developed having as main goal to rally on actual concurrent engineering methodologies and tools [2], [4], [5]. The terms used in the definition of the method deserve explanations:

• ***Functional*** means all aspects related to the usage value, including functional requirements, functional implications of the requirements (functional performance, utility), list of functions for certain product, functional modelling, functional decomposition

• ***Technological*** means all aspects related to the transfer of conception into concretness for the entire life of a product, including diverse ways for the conversion of materials, energy or information imputes into physical form.

Within functional approach conceptual solutions for obtaining desired performances of the future product result but within technological approach there are prescriptive characteristics obtainable under available circumstances. A relation between two approaches should be established and is possible only simultaneously considering both aspects from the general to the particular level (see Fig. 1). The last level is the elementary one as result of a gradual functional decomposition from general to particular and is referring to features belonging to the component, part or item of a product.

Functional decomposition means that starting from general function, continuing with principal functions and then subfunction, with technical functions, finally, necessary elementary functions have to be found for each feature of a part. The functional approach presumes an gradual succession with the analyse of each part considering integral functional, partly functional or specific functional categories (see Table 1) Each category has to be explained and approach differences have to be revealed.

Functional evolutive approach deciphers implications of the functional requirements settling down the Functions List. This could be converted into functional models. Based on geometrical models, functional ports associated to correspondent features are considered and correlation between functional and technological features at elementary level could be revealed.

The function concept in FTE is both self distinguished and related to the technology concept for analysing or defining functional-technological features at the most elementary level: items surfaces as functional technological ports.

Already accepted concepts of ports or distinctive features have been used in FTE but also some new theoretical considerations and practical procedures have been included. Going further for the categorised part, three types of ports or distinctive features should be considered: functional, technological or for physical continuity (see Table 1). Each could be identified as conventional geometrical or technical denomination, position and even sampled.

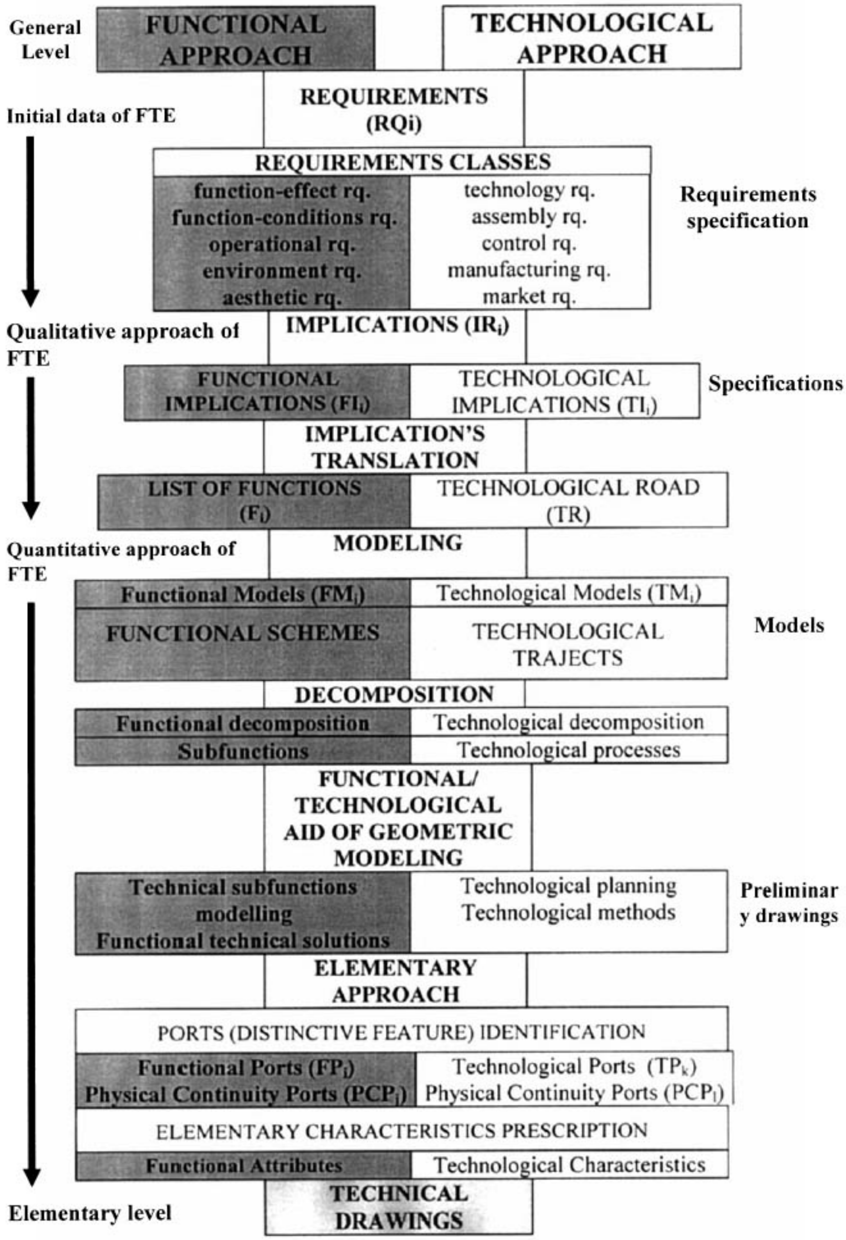


Figure 1 Gradual approach within FTE methodology

For the quantitative conceptual level proposed functional approach is deepened rely on geometrical models represented by the technical drawings. This becomes possible considering the Functional Ports (FP_i) associated to correspondent geometrical elements (surfaces or groups of surfaces) belonging to each piece of a product. Assuming the functional ports, correlation between functional and technological features at this particular level could be revealed. At this ultimate level, associated characteristics of distinctive features either functional, technological or for physical continuity have to be recovered into technical drawings. The main destination of technical drawings is to communicate with manufacturing stages. One of the remaining difficult problem of the design process is the evaluation of each of these characteristics to decide whether or not notified values are corresponding to desired functional performance of the next level, the subassembly and so on. First step is the appointment of typical functional association for each item (see Table 1).

Table 1 Functional definition

FUNCTIONAL AID OF GEMETRIC MODELING		
Functional definition of technical parts (items, elements)		
Integral functional part (having all integrated function)	Partly functional part (requiring at least one new joining function)	Specific functional part (requiring all new functions)
Functional unbreakable parts	Partly functional decomposable parts	Complete functional decomposable parts
Standardised or typified items to be integrated in new assemble	Subassemblies or attached items to new assemble	Special designed items for an assemble
Coupling ports	Coupling ports	Functional ports Technological ports Physical (space) continuity ports

The design engineer identifies and marks functional ports for each part and chooses the most appropriate elementary functions from the general list (see Table 2 and Figure 3). Definitely functional attributes fitted to each particular case can be selected. from the functional data base module (see Table 2 and Figure 2).

Table2 Typical elementary function and functional conditions

Typical elementary function	**Functional conditions**
Rotational movement	Low friction, fitting
Translation movement	Low friction
Contact	Mobility limitation
Fix	Ensured immobility
Pass	Easy transfer
Seal	No leaking
Position	Distance (angle) respected
Space continuity	Integrity body definition
Space definition	Volume of a cavity, section of a channel
Aspect	Ergonomic impact
Inform	Marked suggestions, symbols, words

Table 3 Functional attributes and related categories

Functional attributes	**Related categories (Examples)**
General port type	Functional simple /complex/combined Technological (technological reasons only) simple /complex Physical (space) continuity
Simple geometrical port name	Plane, Cylinder, Cone, Sphere
Complex geometrical port name	Specific complex surfaces (helical, spline etc)
Combined geometrical port name	Typical combinations
Port position	Internal/External/Transitory
Functional geometric dimension	Length/Width/Depth Diameter/Radius
Functional performance measures	Movement, position Speed, acceleration Force (applied, transmitted, reaction, weight, friction) Pressure Temperature
Functional conditions geometry microgeometry surface / structure	 precise, normal, not imposed smooth, rough, adherence, low friction hard, soft, protected / homogeneous, stress resistant

In a similar manner a gradual decomposition of the technological aspects from general to particular could be considered. Technological evolutive approach deciphers technological implications within requirements. The Technological Road is a general resultant technological model. Integral technological, partly technological and specific technological category of each parts are defined and explained (see Table 4).

Table 4 Technological definition

TECHNOLOGICAL AID OF GEMETRIC MODELING		
Technological definition of technical parts (items, elements)		
Integral technological part (without technological intervention except montage)	Partly technological part (requiring at least one new technological intervention)	Specific technological part (requiring all technological interventions)
Technological unbreakable	Partly technological decomposable parts	Complete technological decomposable part
Ready to use standardised/typified parts or items	Decomposable parts like subassemblies or attached items	Special design and manufactured part
Coupling ports	Coupling ports	Functional ports Technological ports Physical (space) continuity ports

The appropriate technological characteristics must be established in very close relation with the functional elementary level approach.

Each port is not only a technical functional solution but also requires most complete technological information for the manufacturing stage, information that completes the orthographic projections of the technical drawing. In such a way, design engineer could be able to establish and to propose more complete and correct technical specifications for post conceptual stages within technical drawings (see Table 5).

Table 5 Technological attributes and related characteristics

Technological attributes	**Related characteristics (Examples)**
Geometric form concretness	Lineal or angular precision (tolerance range) Fit precision (fit classes) Relative position precision Form precision (flatness, roundness)
Microgeometry	Surface roughness Texture
Surface and structure state	Hardness Hardened thickness Layer (painted, anticorrosion layer)

The sketch in the figure 3 is an example of specific technological part *Upper body* from assembles *Valve* with marks for all representative ports. Some of these were selected as examples for introducing the operational elements of FTE (see Table 6).

Table 6 Examples of functional attributes and related functional categories

Port symbol	Q	B	C
Functional attributes	**Related functional categories**		
General port type	Functional	Technological	Functional
Elementary function	Pass	-	Fix
Type category	Simple	Simple	Complex
Geometrical name	Cylinder,	Cone (chamfer)	Thread (type)
Position	Internal	External	External
Dimensions	Diameter	Length, Angle	Diameter
Functional performance measures	No special rq.	-	Force transmitted
Functional conditions geometry microgeometry surface / structure	 passing hole normal no special condition	-	 assembly thread normal no special condition
Technological attributes	**Related technological characteristics**		
Geometric form concretness	Dimensional tolerance ±0.2 Roundness normal	Dimensional tolerance ±0.1 -	Dimensional tolerance g8 -
Microgeometry	Roughness height 63	Roughness height 63	Roughness height 63
Surface and structure state	General prescribed	General prescribed	General prescribed

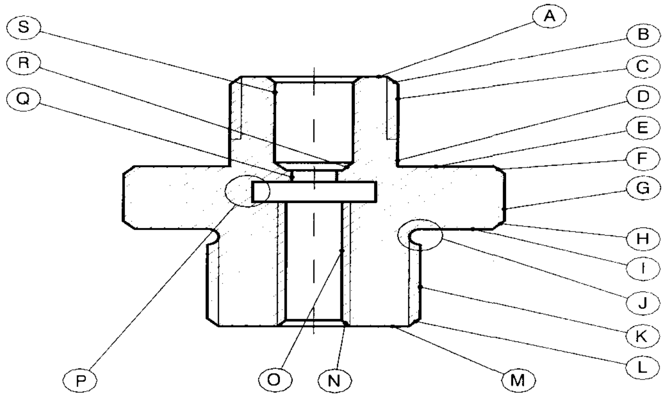


Figure 2 Part *Upper body* from assembles *Valve*

4 Computer Aided Functional Technological Engineering (CAFTE) introduction

As normal for the actual development of engineering tools, the entire methodology of FTE was implemented in a software medium able to aid the whole application subject. FTE is able to offer the possibility not only of an integral but also of a partial (sequential) approach of an engineering design problem through the aiding software support. CAFTE software package includes:

1. general software program for assisting and guiding the user in order to complete his approach and carry out his task basing on the proposed procedure,
2. FTE Data Base System for accessing, administrating and processing all needed information, (see Fig.3).

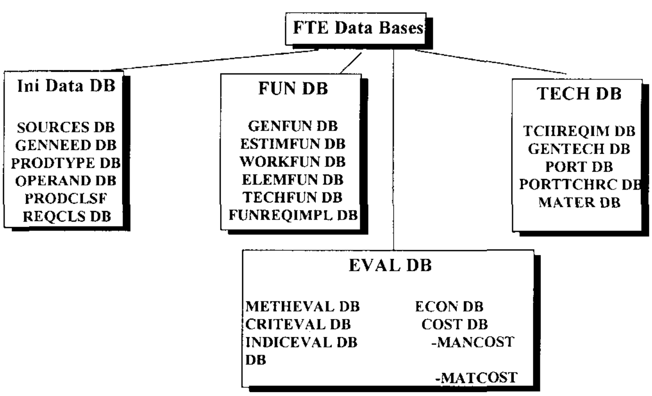


Figure 3 CAFTE Data Base System

3. Connected programs designated to facilitate the settlement through specific techniques (functional modelling, geometric modelling, functional-technological evaluation).

4. Facilitation for connection of other programs in order to join other compatible techniques, which operates with similar features, for instance VE, QFD or FMEA

First results obtained testing CAFTE have required formal supports for the transfer of information, like report sheets. Next step forward is intended to be the dynamic transfer of information in a Knowledge and Operation Computer Based Environment (KOCBE).

Conclusions

The goal of the paper is to convince on the importance of decomposition until the elementary level in the engineering design approach. Only an integral computer aided design process methodology can succeed, final result being more efficient and rational prescriptions of technological characteristics on technical drawings.

Method has already proved his capabilities in educational field and work is going on in order to complete methodology and associated databases and to extend operational capability for the industrial field use.

References

1. Andersson S., Hugnell A.B., Functional Modelling in Mechanical Design, Proceedings of ICED 91, Zurich, 1991
2. Armeanu A., Contribution Regarding Computer Aided Functional-Technological Engineering of Products, Doctoral thesis, University POLITEHNICA, Bucharest, 1998 (in Romanian)
3. Kirschman, C.F., Fadel, G.M., Classifying Functions for Mechanical Design, 1998, Journal of Machanical Design, Vol. 120, pp. 475-482
4. Opran C., Armeanu A., From General to Particular in Functional Technological Engineering of Advanced Materials Products, ICE 2000 Proceedings of the 6^{th} International Conference on Concurrent Enterprising, Toulouse, France, 28-30 June 2000, pp. 205-212
5. Opran C. et al, Development of Functional-Technological Engineering of Products within the Management of Product Life-Cycle, ICME 2000 Proceedings of the 2^{nd} CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, 21-23 June 2000, Capri, Italy

Prof. PhD M.Sc. Eng. Constantin Opran, Lecturer PhD Eng. Alexandru Armeanu
POLITEHNICA University of Bucharest, Department of Manufacturing Technology, Splaiul Independentei St, 313, Bucharest, ROMANIA, Phone (004) 01-4112604, Fax (004) 01-4112687, E-mail:opran@labsmn.pub.ro

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

VERIFICATION OF A NEW MODEL OF DECISION-MAKING IN DESIGN

C T Hansen

Keywords: Engineering design, decision-making, protocol analysis.

1 Introduction

In a recent paper Hansen & Andreasen [1] have proposed a new framework of design decision-making. The framework consists of two models: the decision node and the decision map. The *decision node* is a model of the interrelated decision-making activities when making a decision during design. The node is meant to be generic in the sense that it contains all sub-activities in a decision-making activity. The *decision map* is a model showing what is synthesised during the design process and therefore the object of decision-making.

The aim of the research documented in this paper is to make a first attempt towards verifying the decision node. We analyse two protocols of design work carried out by individual designers in experimental settings. In the protocols we expect to identify several decision-making episodes, where each episode is characterised by consisting of one or more sub-activities as proposed by the decision node.

In this paper we describe the decision-making model and the research method to verify the model. We present the result of the protocol analysis, and finally we conclude.

2 Decision-making in design

In the situation where an industrial company launches a new product to the consumer market, based upon more or less new invented solutions, the evaluation and decision-making activities during design become rather complex. The engineering designer or design team has:

- to target the customer's need and values, and ensure timely market introduction;
- to target the business to create proper profit for the company;
- to find the best design solution and ensure that it fits to all of the product's life phases; and
- to co-ordinate all design and activity elements into a proper totality and timeliness.

The goals of many stakeholders have to be reflected in the product design specifications, and decision-making during the design process becomes both a search for the best design solution and a navigation towards a feasible and efficient process. Thus, the decisions made during the design process have a critical impact both on the design solution obtained, but also on the design process in itself.

Two key questions arise for the engineering designer. Firstly, which activities contribute to establishing a sound basis for making a design decision? And secondly, what is the object of a design decision?

2.1 Related work

A survey carried out in British industry by Wright et al. [2] indicates that design methods are sparsely adopted and used in industrial practice. With respect to decision-making during design this is an unfortunate situation for two reasons. Firstly, because of the importance of making the right decisions in order to ensure high quality results, and secondly because many methods to support decision-making do exist, see e.g. [3] and [4].

A widely referenced book on engineering design is Pahl & Beitz [3]. The authors treat the evaluation of concept variants against technical and economic criteria in detail, and they outline a basic evaluation procedure. The purpose of evaluating concept variants is to provide an objective basis for identifying the concept with which to proceed to embodiment design.

Roozenburg & Eekels [4] set up the Basic Design Cycle consisting of the activities: analysis, synthesis, simulation, evaluation, and decision. Roozenburg & Eekels treat evaluation and decision-making by describing a number of decision methods and their theoretical fundament. Several decision rules are discussed and as a pedagogical means to explain the reliability of these rules the authors have constructed an example, where solution alternatives have properties so that the best alternative varies, depending on the decision rule applied.

According to Ward et al. [5] development of medical devices is a highly regulated area. Any new medical device has to be demonstrated as being "fit for purpose" before it can be released to market. Alexander & Clarkson [6] present a normative design process model focusing on such a validation of medical devices. The aim of the model is to provide the engineering designer with a proactive role towards validation during the design process. The design validation model is a framework consisting of a process model and a number of design tactics.

An element in design decision-making, which to the author's knowledge is not treated in detail in the literature, is the impact of the decisions made on the design process. Asimow [7] states: "The skilled designer is not only goal-oriented; *he also understands the process for reaching the goal.* Although he makes trials, he avoid repeating errors; and further, he seeks positive indications of the realizability of his solutions before he goes too far in a given direction." Thus, the engineering designer or design team has to decide not only upon the solution alternatives, but also on the progression made in the design project.

The goal of Dwarakanath's work [8] is to establish a framework for a computer-based system to support design decision-making. Dwarakanath carries out an empirical study of design work in an experimental setting, and he observes among other things that individual designers tend to apply a single-string solution-oriented approach, where alternatives are not considered unless the pursued direction in the solution space is recognised to be infeasible. From his observations Dwarakanath recognises a need for a structured and explicit basis for supporting design decision-making based on identified types of the decision-making process, the use of criteria, and the types of information used.

From the literature we observe that there exist many methods and guidelines to support decision-making in design, and interesting elements of insight from empirical studies are reported. However, we note that the current situation in industrial practice is characterised by sparse adoption and use of existing methods. We believe the cause of this situation is a lack of a comprehensive understanding of design decision-making.

2.2 A new design decision-making model

As a means to enrich the engineering designer's understanding of design decision-making Hansen & Andreasen [1] have proposed a framework consisting of two models: the decision node and the decision map.

During the design process the engineering designer or design team have to make decisions repeatedly. The decision node models these decision episodes. The node, see Figure 1, is a generic, elementary decision-making activity consisting of six sub-activities: *to specify*, *to evaluate* solution alternatives, *to validate* a design solution, *to navigate* through the solution/activity space, *to unify* the current decision into consistent wholes, and *to decide*.

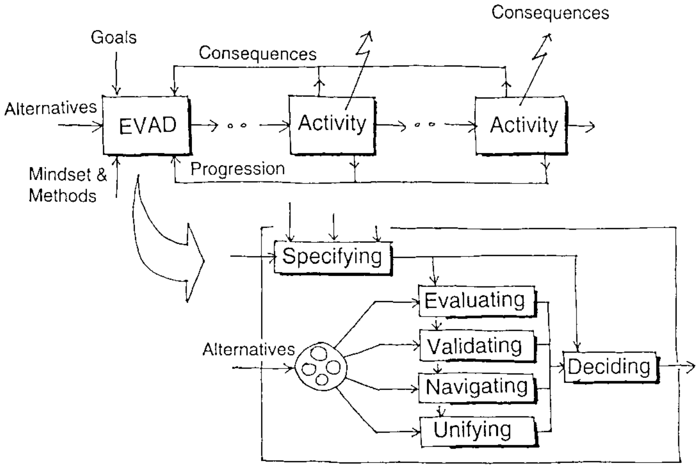


Figure 1. The decision node consists of six decision-making sub-activities.

The decision node is generic in the sense that it contains the full set of sub-activities, which are found in different design decision episodes:

- *To specify* sets the criteria for the decision. Different stakeholders have different goals. It is the engineering designer's task to compile the stakeholders' goals and translate these goals into product design specifications.
- *To evaluate* a number of design alternatives is to identify the better ones or establish a ranking of the alternatives with respect to the current criteria.
- Whereas evaluation implies a distinction between better and worse, validation is concerned with whether a design solution is acceptable. *To validate* is to check whether the current design proposal is "fit for purpose" with respect to identified product life concerns, e.g. manufacturing, distribution, or use.
- The skilled engineering designer is not only goal-oriented; he/she also understands the process for reaching the goal. *To navigate* is to consider not only the current solution alternatives, but also these alternatives' influence upon the progression in the design project with respect to allocated time and resources.
- During the design process a solution is synthesised through a sequence of complex decisions. Both the process and the solution are decomposed into elements, which have to be co-ordinated in order to obtain a satisfactory result. The engineering designer has *to unify* the current decision into the totality of process and solution.

- To evaluate, to validate, to navigate, and to unify are sub-activities, which result in a basis for making a decision. In a decision episode each of the sub-activities carried out provides a signal, and the engineering designer or design team has *to decide* based upon the signals obtained.

The decision map, see Figure 2, is a model showing what is synthesised during the design process and therefore is the object of decision-making. During design three artefact objects are designed, namely the *product*, the *life phase systems* (e.g. production system and distribution system) and the *meetings* between the product, the operator and the life phase system. However, also the *business* for the company and the *design process* itself are designed. Thus, the engineering designer has to be aware of the fact that design decisions have consequences in five dimensions.

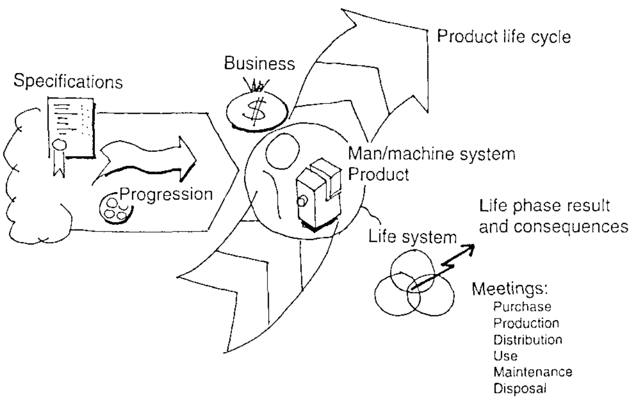


Figure 2. The decision map shows the complex object of design decisions.

Hansen & Andreasen [1] postulate that an engineering designer with a proper understanding makes better decisions, and that the decision node and decision map are fundamental elements in building an understanding of design decision-making.

3 Hypotheses and research method

In order to make a first attempt towards verifying the decision node we have chosen an empirical approach. The method chosen is protocol analysis of individual designers carrying out design work in experimental settings. We identify the decision episodes during design work, and for each episode we classify the sub-activities carried out by the designer.

In recent years several research projects with the aim of understanding the designer's cognitive behaviour have been carried out applying protocol analysis, see [8], [9] and [10]. In Cross et al. [11] the properties and limitations of protocol analysis as a research technique for analysing design work are treated in detail. A design session was videotaped where three designers were solving a given design problem. Various researchers were invited to analyse the video recording. The analyses show the nature of design from various viewpoints, e.g. design strategies, information management in conceptual design, and the use of episodic knowledge in problem solving. Based on the results of Cross et al. we conclude that protocol analysis is an adequate research method for our purpose.

For this research we have chosen to analyse two design sessions video recorded by Blessing. Blessing [9] develops a concept for a design support system. The concept is a process-based system, named PROSUS. In order to evaluate whether PROSUS improves design behaviour, i.e. supports a more effective and efficient design process, Blessing analyses a series of design sessions, where two types of designers individually solve a given problem. The *experimental designers* apply the PROSUS concept to structure their work and result, whereas the *control designers* do not use the structuring imposed by PROSUS. In each session the designer is thinking aloud while working, and the session is video recorded. After the session the videotape is transcribed. Thus, the object of analysis is the videotape, the verbal transcription, and the sketches and drawings produced by the designer.

The problem to be solved by the designers is to design a wall mounted swivel mechanism carrying an optical device, see [9] and [10]. The designer has to document his/her solution in an assembly drawing of the product and in a bill of materials. Since the purpose of our research is to verify the decision node we prefer to analyse design sessions carried out by control designers. Blessing has transcribed two sessions of this type, namely control designers cd1 and cd3, where cd1 uses four hours to finalise the design assignment and cd3 two hours. Thus, our protocol analysis will be based on the sessions of control designers cd1 and cd3.

3.1 Hypotheses

In order to verify the decision node, i.e. to check whether the contents of the node are in accordance with the decision-making phenomenon, we identify the decision episodes during the design sessions, and for each episode we identify the decision-making sub-activities.

The first hypothesis is that the decision episodes found in the sessions are characterised by consisting of one or more sub-activities as proposed by the decision node. Often design decision-making is not only based on one sub-activity, e.g. to evaluate, but on more sub-activities, e.g. to evaluate and to navigate.

The second hypothesis is that the decision episodes will not contain any type of decision-making sub-activity, which are not proposed by the decision node.

3.2 Research method

The examination of the chosen design sessions is carried out in two phases: in the *encoding phase* the decision episodes are identified and each episode is encoded into an encoding scheme. In the *analysis phase* the encoding schemes are analysed and observations regarding the verification of the decision node are formulated.

According to Valkenburg [12] encoding of design sessions recorded on videotape involves two steps. The first step is segmentation of the design session data, and the second step is the encoding of the segments. In our research segmentation is to identify the decision episodes, i.e. the design session data is interpreted into two types of segment: segments denoting decision-making activities, and segments denoting any other activity. Encoding the decision-making segments is to identify the sub-activities in each segment, i.e. to classify the designer's verbal utterances with respect to the sub-activities proposed by the decision node.

To segment the design session data involves answering the question: what does a decision episode look like? We cannot expect the designer to express his/her decision verbally. Thus, we decide to base the segmentation on three assumptions:

1. A decision episode has to be identified by studying both the body language, e.g. gestures and movements, and the verbal utterances of the designer.
2. Decision episodes are characterised by reflections upon either design proposals or design activities.
3. The designer does not make many decisions during a two to four hour design session. Less than ten decision episodes are expected.

According to Valkenburg [12] and Ericsson & Simon [13] the segments have to be encoded into an encoding scheme. We decided to encode the verbal utterances of the designer into a closed set of categories. One category for each sub-activity proposed by the decision node and a category for "other". The classification is as follows:

- *Specify* involves a statement concerning compilation of design criteria.
- *Evaluate* involves a statement concerning either the value of a design alternative, on a design alternative being better/worse.
- *Validate* involves a statement whether a design alternative is "fit for purpose", e.g. "not problematic" or "in accordance with practice", with respect to manufacturing, use, etc.
- *Navigate* involves a statement regarding the progression and feasibility of the design work, i.e. which activity to do next or in which direction to go next.
- *Unify* involves a statement concerning the current design solution or design activity in relation to the totality of product or design session.
- *Decide* involves a verbally expressed decision.
- *Other*: any statement, which does not belong to one of the six first categories.

4 Analysis

After having carried out the encoding, the encoding schemes are analysed. In cd1's design session we identified six decision episodes, and in cd3's we identified seven. Table 1 shows the sub-activities identified in the decision episodes of cd3. The result of cd1, which looks similar, is not shown due to the restriction on the length of the paper.

Table 1. The sub-activities identified in the decision episodes of cd3.

Decision episode	Time (min:sec)	Specify	Sub-activities providing signals: Evaluate	Validate	Navigate	Unify	Decide	Other
No. 1	0:20			•			•	
No. 2	1:20		•		•			•
No. 3	1:00	•			•		•	•
No. 4	1:32		•		•		•	
No. 5	3:57		•		•			•
No. 6	0:57		•		•	•		
No. 7	1:44			•			•	•

When focusing on the decision-making sub-activities, evaluate, validate, navigate, and unify, we observe from table 1 that:

- in three decision episodes one decision-making sub-activity is found;
- in three decision episodes two sub-activities are found;
- and finally in one decision episode three sub-activities are found.

This observation is a very important step towards verifying the decision node. With respect to the first hypothesis we conclude that the decision-making sub-activities proposed by the node can be identified in the decision episodes.

Regarding the second hypothesis of this research we have to determine whether the designer carries out decision-making sub-activities in a decision episode, which are not predicted by the decision node. Thus, we have to analyse the contents of the "other"-statements. As an example we focus on decision episode no. 5. After this episode cd3 begins to make the assembly drawing. During the episode cd3 is concerned about where to put the relevant views on the paper, and he comments: "I am now uncertain where to put the first line/ and that is a matter of personal uncertainty/ wherever I am laying carpets at home I hate committing myself to the first cut/ and this is a bit like laying a carpet". We observe that this comment by cd3 is a comment regarding his personal attitude or behaviour, and that the comment is not a kind of decision-making activity. This observation holds for all "other"-statements made by cd1 and cd3 during their design sessions. Thus, we conclude that we have not identified sub-activities to establish a basis for making a decision, which are not contained by the decision node.

From Table 1 it is interesting to note that cd3 in many decision episodes navigates. As stated in section 2 navigation is an element in design decision-making, which to the author's knowledge is not treated in detail in the literature. Thus, this observation raises an interesting question for future research: By which means can the engineering designer be supported in his/her navigation through the design activity/solution space?

Another interesting question is whether the pattern of decision-making changes during the design process. We could imagine that in the early phases of the design process where many solution alternatives are considered, evaluation is the most frequent decision-making sub-activity, but when a solution concept has been chosen, the decision episodes change nature and validation becomes more important.

5 Conclusions

In this paper we have presented a new framework of design decision-making, which consists of two models: the decision node and the decision map. As a first attempt to verify the decision node, i.e. to check whether the contents of the node are in accordance with the design decision-making phenomenon, we have analysed two design sessions carried out by designers working individually on a given design problem in experimental settings. The result of the analysis is a twofold verification of the decision node. Firstly, the sub-activities contained in the decision node can be identified in the decision episodes found in the sessions, and secondly, in the sessions we have not identified sub-activities to establish a basis for decision, which are not contained in the decision node.

This first attempt to verify the decision node is based on observing only two designers working in an experimental setting. However, when we in this situation find complex decision episodes characterised by more decision sub-activities, we expect that in a real life design project the decision episodes are equally complex. Thus, we have reason to believe that the decision node provides a valid description of the decision-making phenomenon.

The research documented in this paper has two perspectives: a research perspective and a practice perspective. We have outlined questions for future research: by which means can the engineering designer be supported in his/her navigation? And does the pattern of decision-making changes during the design process? From a design practice perspective we believe that an engineering designer with a proper understanding makes better decisions, and that the decision node and map are fundamental elements in building an understanding of design decision-making.

Acknowledgement

I am indebted to Professor Luciënne Blessing for allowing me access to the videotapes.

References

[1] Hansen, C.T. & Andreasen, M.M., "Basic thinking patterns of decision-making in engineering design", International Workshop on Multi-criteria Evaluation, MCE 2000 Neukirchen, September 14-15, 2000

[2] Wright, I.C., Campello, A.C. et al., "A Survey of Methods Utilisation During Product Design Process in UK Industry", Engineering Design Institute, Loughborough University of Technology, 1995

[3] Pahl, G. & Beitz, W., "Engineering Design – A Systematic Approach", Second Edition, Springer-Verlag, London, 1996

[4] Roozenburg, N.F.M. & Eekels, J., "Product Design: Fundamentals and Methods", John Wiley & Sons, Chichester, 1995.

[5] Ward, J.R., Clarkson, P.J. & Bracewell, R.H., "Supporting Medical Device Verification", Proc. ICED '99, Münich, Vol. 1, pp. 429-432, 1999.

[6] Alexander, K. & Clarkson, P.J., "A Validation Model for the Medical Devices Industry", Proc. EDC 2000, Brunel University, UK, pp. 389-396, 2000.

[7] Asimow, M., "Introduction to Design", Prentice-Hall, Inc., Englewood, N.J., 1962.

[8] Dwarakanath, S., "Understanding and Supporting Decision Making in Engineering Design", dissertation, Cambridge University, England, 1996.

[9] Blessing, L.T.M., "A process-based approach to computer supported engineering design", dissertation, University of Twente, the Netherlands, 1994.

[10] Dylla, N., "Denk- und Handlungsabläufe beim Konstruieren". München TU, Dissertation 1990 und München Hanser Verlag, Dissertationsreihe 1991.

[11] Cross, N., Christiaans, H. & Dorst, K. (eds.), "Analysing Design Activity", John Wiley and Sons, Chichester, 1995.

[12] Valkenburg, R.C., "The Reflective Practice in product design teams", dissertation, Delft University of Technology, the Netherlands, 2000.

[13] Ericsson, K.A. & Simon, H.A., "Protocol Analysis - Verbal Reports as Data", A Bradford Book, The MIT Press, 1996.

Claus Thorp Hansen
Section of Engineering Design, Department of Mechanical Engineering, Akademivej, Building 358, Technical University of Denmark, DK-2800 Lyngby, Denmark, Tel: + 45 45 25 62 73, Fax: + 45 45 88 14 51, Email: cth@mek.dtu.dk, www.iks.dtu.dk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

STRATEGIES FOR SOLVING COMPLEX PROBLEMS IN ENGINEERING DESIGN

W Lewis, J Weir, and B Field

Keywords : Complexity management, design strategy, original design.

1 Introduction

This paper focuses on one of the recurring themes of engineering design practice and research, namely the complexity of design problems and the methods by which engineering designers respond to and cope with the heavy cognitive loads imposed on them during the problem solving process. In a recent paper Kermode and Sivaloganathan [1] pointed out that "The pressure to generate better designs in less time means that designers must be able to handle complexity effectively". In their analysis of parameters characterising problem solving in engineering design, Samuel and Weir [2] commented that "Abstract combinatorial reasoning in design provides the greatest leverage to the problem solving process". But designers are human and constrained in their thinking by the limitations of the human information processing system [3]. The system is basically serial in operation, and limited memory structures may give rise to difficulty in executing combinatorial strategies in problem solving [3, pp. 35, 53].

The opportunity to initiate the investigation on which this paper is based arose from an industry/university liaison activity in which the senior author was briefed by a local design engineer on a problem his team were currently facing. The brief defined a complex problem (hereafter referred to as the "case problem") in the design of part of a rotary drier, basically a large rotating drum for drying powdered process material in a chemical plant. The problem was presented subsequently to three professional engineering designers working individually and to undergraduate design project groups in the University of Melbourne, with arrangements for observing and recording their responses in the form of design diaries and written reports. This was a pilot investigation, the first stage in a larger programme of research into cognitive aspects of designer behaviour.

Broadly speaking, we want to know how designers handle the complexity which is an inherent part of their work. Can patterns of behaviour be observed which are of interest and significance in their own right as well as capable of generalisation from the specific case studied to a wider class of design problems ? In the context of these general concerns the specific aims for this investigation were formulated as follows.

(a) To determine the distinctive features of the responses of professional designers and engineering student designers to the targeted complex design problem.
(b) To analyse the results from (a) in the light of the characteristics of the problem, and examine implications for design research and education.
(c) To review the research outcomes and draw appropriate conclusions.

2 The design problem

2.1 Administration

The brief provided by the local design engineer (see Appendix A) was presented to :
(a) three experienced professionals, designated here as A, B and C;
(b) groups of engineering students in the third year of a four year undergraduate course.
All of the professionals had had 10 to 15 years experience at responsible levels in their fields.
A - Product design and development and related R & D, has 8 patents to his credit.
B - Design of capital equipment, known as thoughtful designer and abstract thinker.
C - Experienced in variant design, development of established ideas, good analyst.

The professionals were asked to solve the problem in their own time and keep written records of their work. They were given considerable leeway in the compilation of these records so that they could adopt whatever method of record keeping they felt most at home with and which gave rise to minimum cognitive strain. Students worked in teams of four and the case problem formed one of two projects undertaken by them in the second semester of the academic year 2000, this as part of their professionally oriented practical work in mechanical engineering. They were instructed to keep design journals and advised to follow Nagy et al.'s format of Issue - Proposals - Argument - Decision [4], although this format was not mandatory. Times available to the student teams were ten hours per student of supervised design laboratory time and an equivalent time outside formal classes. The professionals did not work to set times, but in the event each spent about ten hours on their response.

2.2 Characteristics of Case Problem

Methods of classifying problems in engineering have been proposed by a number of workers, e.g. Ullman [5] and Samuel and Weir [2] at a general level, and Griffin [6] and Mills [7] in relation to product development. The characteristics of the case problem are here specified in terms of relevant factors extracted from these references, namely novelty, complexity (the key factor from the point of view of this investigation), and the working environment.

Working environment : The subject of the design activity is a one-off item, a critical component of a major piece of capital equipment in a new chemical plant.
Novelty : The case problem was rated as innovative design by local industry design team. They had no precedents to work from. We would describe it as an example of "incremental innovation" as in Marples [8], where a succession of relatively small changes rationally worked out over a period of time result finally in a significant innovation. But the innovation does not have the "step change" quality of those described by Jewkes et al. [9].
Complexity : Various definitions of the complexity of design problems have been proposed, e.g. [2, 6, 10]. We regard the development of design metrics for problem complexity as an open research question. For this investigation we draw attention to the functional complexity of the case problem, in that a relatively large number of primary functions flow directly from the design brief - 13 according to the list prepared by the authors and set out in Appendix B (relatively large in comparison with published function/means trees for a variety of products, e.g. [11]). Functions are stated in verb-object phrases as recommended by Stone et al. [12]. The resulting Function/Means tree has been constructed, based on the information given in Appendix B [13]. The breadth or span of the tree indicates the number of paths which the designer must explore - in instance 13 - and is therefore proposed as a measure of the cognitive load experienced.

3 Results and discussion

3.1 Features of professionals' responses to case problem

All three professionals engaged in an initial review of the brief supplied to them; they identified gaps in the information provided in the brief (re magnitudes of allowable relative movements and reactive forces) and proceeded to fill those gaps using their own judgement.

Professional A - organised his work on the work on the case problem in five well-defined phases, as follows : (i) appreciation of problem, statements of goal, objectives, key issues; (ii) discussion of first key issue, control of axial end thrust due to internal steam pressure; (iii) discussion of second key issue, control of small movements of non-rotating part of gland assembly; (iv) examination of embodiment problems, moving on to (v) sketching the layout of a proposed gland assembly. Discussion of key issues in (ii) and (iii) closely followed the I-D-A-P format. Notable features of A's response are that he :

- consciously sought out a small number of key issues to organise his thinking;
- identified end thrust as a key issue early in his thinking and created "cap" and "collar" design concepts as major options;
- thought a lot about the problem and developed concepts and important arguments before undertaking any sketching; (This implies heightened powers of visualisation of phenomena and objects in 3D without the need to rely on sketches and models.)
- referred on many occasions to need for access, ease of assembly and disassembly, manufacturability, thus showing high level of awareness of engineering values.

Figure 1 is a copy of a sketch by A indicating one way his ideas could be worked out.

Another point arises in connection with A's work. In his design history A described a method he had devised for generating ideas to solve design problems. Suppose that in a particular case n classes of candidate solutions have been created, denoted $X_1, X_2, \ldots X_i \ldots X_n$. Then in A's method new candidate solutions are deliberately generated in a boolean X and *not-X* way, so that for each X_i a *not-*X_i class is created. This procedure is repeated through a hierarchy of subsolutions to deliver a conceptually diverse set of options for the designer to consider, as indicated in Figure 3, Appendix B, for the case problem.

Professional B - B's design history closely followed Hales' model of the design process [14] with sequential coverage of "task clarification", "conceptual design" and "embodiment design", interspersed with descriptions of his problem solving strategy. Features were :

- Carefully constructed task clarification, rehearsal of information in brief followed by statements of goal, objectives, constraints, plus quick calculation of thermal expansion.
- Conceptual design - focussed on fluid flow passages, separation of fluid streams, sealing. "Consciously attempted to dissociate major design issues" in his thinking.
- Conceptual design - Invented a praphical language of line diagrams for expressing ideas about possible layouts, example in Figure 2. The language had 12 defined symbols covering axisymmetric surfaces, flow directions, relative movements, bearings, seals.
- Conducted a personal brainstorming session and postponed evaluations during it..

Professional C - C's design history traced out the sequence of steps adopted in his approach to reaching a successful design of gland. He compiled a list of all the factors thought to be relevant to the case problem : flow passages, thermal effects, structural rigidity, relative movements, bearings and seals. Order of magnitude calculations checked magnitudes of thermal expansion and deflection of drier under gravity loads. Features of his work were :

- Adoption of a strategy which might be describe as "conceptual hill climbing".
  A preliminary design scheme was sketched to provide a focus for thinking. This scheme was then critiqued against the list of factors, mismatches between the candidate scheme and factors noted and used as input to a revised scheme which was in turn critqued and revised, and so on until all mismatches were removed. (The end thrust issue arose in the first critiquing session.)
- Sketches were made early in problem solving process, this being necessary for implementation of chosen strategy.

Reviewing the work of the professionals, we see a variety of responses comprising different strategies and different priorities. It is difficult to discern the characteristic patterns of behaviour hypothesised in the Introduction. However, if we look forward to the students' work outlined below and compare the professionals to them we find that the professionals

- displayed greater sensitivity to potential subproblems by going beyond the information supplied in the design brief to explore the ramifications of the design problem, i.e. they performed better on the intellectual dimension "cognition of implications" [15];
- showed greater self-awareness in consciously formulating and executing problem solving strategies ; A - identification of key issue, B - line diagrams plus postponement of evaluation, C - conceptual hill- climbing;
- identified the issue of axial end thrust early on in their thinking (an issue overlooked by many student design teams);
- generated more ideas and design concepts, showed greater fluency of ideation.

Professionals coped with the functional complexity of the case problem by focussing on one or two key issues associated with key functions. In the notation of Appendix B, A focussed on function F13 while B and C focussed on functions F1 and F2, thus reducing complexity measured by number of strands at the top of the function/means tree from 13 to 1 or 2. Two aspects of the professionals' behaviour were not foreseen at the outset, namely A's *X/notX* strategy and B's invention of a graphic design language.

3.2 Features of students' responses

I-P-A-D : Results were mixed. Some teams did not use I-P-A-D format all, were unwilling to devote the time required to become familiar with an unfamiliar process. Others applied it in a learning context to educational issues, e.g. Team 11's first issue was "understanding design problem", which covered a lot of things including gaining information from reference books and catalogues. Comment : Prior training in I-P-A-D required, if this format is to be applied in a consistent manner by designers.

Conceptual design : Only a limited number of alternatives were considered by any one team, e.g. four by one leading team (compare 10 by B), indicating lack of suitable idea generating mechanisms. Comment : More practice in idea generation desirable.

Key issues : No early recognition of end thrust/cap-collar issue, so A's approach to managing the case problem not used by any student design team. Comment : Students appreciate and solve numerous "structural distillation" exercises earlier in their course [16] but this for solid components. Perhaps there should a series of special exercises emphasising the behaviour of fluids in engineering systems, giving practice in the ability to visualise and graphically manipulate fluid forces.

Delay avoidance : There was a tendency to postpone decisions even when all relevant information was to hand, e.g. one student team's "decision" to make (unspecified) calculations concerning a seal before deciding on the type of seal to be used - a sort of delay avoidance behaviour not to be encouraged, and not tolerated in professional work..

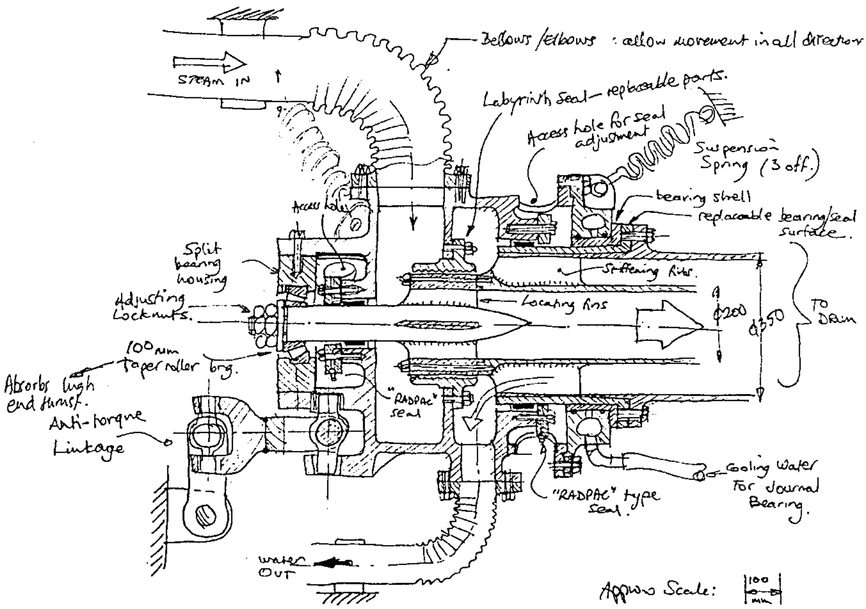


Figure 1. Professional A - sketch of a proposed gland assembly

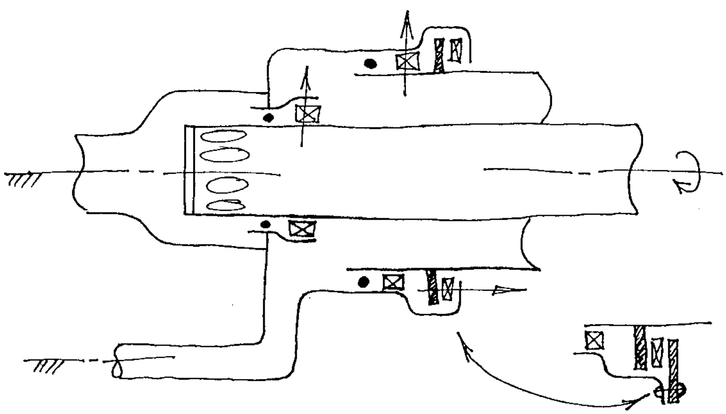


Figure 2. Professional B - line diagram of a possible layout

4 Conclusion

A case problem in engineering design has been presented and evidence adduced regarding its key characteristics of novelty and complexity, together with information on the working environment in which it arose. Significant features of the responses of both professional and student designers to the case problem have been reported.

A method of classifying design problems is required which reflects the cognitive load imposed on the designer during solution of the problem. A numerical measure of design problem complexity based on function structure has been proposed. Further research is required to refine this approach and to assess its applicability across a range of problems.

To cope with the complexity of the case problem the professionals devised strategies which focussed on a limited number of the many functions the gland assembly had to perform, functions which in their judgement were critical to managing the design activity. The emergence of two unforeseen outcomes - A's *X/notX* strategy and B's graphic design language - emphasised the personal nature of design problem solving. Future research must be planned in such a way that it allows such distinctively personal behaviour to be recorded.

Analysis of the students' responses in the light of these comparisons led to sharper definition of educational objectives - extending the scope of studies in structural distillation, and giving more practice in mechanisms of idea generation and decision making.

References

[1] Kermode, G.R. and Sivaloganathan, S., "Complexity Resolution for Design Excellence", Proc. EDC 2000, Brunel Uiversity, London, 2000, pp.397-406

[2] Samuel, A.E. and Weir, J.G., "A Parametric Approach to Problem Enformulation in Mechanical Design", Proc. ICED '97, Tampere, 1997, pp.3/83-3/90

[3] Simon, H.A., "The Sciences of the Artificial", M.I.T. Press, Cambridge, Mass., 1991

[4] Nagy, R.L., Ullman, D.G. and Dietterich, T.G. "A Data Representation for Collaborative Mechanical Design", Research In Engineering Design, 3, 1992, pp.233-242

[5] Ullman, D.G., "A Taxonomy for Mechanical Design", Research in Engineering Design, 3, 1992, pp.179-189

[6] Griffin, A., "Metrics for Measuring Product Development Cycle Time", Journal of Product Innovation Management, 10, 1993, pp.112-125

[7] Mills, J.J., "A Taxonomy of the Product Realization Process Environment", Research in Engineering Design, 4, 1993, pp.203-213

[8] Marples, D.L., "The Decisions of Engineering Design", I.E.E.E. Transactions on Engineering Management, Vol. EM-8, 1961, 55-71

[9] Jewkes, J., Sawers, D. and Stillerman, R., "The Sources of Invention" (2nd edition), Macmillan, London, 1969

[10] Burns, J., Barclay, I. and Poolton, J., "A Structured Methodology for Implementing Concurrent Engineering", in C.J. Backhouse And N.J. Brookes (eds.), Concurrent Engineering : What's Working Where, Gower Publishing, Aldershot, 1996

[11] Perera, I.P., Andrews, P.T.J. and Sivaloganathan, S., "Development of a Modular Design Methodology", Proc. EDC 2000, Brunel University, London, 2000, pp.261-269

[12] Stone, R.B., Wood, K.L. and Crawford, R.H., "A Heuristic Method for Identifying Modules for Product Architectures", Design Studies, 21, 2000, pp.5-31

[13] Lewis, W.P., Weir, J.G. and Field, B.W., "Investigation into a Complex Problem in Mechanical Design", Engineering Design Report, University Of Melbourne, 2000

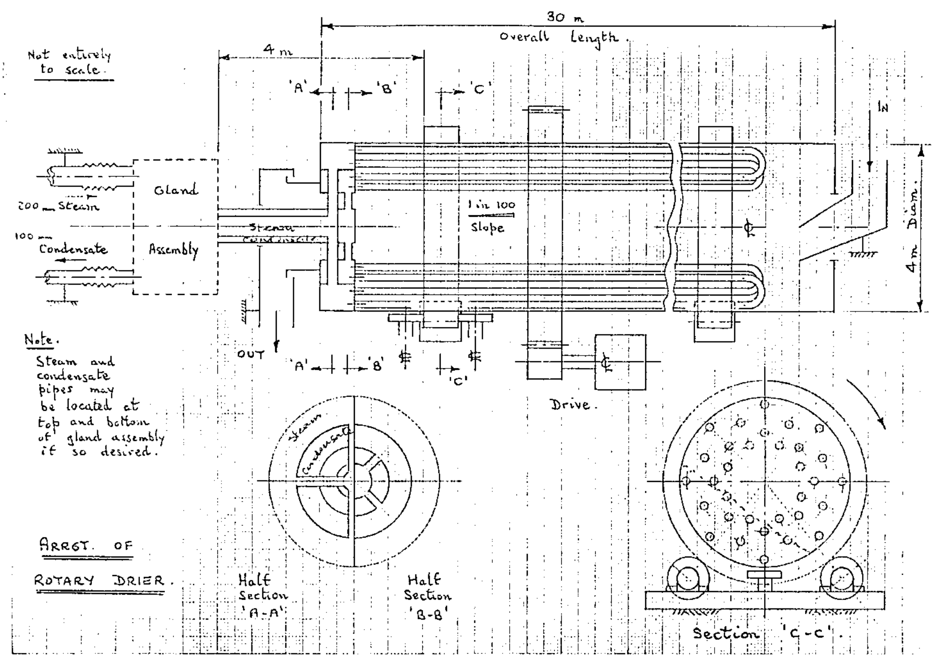
[14] Hales, C., "Analysis of the Engineering Design Process in an Industrial Context" (2nd edition), Gants Hill Publications, Eastleigh, U.K., 1991

[15] Guilford, J.P., "The Nature of Human Intelligence", McGraw Hill, New York, 1967

[16] Samuel. A.E. and Weir, J.G., "Introduction to Engineering Design : Modelling, Synthesis and Problem Solving Strategies", Butterworth-Heinemann, Oxford, 1999

APPENDIX A - DESIGN BRIEF FOR CASE PROBLEM

The accompanying sketch shows the essentials of a large rotary drier for process material. It is heated by saturated steam at 1,500 kPa, and rotates at speeds in the range 5 to 10 r.p.m.



Required : A double gland is to be designed that will allow steam to be fed to, and condensate removed from, the rotating drier. The pressure drop between steam and condensate is very small, around 5 kPa. The drum proper has a 200 mm bore steam inlet on the centreline, surrounded by an annular exit for condensate, of width 50 mm. The design should allow for the following factors.

(a) Thermal movements between cold start up and hot operating conditions.
(b) Due to inevitable but significantly unequal temperatures throughout the drier, it is never perfectly straight or round. Hence the "fixed" part of the gland assembly must be positively located with respect to the rotating part, to ensure that the gland does not leak. However, this means that the "fixed" part will have some small movement relative to the ground and structure supporting the steam supply and condensate mains. For this reason high pressure flexible bellows are provided at these mains. The bellows are adequate to allow for "wobble", and also for change in centreline height of the drier, but not for axial movement.
(c) Some means to relieve the guidance system between "fixed" and rotating parts of most of the weight of the "fixed" part. Some means to restrain the "fixed" part from rotating or moving axially, as the bellows are not designed to take these loads

APPENDIX B - DESIGN STRATEGIES : FUNCTIONS, CONCEPTS

The 13 functions are expressed in verb-object phrases as recommended by Stone et al. [12].

F1 - to transfer steam : gland to drier
F2 - to transfer condensate : drier to gland
F3 - to transfer steam : supply to gland
F4 - to transfer condensate : gland to waste
F5 - to prevent leakage of steam to air
F6 - to prevent leakage of condensate to air
F7 - to prevent leakage of steam to condensate space
F8 - to support weight of gland
F9 - to allow desired relative vertical movement : gland and earth
F10 - to position gland with respect to earth
F11 - to position gland with respect to drier in axial direction
F12 - to position gland with respect to drier, rotational motion
F13 - to support axial end thrust due to steam pressure.

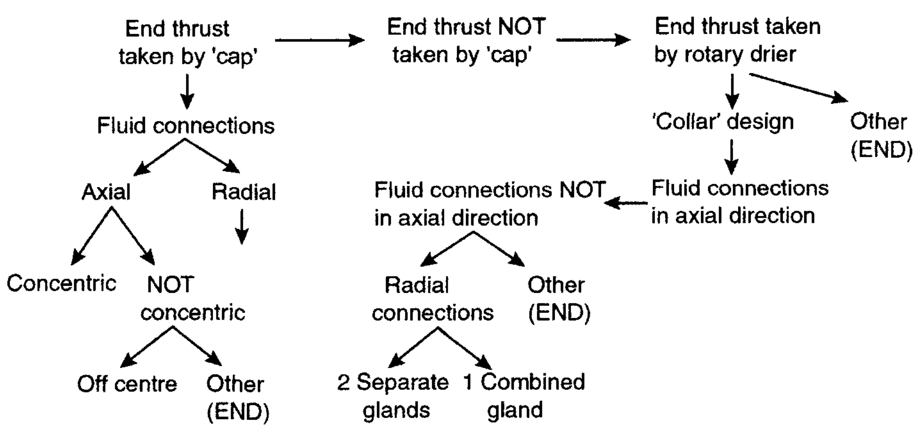


Figure 3. Flow of argument in *X/not X* method

Dr. William Lewis,
Engineering Design Group, University of Melbourne, Parkville, Australia, 3010,
Tel and Fax : 61 3 9596 1105, e-mail : lewis@mame.mu.oz.au

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

TOPOLOGICAL STRUCTURES FOR MODELLING ENGINEERING DESIGN PROCESSES

D Braha and Y Reich

Keywords: mathematical models of design knowledge and processes, process modeling, descriptive models of the design process, collaborative design

1 Introduction

Mathematical formalisms of design have excited many researchers over the years. The ability to formalize a complex problem such as design and solve it using algorithms whose results could be proven and performance guaranteed have been a major attracting force. This excitement has been equally received with skepticism since these ideas have made limited-scope impact on real design (e.g., General Design Theory (GDT) [1,2]).

Most researchers and practitioners, including us, recognize that design processes cannot be fully formalized by mathematical means and cannot be automated. Nevertheless, casting design in mathematical terms serves several goals. First, by developing increasingly better mathematical models of design we improve our understanding about the limit of formalizing design and the limit of automating it. Second, studying mathematical models of design could produce practical guidelines or ideas for implementing design support procedures or systems.

Consider the following design scenario. A designer obtains general specification from a customer. She has several general ideas about addressing it in mind. She studies the specification in relation to the general ideas and refines them. She selects two of the ideas and details them. The analysis that follows uncovers constraints not previously anticipated. These are added to the specification. The analysis results are contrasted against design codes and requirements. New laws for recycling take effect and need to be considered as part of the unfolding specification. Therefore, a redesign process starts until it terminates with an acceptable solution.

Our goal is to develop a mathematical framework that is broad enough to describe various complex and practical design processes including observations such as:

1. In design, the product specification is gradually refined. Better understanding of the specification is a by-product of design.

2. Design progresses by context-dependent activities: Our thinking, alternatives, and decisions are emerging from the situation as it unfolds by bringing diverse knowledge to bear on the present context.

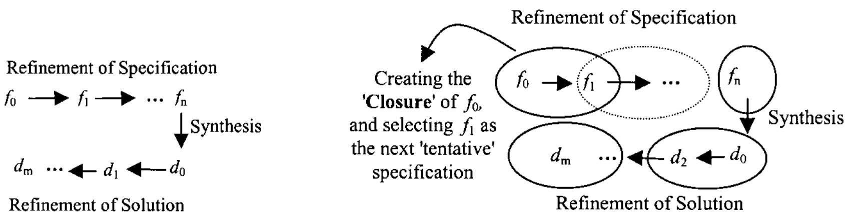
The remainder of the paper is organized as follows: Section 2 provides an overview of the models. Section 3 describes the framework in more details. Section 4 illustrates the benefits of the framework and Section 5 concludes the paper.

2 Overview of the Models

The proposed models clarify and extend the work presented in [3, 4], motivated by empirical observations on how engineers design [e.g., 5, 6], and the nature of complex product design activities. In the following, a basic ideal model for design synthesis is described introducing several key concepts such as *proximity* and *process*. Subsequently, an extended model for design synthesis is introduced that reflects more realistic scenarios.

2.1 Basic Ideal Model

The basic model formulates design as a process that starts from abstract requirements such as customer needs or functional specifications in the *Function Space.* These requirements are iteratively refined[1] by moving to a better "near" specification. At some point, the designer is able to match partial *structural* information, in the *Structure Space*, to the current refined specification. Then, the process continues in the Structure Space by refining the partial structural information until a solution is obtained. This process is depicted in Figure 1(a).



(a) (b)

Figure 1: The basic Ideal model

Roughly speaking, we look for a local exploration procedure that finds a design solution (*terminal state*) with respect to the 'nearness' structure. That is, if f_i is the 'current' ('tentative') specification, then f_{i+1} is a 'near' refined specification if it does not differ substantially from f_i. Given a 'tentative' specification f_i, the local exploration looks for a 'near' refinement of f_i. When no further refinement is possible (e.g., when knowledge is insufficient to proceed), the 'refinement phase' stops, synthesis begins, and a similar refinement process is carried out in the structural domain.

To support local exploration for a specific design problem we need a method for finding the initial description at each phase of refinement (i.e., f_0 or d_0), and a 'nearness' structure of any feasible description. We also need to know how to select among several near items at each refinement step so as to arrive at better solutions. This is done by introducing a 'nearness' structure called *'closure space'* (or *'proximity space'*). For each functional description f (respectively, structural description d) in the Function Space $\mathbf{F}$ (respectively, in the Structure Space $\mathbf{D}$), a *closure* $U_{\mathbf{F}}(f)$ (respectively, $U_{\mathbf{D}}(d)$) is identified. Given two functional descriptions f and g, if $g \in U_{\mathbf{F}}(f)$ we say that 'g *generates* f', or 'f *is generated by* g.' We also define the closure of any *set* of functions. In a logical framework, the closure

[1] The term "refined" loosely mean elaborating the requirements with additional requirements as well as transforming some to more detailed or different requirements. The precise meaning is given in Section 3.

operation is associated with *abduction*. That is, for any two expressions a and b, if $b \in U(a)$ than $b \rightarrow a$. Following the general view of local exploration presented above, the model in Figure 1(a) is elaborated to the model shown in Figure 1(b).

Each circle in Figure 1(b) represents the closure of the current description (functional or structural). In general, the closure of a description would contain more than a single element. A *unique* design process is obtained by selecting at each refinement stage a single generating element out of the many possible ones.

The creation of the closure and the selection of a generating element are knowledge driven. Therefore, the availability, richness, and coherence of knowledge highly influence the ability to obtain quality design solutions. Missing, partial, or otherwise poor quality knowledge will lead to exploring inferior parts of the closure, leaving out the quality promising solutions.

We denote the synthesis operation as a mapping Υ from a functional description to a set of structural descriptions, where each selected structural description may only be described in terms of partial information. The selected structural descriptions correspond to the *output* of a particular synthesis method the designer uses. Our formalism provides the means of representing *several* possible design processes. For instance, the set of *all* design solutions (structural descriptions) that can be obtained from the initial specification f_0 by applying 2 steps of functional refinements, synthesis, and 2 steps of structural refinements can be formally described as follows:

$$U_{\mathbf{D}}(U_{\mathbf{D}}(\Upsilon(U_{\mathbf{F}}(U_{\mathbf{F}}(f_0))))), \qquad (1)$$

Equation 1 encapsulates the set of *all trajectories* that are obtained starting from f_0 and ending with a design solution with the same given number of steps. More generally, different design solutions or trajectories might require different number of design steps. A single trajectory, which defines only one possible candidate development, is a list of 'visited' functional and structural descriptions. A single trajectory can be described by selecting, at each stage, one description from the closure of the current description. In reality there are sometimes few parallel candidate developments.

The ideal process model has several limitations. First, it is too linear. The designer carries out an extensive phase of specification refinement, followed by a "synthesis jump" and an extensive phase of solution refinement (Figure 1(a)). This model is 'ideal' in the sense that 'real design' involves a *repeated interplay* between the function and structure spaces that is not captured by the model [3]. Second, the functional refinement is context independent in that it does not involve a product being designed. To overcome these and other simplifications, we propose an extended model called '*coupled design process*.'

2.2 Real Design Model

The extended model captures the interplay between *design descriptions* (or '*process states*'), each of which is represented by a *pair* of functional and structural descriptions $<f,d>$. This extension which simultaneously handles the function and structure spaces, could be expanded to deal with more complex models that have functional, structural, behavioral, or other design aspects.

Formally, design descriptions are elements of the *Cartesian product* of the function and structure spaces $\mathbf{F} \times \mathbf{D}$, which is called the *design space*. A '*coupled* design process' is performed as follows. The designer starts with the initial design description $<f_0, d_0>$. The f_0 is the initial specification and d_0 is the initial context description; e.g., a product during redesign or improvement or an abstract idea. In general, the designer applies a transition from the design description $<f_i, d_j>$ to $<f_{i+1}, d_{j+1}>$, where both structural and functional descriptions could have been refined. This transition can explain complex processes involving simultaneous refinement of the functional and structural descriptions. It can also capture pure synthesis, which maps *part of* the functional description f_i to a structural description that *augments* the current structural description d_j. In this case, the new functional description f_{i+1} incorporates part of the functional description f_i that remains to be satisfied after the transition as well as previously satisfied functions.

The refinement and synthesis operations are *context dependent*; that is, both f_i and d_j are needed in order to carry out the operations. In addition, $<f_i, d_j>$ may be refined in case some design *constraints* are violated.

To capture the transition operation using our formalism, we simply introduce a 'proximity structure' for the space $\mathbf{F} \times \mathbf{D}$. That is, for each design description $<f_i, d_j>$, a closure operation $U_{\mathbf{F}\times\mathbf{D}}(<f_i, d_j>)$ is identified. Every design description in $U_{\mathbf{F}\times\mathbf{D}}(<f_i, d_j>)$ *generates* the design description $<f_i, d_j>$. The 'coupled design process' is described in Figure 2. Each circle in Figure 2 represents the design descriptions obtained by applying the closure operation. In real design, a *unique* design process is obtained by selecting at each stage a single element out of the many possible design descriptions without creating the complete closure. A design process ends with a design description of the form $<f_n, d_n>$ where f_n is a detailed specification manifesting a deeper understanding of the problem, and d_n is a design description that satisfies this specification and can be realized. Given $<f_n, d_n>$, if d_n satisfies f_n though may not be realized, $<f_n, d_n>$ is not a final state, even though it marks an intermediate favorable situation, which can happen during design stages such as conceptual or detailed design. In these situations designers continue to refine the specification knowing it is still incomplete. We might be able to say that in such a situation conceptual design terminates and another stage starts in a new closure space.

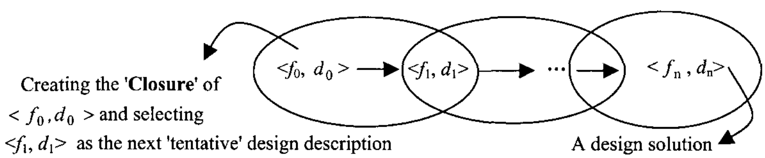


Figure 2: The coupled design space

In the full paper [7], we show that the 'ideal design process' presented in Figure 1 is a special case of the 'real design model' discussed above.

3 Topological Structures for Design

In this section, we present a model that attempts to cast design in the framework of closure spaces more formally. For brevity and clarity, some of the formal definitions and theorems are omitted and concepts are presented intuitively. First, we define the nature of the universes we handle.

Definition 1: A structural description d is defined by its observable or measurable attributes. The attributes may be of several types, e.g., physical or geometrical. A structural description has respective values for its attributes. For designers, the structure space is the set of *all* structural descriptions they can generate with their present knowledge, and is denoted by $\mathbf{D}$.

Definition 2: A functional property is the behavior that an artifact displays when it is subjected to a situation. The collection of all functional properties observed in different situations is the functional description of the artifact. The function space is the set of *all* functional descriptions, and is denoted by $\mathbf{F}$.

In a similar way we could introduce additional aspects beyond $\mathbf{D}$ and $\mathbf{F}$ such as behavior, into the formulation if desired by a particular design approach. The next definition formalizes the intuitive notion of 'proximity' in the structure and function spaces. For brevity, the definitions are provided for the related function space. Similar definitions can be provided for a structure space, or the coupled space discussed in Section 2.

Definition 3 (*Closure Operation and Function Space*): If $U_{\mathbf{F}}$ is a function on $2^{\mathbf{F}}$ with a range of $2^{\mathbf{F}}$, then we say that $U_{\mathbf{F}}$ is a *closure operation* (or "closure") for $\mathbf{F}$ provided that the following conditions are satisfied:

(1) $U_{\mathbf{F}}(\varnothing) = \varnothing$,

(2) $F \subseteq U_{\mathbf{F}}(F)$ for each $F \subseteq \mathbf{F}$, and

(3) $U_{\mathbf{F}}(F \cup G) = U_{\mathbf{F}}(F) \cup U_{\mathbf{F}}(G)$, for each $F \subseteq \mathbf{F}$ and $G \subseteq \mathbf{F}$.

The structure $< \mathbf{F}, U_{\mathbf{F}} >$ is called a *closure function space*. A *closure operation* $U_{\mathbf{D}}$ for the structure space $\mathbf{D}$ is defined in a similar manner. The structure $< \mathbf{D}, U_{\mathbf{D}} >$ is called a *closure structure space*. Note that we are not restricting the source of closure to be logical or mathematical. Rather, in real design, the closure would originate from designers' knowledge.

If $F = \{f\}$, then every functional description *g* in the closure of f (i.e. such that $g \in U_{\mathbf{F}}(f)$) is termed as *generator of* f. Alternatively, we say that *g generates f*, or that *f* is *generated by g*. Generally speaking, the relation 'generated by' is associated with the relation *'refined by'*. That is, following the view presented in Figure 1, the progression from a specification f_i to a refined specification f_{i+1} implies that the specification f_i is *generated by* the refined specification f_{i+1}, or that f_{i+1} *generates* f_i. In formal terms, the refined specification f_{i+1} is included in the closure of f_i.

As discussed in Section 2, in reality there are sometimes several parallel candidate developments starting from the initial specification f_0. Applying Definition 3, this situation may be described as follows. The designer starts by creating the closure of f_0, $U_{\mathbf{F}}(f_0) = F$. Creating the entire closure may be an expensive operation. Therefore, the designer may create *part of* the closure, and make a decision based on the *partial* knowledge of the complete closure. Each specification in *F* generates f_0. While creating the closure, the designer may select any refined specification in the part of *F* thus far created to proceed with. The next refinement step involves finding the closure of the selected refined specification, etc. Thus, *F* denotes the set of *all* possible refined specifications of f_0. Similarly, the closure of *F*,

$U_F(F) = \bigcup_{f \in F} U_F(f)$, denotes the set of *all* possible refined specifications in the second stage, etc. A *trajectory* of a refinement process is created by selecting a single specification from each closure in each refinement stage (Figure 1(b)). In real design, one or several trajectories are explored, and the complete **F** or $U_{\mathbf{F}}(F)$ are not exhaustively created due to limited resources of various kinds such as time, money, knowledge.

Often, solving a design problem is a *collaborative* effort carried out by several designers, each of which possesses knowledge originating from a different source. The notion of a closure may be useful in interpreting some aspects of collaborative design. The following example illustrates a simple collaborative scenario. Assume that there are two designers, Alice and Bob, working collaboratively on a certain problem. Bob and Alice each carry out (independently) few parallel candidate developments based on his/her private knowledge. At some point in the refinement process, Bob and Alice share their intermediate results F_n^{Bob} and F_m^{Alice}, respectively. Each set represents a set of possible refined specifications (starting from f_0). If F_n^{Bob} and F_m^{Alice} intersect, they can continue together to refine the intersection; otherwise, they will have to collaborate together to find refinements within a closure created by their mutual knowledge. The intersection of sets or the closure created by mutual knowledge is more focused than each of the sets or closures separately. This illustrates, by way of topological arguments, the usefulness of collaborative design.

Definition 4 (Closed Sets): A subset F of a closure function space $< \mathbf{F}, U_{\mathbf{F}} >$ is called *closed* if $U_{\mathbf{F}}(F) = F$. Closed sets in a closure structure space $< \mathbf{D}, U_{\mathbf{D}} >$ are defined in a similar manner.

Intuitively, a set F is closed if every element in F is generated *only by* elements in F. As explained above, the set of *all* possible refined specifications (starting from f_0) in a particular stage of refinement can be defined recursively as follows:

1. $F_0 = \{f_0\}$;
2. $F_1 = U_{\mathbf{F}}(F_0)$;
3. $F_{i+1} = U_{\mathbf{F}}(F_i)$.

If at some stage, the derived set of refined specifications F_n is a *closed* set, then we are assured that every specification in F_n is generated only by specifications in F_n. Thus, no further refinement stage that yields *new refined specifications* (or 'new information') is possible. Formally, it means that $U_{\mathbf{F}}(F_n) = F_n$.

In a real situation, this means that there is not enough information to further continue design exploration in the function space. Therefore, the refinement process stops and *synthesis* must begin (Figure 1(a)). Prior to performing the synthesis operation, the designer searches within F_n for a 'suitable' refined specification that can be mapped to structural descriptions in the structure space. The refinement process in the structure space then begins. It is important to note that *not every* refinement process will end with a closed set in a *finite number* of steps. But, if at some point a closed set is arrived at, then the refinement process stops.

As stated above, the set of *all* possible refined specifications (starting from f_0) at any particular stage of refinement can be defined recursively as $F_{i+1} = U_{\mathbf{F}}(F_i)$. It can be shown that *after enough time* (even an infinite number of steps) repeated application of the closure operation $U_{\mathbf{F}}$ will result in a closed set. That is, $\lim_{i \to \infty} F_i$ is a closed set. Therefore, every refinement process *must* terminate at some point (though not necessarily at a usable or good specification).

Note that computationally, it could be prohibitively expensive to generate the closure or check that a set is closed. Therefore, we are developing a concept of an approximate closure that can be a model of real imperfect and partially incorrect knowledge. Approximate closure could be used to explain design failures whereas limited resources could explain obtaining sub-optimal designs. Studying the nature of approximate closures is left for future work.

4 Putting the framework to use

In order to illustrate the descriptive usefulness of the framework we need to show that it can be used to describe diverse approaches. In this paper we report on three results, leaving their details and other examples to future papers.

The first example shows that General Design Theory (GDT, see [1, 2]) is a special case of our framework. General Design Theory (GDT) is defined in terms of the set of all real artifacts that existed, exist presently, and that will exist in the future **S**, and a collection $\Im$ of subsets (called *abstract concepts*) of the space **S** that satisfy the axioms of point-set topology [2]. Let us call <**S**, $\Im$> the Yoshikawa-Tomiyama space (**Y-T** space). According to GDT, a design process is a mapping between a **Y-T** function space to a **Y-T** structure space [2]. In the full paper [7], we show that for any **Y-T** space there corresponds a unique closure space, called topological closure space. This quality makes closure spaces more general than **Y-T** spaces.

The second result shows that any design system implemented as a rule-based system can be described in our framework. The forward chaining corresponds to analysis whereas the backward chaining to synthesis.

The third example demonstrates that ECOBWEB [8], a learning system that creates a classification hierarchy from design examples, can be illustrated in our framework. In particular, the hierarchy created by learning from examples can be interpreted as a sequence of closures. The closure operation could be defined as if $< f_i, d_j >$ is the present description than its closure $< f_{i+1}, d_{j+1} >$ includes the properties in $< f_i, d_j >$ with several additional properties. ECOBWEB only approximates this closure (see section 3.4), therefore, its synthesis success is not guaranteed at all times.

5 Conclusions

We have described a mathematical framework of design processes based on closure spaces. In the full paper [7], we prove that our framework is more general than General Design Theory. In addition, we utilize the framework to describe the operation of rule-based systems, and a particular learning system. These results demonstrate that our model has the desirable quality of representing several, seemingly distinct approaches, as instances of the same framework.

We believe that the proposed framework has the potential to serve as a basis for a descriptive study of design. For instance, in the full paper [7], various design phenomena such as design failure, identification of design knowledge bottlenecks, as well as the benefits of collaborative design are interpreted using the framework. It is important to note, however, that our framework is not suggested as a means for supporting or proving theorems about automated design.

In the future, we intend to address the following issues: 1) demonstrating the utility of the proposed framework in classifying and analyzing different design methodologies, such as conceptual design with a 'function-structure' in Pahl and Beitz [9]; 2) showing the utility of our framework in describing collaborative design, and expanding on the consequences of design robustness; and 3) exploiting the computational aspects (considering realistic approximate procedures) of the underlying topological concepts (e.g., closure and neighborhood).

References

[1] Tomiyama, T., and Yoshikawa, H., "Extended General Design Theory," in *Design Theory for CAD*, Proceedings from IFIP WG 5.2, Amsterdam, 1987.

[2] Reich, Y. "A Critical Review of General Design Theory," *Research in Engineering Design*, 7(1), pp. 1-18, 1995.

[3] Braha D. and O. Maimon, *A Mathematical Theory of Design: Foundations, Algorithms, and Applications*, Kluwer Academic Publishers, Boston, 1998.

[4] Maimon O. and D. Braha, "On the Complexity of the Design Synthesis Problem," *IEEE Transactions on Systems, Man, and Cybernetics*, 26 (1), 1996.

[5] Hales, C., *Analysis of the engineering design process in an industrial context*," Dissertation, University of Cambridge, Gants Hill Publications, Hampshire, 1987.

[6] Blessing, L. T. M., "*A Process-Based Approach to Computer-Supported Engineering Design*," 1994.

[7] Braha, D., and Reich, Y., "Topological Structures for Modeling Engineering Design Processes," Technical Report, Computational Systems Lab, Tel Aviv University, 2001.

[8] Reich, Y. and Fenves, S. J., "Inductive learning of synthesis knowledge," *International Journal of Expert Systems: Research and Applications*, 5(4), pp. 275-297, 1992.

[9] Pahl, G., and Beitz, W., *Engineering Design – A Systematic Approach*, 2nd Ed., Springer, London, 1996.

Prof. Yoram Reich
Department of Solid Mechanics, Materials and Systems, Faculty of Engineering, Tel Aviv University, Tel Aviv 69978, Israel
Phone: +972-3-6407385
Fax: +972-3-6407617
E-mail: yoram@eng.tau.ac.il

Dan Braha
Visiting Professor, Sloan School, Massachusetts Institute of Technology
Department of Industrial Engineering, Ben-Gurion University, Israel
E-mail: brahad@bgumail.bgu.ac.il

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

TOWARDS A THEORY OF COMPLICATEDNESS – FRAMEWORK FOR COMPLEX SYSTEMS ANALYSIS AND DESIGN

V Tang and V K Salminen

Keywords: theory of technical systems, complexity management, adaptive design

1. Introduction

Global, dynamic, and competitive business environment has increased the complexity in product, service, operational processes and human side. Much engineering effort goes into reducing systems complexity. We argue that the real issue is reducing complicatedness. This is an important distinction. Complexity can be a desirable property of systems provided it is *architected complexity* that reduces *complicatedness.* Complexity and complicatedness are not synonyms. Complexity is an inherent property of systems; complicatedness is a derived function of complexity. We introduce the notion of complicatedness of complex systems, present equations for each and show they are separate and distinct properties. To make these ideas actionable, we present a design methodology to address complicatedness. We show examples and discuss how our equations reflect the fundamental behavior of complex systems and how our equations are consistent with our intuition and system design experience. We discuss validation experiments with global firms and address potential areas for further research. We close with a discussion of the implications for systems design engineers. As engineers, we believe our strongest contributions are to the analysis, design, and managerial practice of complex systems analysis and design.

We illustrate the difference between complexity and complicatedness. Relative to a manual transmission, a car's automatic transmission has more parts and more intricate linkages. It is more *complex.* To drivers, it is unquestionably less *complicated*, but to mechanics who have to fix it, it is more *complicated.* This illustrates a fundamental fact about systems; *decision units* act on systems to manage their behavior. Complexity is an inherent property of systems. Complicatedness is a derived property that characterizes an execution unit's ability to manage a complex system. A system of complexity level, $\mathbf{C_a}$, may present different degrees of complicatedness, $\mathbf{K}$, to distinct execution units **E** and **F**; $\mathbf{K_E}=\mathbf{K_E(C_a)} \neq \mathbf{K_F}=\mathbf{K_F(C_a)}$.

We summarize relevant literature on systems complexity in Figure 1. Columns [1] to [15] are keyed to the references. Rows identify key areas of research results; e.g., metrics, complicatedness, etc. We make four observations about the locus of results. One, there is a dearth of quantitative frameworks or metrics. There is no research on complicatedness and complexity as distinct properties of systems. Two, research seems to cluster around engineering management and physical products. The focus is on modularization and interactions with a bias to linear systems and qualitative metrics. Three, there are efforts on methodologies and tools, but theory, foundations and software have a demonstrably lesser presence. Ferdinand's work in software systems complexity is a happy exception [1]. It is analytical, rigorous and elegant. Three, services and enterprise solutions are barely addressed.

This is a serious omission given the high proportion of services in industrialized economies. Fourth, although layering of abstract systems and reintegration have a long history; the literature is skewed to decomposition rather than integration.

Reference	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]
Main Focus of Investigation															
Foundations & Theory	■	□	□	□	□	□	□	■	■	□	■	■	□	□	□
Linear Systems Engineering	■	□	■	□	■	□	□	■	□	■	■	■	□	□	■
Non-linear Systems Engineering	□	■	□	□	□	■	□	□	■	□	□	□	■	■	□
Systems Architecture/Structure	■	□	□	□	■	■	■	■	■	■	■	■	■	■	■
Management	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
Design Methodology	□	□	□	□	■	□	■	■	□	■	■	■	■	■	■
Design Tools	□	□	□	□	□	□	■	□	□	□	□	□	□	□	■
Complicatedness	■	□	□	□	□	□	□	□	□	□	□	□	■	□	□
Strategy to Address Complexity															
Modularization & decomposition	■	□	■	□	■	■	■	■	□	■	■	■	■	■	■
Interactions and Dependencies	■	■	■	□	■	■	■	■	■	■	■	■	■	■	■
Layering & Abstraction	■	□	□	□	□	■	□	□	□	■	□	□	■	□	□
Integration	□	□	□	□	□	□	■	□	□	□	□	□	■	□	□
Complexity Metrics															
Quantitative	■	□	□	□	□	□	□	■	□	□	□	□	□	□	■
Qualitative	□	■	■	■	■	■	■	□	■	■	■	■	■	■	■
Domain															
Physical Products & Systems	□	□	■	□	■	■	■	■	■	■	■	■	■	■	■
Software Products & Systems	■	□	□	□	■	□	□	□	□	□	□	□	■	□	□
Services	□	□	□	□	□	■	□	□	□	□	□	□	■	□	□
Enterprise Solutions	□	□	□	□	□	□	□	□	□	□	□	□	■	□	□
Social & Organizational	□	■	□	■	■	□	□	□	□	□	□	□	□	■	■
Applications															
Engineering	□	□	■	□	□	■	■	■	■	■	■	■	■	■	■
Organizational Theory	□	■	□	■	■	□	□	□	□	□	□	□	□	■	□
Quality Management	■	□	□	□	□	□	□	□	□	□	■	□	□	□	□

■ Indicates a strong element in the publication. □ Indicates a lesser or absent element.

Figure 1. Systems Complexity Summary of the Literature

2. Complexity

Overwhelmingly, the literature considers a system with a large number of elements as complex. Very few address the linkages among the elements and no one, to our knowledge, considers their bandwidth. All these factors are inherent characteristics of systems. Therefore, we argue that the number of elements, the number of *interactions* among them and the *bandwidth* of these interactions determine complexity of the system. As any of these increases, we expect complexity to increase. For example, a system $\mathbf{N}=\{\mathbf{n}_i\}_{i=1,2,\ldots,p}$ with binary interactions among the elements. Complexity, $\mathbf{C}_N$, of this system does not exceed p^2, we denote this by $\mathbf{C}_N=\mathbf{O}(p^2)$. System $\mathbf{M}=\{\mathbf{m}_j\}_{j=1,2,\ldots,p}$ can have complexity $\mathbf{C}_M=\mathbf{O}(p^k)$ where $k>2$. When $\mathbf{M}$ admits $\{\mathbf{m}_j \mathbf{x} \mathbf{m}_r\}_{jr}$ and $\{\mathbf{m}_j \mathbf{x} \mathbf{m}_r \mathbf{x} \mathbf{m}_s\}_{jrs}$ interactions, $\mathbf{C}_M=\mathbf{O}(p^3)$. If $\mathbf{M}$ admits $\{\mathbf{m}_j \mathbf{x} \mathbf{m}_r \mathbf{x} \mathbf{m}_s \mathbf{x} \mathbf{m}_t\}_{jrst}$ interactions, $\mathbf{C}_M=\mathbf{O}(p^4)$. This characterization of complex systems admits systems with feedback loops of arbitrary nesting and depth, and high bandwidth interactions among system elements. Complexity is a monotonically increasing function as the size of the system size, number of interactions increases, and bandwidth of interactions increase. In the limit, complexity $\rightarrow \infty$. We define complexity by $\mathbf{C}=\mathbf{X}^n\Sigma_b\,\mathbf{B}_b$

where $\mathbf{X}$ is an integer denoting the number of elements $\{x_e\}_{e=1,\ldots,p}$

n is the integer indicated in the relation $O(p^n)$

and $\mathbf{B}_1=\Sigma_{ij}\lambda_{ij}\beta_{ij}$

λ_{ij} is the number of linkages between x_i and x_j

β_{ij} is the bandwidth of the linkages between x_i and x_j

$\mathbf{B}_2=\Sigma_k\lambda_k^{ij}\,\beta_k^{ij}$

λ_k^{ij} is the number of linkages between x_k and (x_i,x_j)

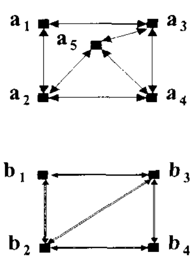
β_k^{ij} is the bandwidth of the linkages between x_k and (x_i,x_j) and in general,

$\mathbf{B}_n=\Sigma_n\lambda_p^{ijk\ldots n-1}\,\beta_n^{ijkl\ldots n-1}$

$\lambda_n^{ijkl\ldots n-1}$ number of linkages among x_k and $(x_i,x_j),(x_i,x_j,x_k),\ldots,(x_i,x_j,x_k,x_k,\ldots,x_{n-1})$

$\beta_n^{ijkl\ldots n-1}$ linkage bandwidth among x_k and $(x_i,x_j),(x_i,x_j,x_k),\ldots,(x_i,x_j,x_k,x_k,\ldots,x_{n-1})$

B is a measure of the information capacity among the elements of the system. Note that the monotinicity properties are not violated. In Figure 2 we give an example.



$A=\{a_1,a_2,a_3,a_4,a_5\}$

β_{ij}=bandwidth between $\{a_i,a_j\}$

Let $\beta_{ij}=1$ bit

$B=\Sigma_i\,\Sigma_j\,\lambda_{ij}\,\beta ij=7$

$C=X^nB^n=(5)^2(7)^2=1225$

$B=\{b_1,b_2,b_3,b_4\}$

β_{ij}=bandwidth between $\{b_i,b_j\}$

Let $\beta_{ij}=2$ bits

$B=\Sigma_i\,\Sigma_j\,\lambda_{ij}\,\beta_{ij=}10$

$C=X^nB^{n=}(4)^2(10)^2=1600$

Figure 2. Complexity Example of Two Systems

3. Complicatedness

Complicatedness is the degree to which a decision unit for the system is able to manage the level of complexity presented by the system. The decision unit can be another system or a person. Complicatedness is a function of complexity, $\mathbf{K}=\mathbf{K}(\mathbf{C})$. Let us explore the properties we expect from a complicatedness function. We expect monotonicity of complexity is imposed on complicatedness, but do not expect that they are identical. Clearly at $\mathbf{C}=0$, $\mathbf{K}=0$.

Consider **K** when $\mathbf{C}\rightarrow\infty$. Intuitively, there is a level of complexity at which the decision unit can barely cope with the system. The system is becoming unmanageable. For example, most people can visualize a graph, $\mathbf{g}=\mathbf{g}(x,y)$ of $\mathbf{C}_g=\mathbf{O}(p^2)$, but it is harder for $h=h(x,y,z)$ with $\mathbf{C}_h=\mathbf{O}(p^3)$. Few can visualize a surface four variables, although complexity has only reached $\mathbf{O}(p^4)$. Consider, equally incomprehensible systems **A** and **B** where $\mathbf{C}_A=\mathbf{O}(p^{100})$ and

$C_B=O(p^{100,000})$ respectively; $K_A \gtrsim K_B$ although $O(p^{100,000}) >> O(p^{10,000})$. Therefore, when $C=0$, $K=0$ and when $C \to \infty$, $K \to K_{max}$ asymptotically.

Systems are designed to operate and be managed around at an optimal point of complexity, say C^*. For $C<C^*$, although complexity increases, it is well within the interval of manageability. At $C=C^*$ the system complexity is optimal for the decision unit. For $C>C^*$, complexity is increasing and the decision unit can manage the system with decelerating effectiveness. Mathematically, $dK/dC>0$ in the open interval $(0, \infty)$. At $C=C^*$, $dK/dC=0$ and $d^2K/dC^2=0$. Complicatedness has reached an inflection point. So that for $C>C^*$, $d^2K/dC^2<0$, i.e., complicatedness is reaching saturation. The decision unit's ability to manage complexity has reached diminishing returns. For $C<C^*$, $d^2K/dC^2>0$, complexity is growing faster than complicatedness. Because the logistic function is one of the simplest mathematical expressions that has all the above properties, figure 3. We adopt it to express complicatedness. $K(C)=K_{max}/(1+e^{-\alpha C})$

where e is the transcendental number e=3.2718 2818 284...
α is a constant specific to the decision unit
C is the complexity of the system

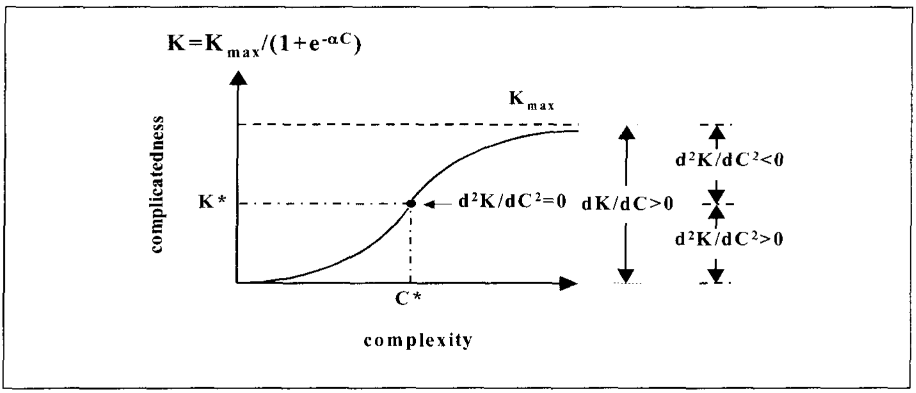


Figure 3. Complexity and Complicatedness

Without loss of generality, we set $K_{max}=1$ to indicate abject complicatedness. There are other functions that can be used; such as, the Gompertz curve, Weibull distribution, log-reciprocal function, etc. The major differences are the location of the inflection point, the growth pattern before and after the inflection point, and the symmetry around the inflection point.

4. Examples of Uncomplicated Complex Systems

Earlier we presented the automobile transmission as a complex system that is uncomplicated. Neural networks are more interesting as a systems engineering example. Typically they are applied to situations where there are an intractable number of data points to analyze in order to set a course of action. To solve this difficulty, the neural network is layered, Figure 4. The complexity has increased relative to the input vector. Many new elements, new interactions and their bandwidth have all increased the initial complexity. But *architected complexity* has reduced an intractably complicated input vector to an output vector that now makes the system manageable. This approach has proven effective for engineering paper machines [16]. This is a non-trivial example. The purchase price of paper machine ranges around $50 M. The mill generates about 10^9 data points, which are processed in real-time by adaptive and distributed neural networks embedded in the machine.

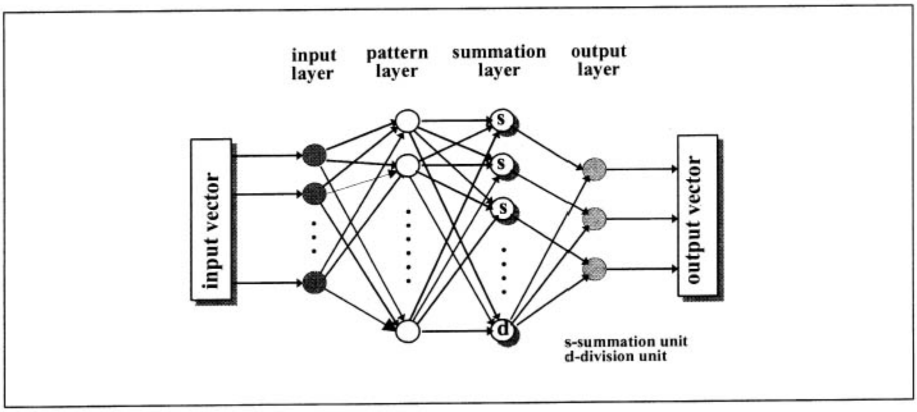


Figure 4. Use of Neural Network as Architected Complexity

The telecomm infrastructure is one of the most massive systems in the world. On demand, it interoperates an immense array of networks, products and computers. The system complexity is enormous, yet we routinely make transcontinental telephone calls and download music and pictures from the Web. *Architected complexity* has made telecomm networks manageable. Engineers created the OSI Reference Model by partitioning the network functions into distinct layers. This architectural innovation creates, at each level, a distinct presentation of the network that is more abstract at each successive layer. Each layer presents to decision units a specific *system image* of the network that is vastly less complicated. Layering system images is a widely adopted doctrine in computers; e.g. programming languages. With the first computers, applications programming was very difficult. Programmers had to embed arcane hardware details into their algorithms. High Level Languages were invented to present an abstract, but domain specific, system image for programming. A layer of software hid and encapsulated, transparently to the programmer, all machine specificities. *Architected complexity* is a very effective complexity management strategy; it reduces complicatedness.

5. Examples of Complicated Complexity

It suffices to present three examples. The typical VCR control panel is a classic example of complex and complicated design. Another example is PC software or "bloatware." So many application packages are functionally so extravagant that the average person can learn only a fraction of their functionality. Cellular phones are in danger of becoming examples of complex and complicated products.

6. Calibrating Complicatedness

Consider a car's transmission. To drivers, the automatic transmission presents the well known gear shift *system image* of **A**={P,R,N,D1,D2,D3}, where λ_{ij}=24 with β_{ij}=1; thus $\mathbf{C_A}=6^2(24)(1)=864$. On the other hand, a manual transmission presents the more complicated *system image* of **M**={P,R,N,D1,D2,D3,F} where F is the foot clutch. It needs to be engaged and disengaged, so F's interaction bandwidth is 2. Thus, λ_{ij}=10 with β_{ij}=1, and λ_{mn}=14 with β_{mn}=2, thus $\mathbf{C_M}=7^2[10+(14)2]^2=38416$. For the novice driver, $\mathbf{C}^* \approx \mathbf{C_A}=864$. At $C \approx 40000$, we can say that $\mathbf{K_{max}}=1$. Therefore, we can create instruments to determine the analytic form of the complicatedness function. For a system with complexity C, and a decision unit **K**, we can readily design questionnaires or interviews that provide sufficient data for designers to:

[1] determine the optimal complexity, **C*** that **K** can manage optimally
[2] in the (C,K) space, at **C*** set $\mathbf{K}^*=1/2$
[3] solve for α using equation $1/2=1/(1+e^{-\alpha C^*})$, recall that and $\mathbf{K_{max}}=1$
[4] get $\mathbf{K(C)}=1/(1+e^{-\alpha C})$.

7. Engineering Complex but Uncomplicated Systems

There is good and bad cholesterol. Similarly, there is *architected* and *unarchitected complexity*. The former reduces complicatedness; the latter does not. There are two important principles in *architected complexity:* partition the system into modules, reintegrate them while maintaining system integrity. Many decomposition schemes address the first principle. Karnaugh maps for digital circuits, Djysktra architectures for computers, Design Structure Matrix for mechanical products [15], etc. They are effective tools, but when the decomposition creates a large number of new components and interactions, the result can become intolerably complicated and make reintegration impractical. Reintegration is less visible in research, although widely practiced by engineers. Consider system $\mathbf{M}=\{m_j\}_j$, $C_M>K_M^*$ for **M**'s decision unit, figure 5.

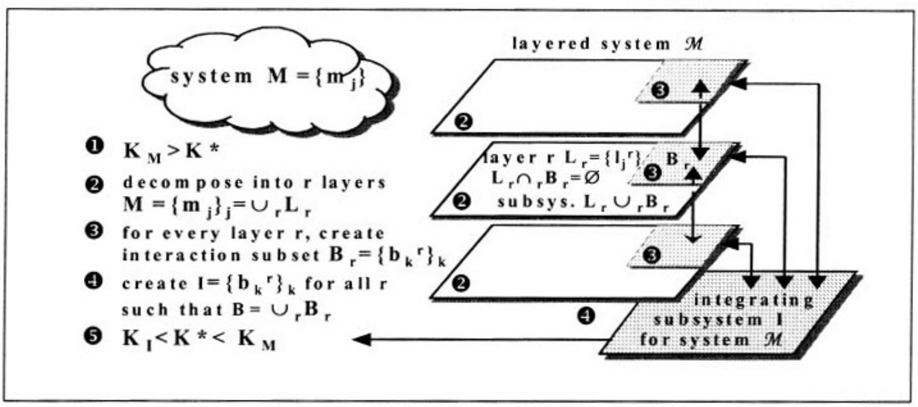


Figure 5. Architected Complexity to reduce Complicatedness

The goal is to architect complexity so that **M** is transformed into $\mathcal{M}$, such that $K_{\mathcal{M}}<K_{M^*}$, although $C_{\mathcal{M}}>C_{M^*}$. Partition **M** into layers, $\mathbf{L_r}=\{\mathbf{l}_j^r\}_j$ such that $\mathbf{M}=\cup_r\mathbf{L_r}$, i.e., all the elements of

M appear in some specific layer. Functional decomposition is an engineering application of this principle. For a paper mill, these can be the mechanical, control, and process domains [13], or for a computer, the arithmetic unit, main memory, the I/O units, etc. Design the layers so that there are only intra level interactions among the elements of a layer. Create $\mathbf{B}_r=\{\mathbf{b}_k^r\}_k$ for ever layer $\mathbf{L}_r$, so that $\mathbf{L}_r \cap_r \mathbf{B}_r=\varnothing$. Design **B** so that only elements of **B** communicate with each other. $\mathbf{B}=\cup_r \mathbf{B}_r$ is $\mathcal{M}$'s communications subsystem. For $\mathcal{M}$, design a system integration unit $\mathbf{T}=\{t_x\}_x$, which on one side interfaces with **B** and the other with the decision unit. Note that T presents the decision unit with an *image* of the system $\mathcal{M}$. This is a hallmark of a good architecture. Good design always presents a less complicated *system image* to a decision unit. For complex computer systems, an uncomplicated system image is important for market and customer acceptance [17].

8. Areas of Potential Research

Complexity should be studied further with industry experiments. Begin by selecting a set of simple systems with unambiguously matched by an identifiable class of decision units. Then calculate the systems' complexity and derive complicatedness functions for the set of decision units. These experiments would serve as case studies for the behavior of the complicatedness function and help determine whether our simple logistics function serves its purpose well. If not, different analytic functions should be tried as suggested earlier. In addition, experiments in an organizational setting should also be studied. The decision units are people with specific complicatedness functions. An executive who must negotiate with a large number of departments is such an example. In this case the system elements are the department heads, the bandwidth of interactions is determined by the specific nature of the negotiations. The extent to which organizational structures, communication styles and boundary objects is effective *architected complexity* are fruitful areas of investigation [18].

9. Conclusions

Separating complicatedness and complexity improves the clarity by which systems can be described and analyzed. In this way we can clearly separate what is an inherent property of the system, complexity, from a derived attribute, which is complicatedness. The mathematical expressions we formulate capture additional properties of systems that have heretofore remained largely unaddressed. We are able to derive results that give us valuable insight into the behavior of systems. These insights are useful in the analysis and design of very large complex systems, and also move us towards a theory of complicatedness.

References

[1] Ferdinand, A.E., Systems, Software and Quality Engineering. New York: Van Nostrand Reinhold, 1993.

[2] Kelly, S. and Allison, M-A, The Complexity Advantage. Business Week Books, McGraw-Hill, 1998.

[3] Khurana, A., Managing Complex Production Processes. Sloan Management Review, Winter 1999, pp.85-97.

[4] Bar-Yam, Y., New England Complex Systems Institute (NECSI) home page: http://www.necsi.org/

[5] Levitt, R., Thomsen, J., Christiansen, T., Kunz, J., Jin, Y., Nass, C., Simulating Project Work Processes and Organizations: Towards a Micro-Contingency Theory of Organizational Design. Management Science, Vol. 45, No.11, November 1999, pp.1479-1495.

[6] Salminen, V. and Pillai, B., Non-linear Dynamics of Interacting Mechanisms in Distributed Product Development. International Conference on Complex Systems (ICCS), May 21-26, 2000, Nashua, NH.

[7] Salminen, V., Yassine, A., Riitahuhta, A., Strategic Management of Complexity in Distributed Product Development, NordDesign 2000, 24-25th August 2000.

[8] Suh, N.P., A Theory of Complexity, Periodicity, and the Design Axioms, Research in Engineering Design, Vol. 11, 1999, pp. 116-131.

[9] Holland, J., Hidden Order: How Adaptation Builds Complexity. Addison Wesley publishing Company, Reading, MA, 1995.

[10] Baldwin, C., Clark, K., Managing in an Age of Modularity. Harvard Business Review, September-October 1997.

[11] Boppe, C., 16.880- Systems Engineering. MIT-report 1998.

[12] Warfield, J.N., A Structure-Based Science of Complexity: Transforming Complexity Into Understanding. Kluver Publishing, Amsterdam, 2000.

[13] Tang, V., Salminen, V., Pillai, B, Extending DSM to Very Complex Systems. Design Structure Matrix World wide-conference, 18-19 September 2000, MIT.

[14] Gharajedaghi, J., System Thinking: Managing Chaos and Complexity. Butterworth- Heineman Publishing, 1999.

[15] Eppinger, S.D. Innovation at the Speed of Information, Harvard Business Review, vol. 79, no. 1, pp. 149-158, January 2001.

[16] Pillai, B., Adaptation of Neural Network and Fuzzy Logic for the Wet-End-Control and Process Management of Paper or Board Machines: A Tool Making Approach. Acta Polytechnica Scandinavica, Mech.Engineering Series No. 138.

[17] Bauer, R.A., Collar, E., Tang, V. The Silverlake Project: Transformation at IBM. Oxford University Press, 1992.

[18] Carlile, P., Lucas, B., Cross-Boundary Work as Cross-Boundary Development. Case Study. ICRMOT Working Paper. MIT Sloan School of Management.

Contact Address

Victor Tang and Vesa Salminen
Massachusetts Institute of Technology, Center for Innovation in Product Development, CIPD
30 Memorial Drive, E60-236,
Cambridge, MA 02139 USA

Tel: +1.617.457.2741
Fax: +1.617.258.0485
Email: victang@mit.edu, vesas@mit.edu

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

THE CASE FOR PROCESS COCKPITS IN MECHANICAL ENGINEERING AND PLANT MANUFACTURING

M Gruenenfelder and A P Hofer

Keywords: complexity management, business process re-engineering, process controlling.

1 Introduction

The more complex a product the more its success depends on its state of the art and the more management attention is absorbed by achieving this state of the art. While hairdryers are not sold for their state-of-the-art technology, products like aircraft and power plants are. They tend to be engineered to order, making the bid preparation (incl. detail engineering and customer adaptation) one of the key factors of market success. Customers for complex products tend to have their own engineering departments which spend a lot of effort specifying exactly what they want to buy. On the producers' side this leads to a huge amount of customer specific engineering for each order. Processes are largely run on a one-off basis for each order, which hampers learning from recurring issues. Both effects – customers' over-specification and internal sub-optimisation – result in huge differences between products of the same family sold to different customers. Potential volume, learning, quality, and cost effects are all squandered. In addition, product development, bid preparation, and order processing are often managed by specialist departments whose interests and goals tend to be uncoordinated. This leads to the loss of strategic direction in the company's core processes, aggravating the problems specified above.

While customer specific adaptation engineering is often necessary to preserve a company's strategic differentiation, its sub-optimal execution may undermine this very differentiation by prolonging time to market, cost and time of bid preparation, quality, time and cost in order processing, and – last but not least – quality in the final product. Therefore, the goal cannot be to abolish customer specific adaptation engineering but rather to align it with the overall strategic goals of the company.

Privatisation and globalisation of markets have led to two basic developments in the customer base of mechanical engineering and plant manufacturing companies. First, new customers entered the market (e.g. independent power producers in the liberalised markets for electricity) aiming primarily at financial return on investment. Second, existing customers started to re-design their purchasing operations (e.g. the US Air Force by adopting commercial procurement practices) in order to save cost and time by ditching non-value-adding over-specification. The customers' drive towards accepting less over-adapted products and the resulting reduction in the customer-specific differences between orders opens the opportunity to optimise the engineer-to-order phase in the gestation of the product.

Internal sub-optimisation can be tackled in two ways. First, isolating customer-specific from customer-independent parts of a product pushes back the order penetration point in order processing. The customer-specific parts have to be structured to mirror customers' requirements, thereby facilitating and speeding up the specification and engineer-to-order phases for such products (see Schuh et al. 2000 for a product architecture perspective). Second, the processes to create the optimally structured products have to be optimised. While there is a plethora of methods and tools to optimise processes, not much has been written on the management of complex processes. This paper proposes a method using so called process cockpits. The analogy with aviation is intended, the process of flying a mission and an aircraft are very complex processes indeed. We illustrate our methodology with a producer of turn-key steam power plants for the core process reference plant development (the other two core processes, bid preparation and order processing were dealt with in the project but are not described in this paper).

Our steam plant producer's strategy was to become world market leader in turn-key steam plants, half average erection time from 36 to 18 months, and implement its reference plants with clients and with its sales organisation. Reference plants are pre-designed plants with the aim to speed up bid preparation (which may last months), enable the organisation to learn from its successes and failures (thereby improving quality) and lower costs significantly (while keeping risks under control). Reference plants can be imagined as consisting of several layers: the core of which is fully designed and standardised (the steam turbine, its auxiliaries, the generator etc.). With each layer the design gets less explicit and more a definition of rules (civil engineering constitutes the outermost layer, with almost nothing pre-engineered and only few issues of reference plant relevance. State of the art civil engineering methods are the only applicable solution). The larger model conforms with the idea of separating the order-specific content from the order-neutral content (moving inside out you move from order-neutral to order-specific content).

2 Making processes manageable

Complex processes can be optimised using well known and documented methodology (for a brief survey of the literature see e.g. Schuh, 1999). In their optimal form, they need constant monitoring and management. The optimisation of a complex process only realises its full potential, if a tool is devised to make the optimised process manageable. Such a tool is the "process cockpit". Its functions can be compared to a real-word cockpit: it must supply the pilot with a good overview of his flight, provide early warning on malfunctions in the aircraft's systems and suggest remedies, and provide the pilot with the necessary support to fly the mission rather than the aircraft. At the centre point of our methodology is the cause-and-effect diagram. Existing sets of index numbers often fail due to a lack of understanding of the underlying process interactions. These systems can be cheated by optimising only single index numbers leading to sub-optimal overall results. As an illustration we may look at order processing for a power plant producer: the "index number of changes in design and procurement engineering" is certainly worthwhile minimising by the responsible department. However, minimising this index number alone is in line with the company's overall goals, if it does not lead to an increase in the number of changes on the building site. This is especially so, since changes on the building site are usually much more expensive and time critical than changes in design and procurement engineering. The cause-and-effect diagram makes these connections explicit, a properly designed process cockpit will take account of these connections, thereby aligning individual goals with those of the overall company.

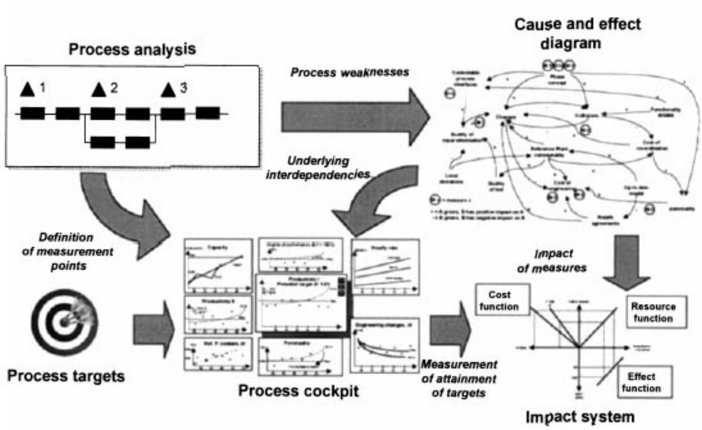


Figure 1: The way to devise a process cockpit

2.1 Cause-and-effect diagrams

The purpose of cause-and-effects diagrams is embedded in their name. They separate causes from effects. While the former are the spots where measures have to apply a lever, the latter are the indicators of success and failure - the bezels and hands in our cockpit instrumentation.

The identification of all factors involved and their clustering into causes and effects is derived from the existing process description and modified in workshops with key people from all processes under scrutiny.

The identification of the interactions between factors poses the question in what direction one factor causes an effect on another. Effects are strictly one way. For instance the communality of a power plant project bid with one of the company's pre-engineered reference plants will cause an effect on the number of engineering changes necessary in the bid phase. The same factor will also have an effect on the deviations experienced in the order processing process (i.e. the erection and commissioning phase). Both effects are negative, i.e. the more communality, the less engineering changes and the less deviations. Of course, this argument does not work the other way around.

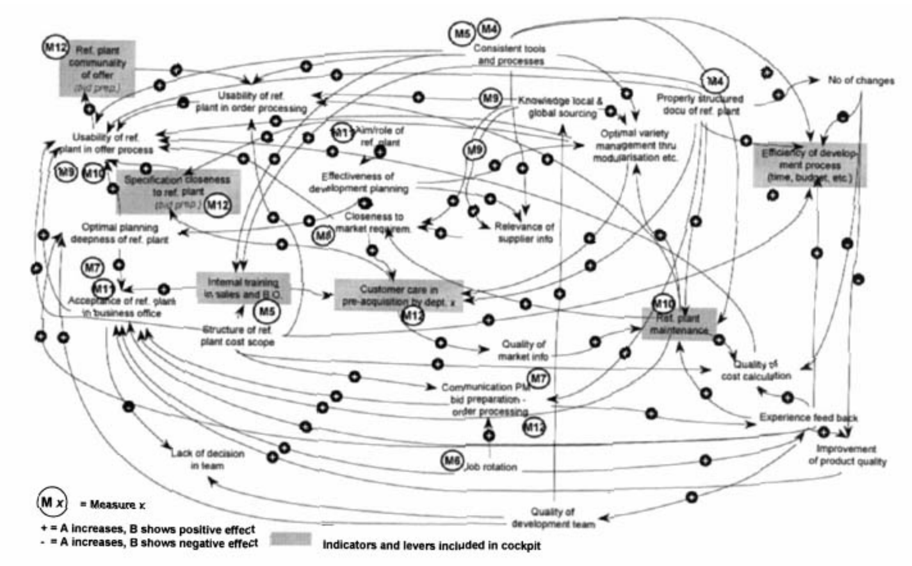


Figure 2: Cause-and-effect diagram

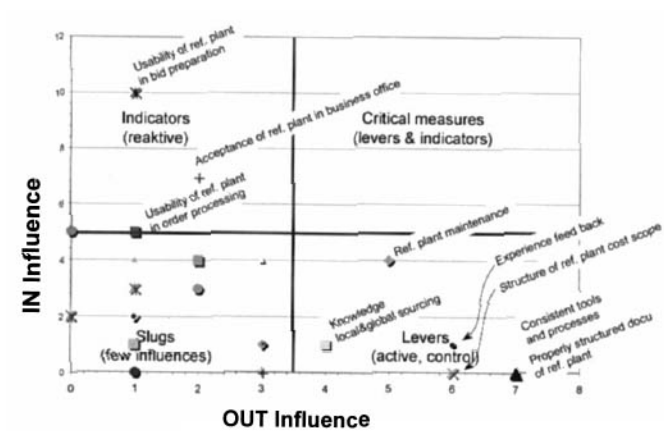


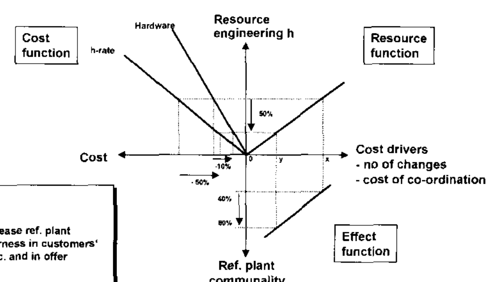
Figure 3: The input output diagram measures the impact of the cause and effects diagram

The aim of a cause-and-effect diagram is to identify and simulate the dynamics of causes and effects. One often finds self-enforcing cycles leading to self-fulfilling prophecies. The team is now able to make use of such cycles in scenarios. The identification of levers is crucial. Improvement measures can be based upon them. Equally, reactive indicators are important because they measure the success of those improvement measures over time. Critical measures are those which are levers and indicators at the same time. They serve as both, index numbers as well as targets for improvement measures.

2.2 Quantification of effects

One of the tasks of the process cockpits will be to show the impact of the process improvement measures defined in the optimisation phase. In order to do that, we need to know, what the technical effect of an improvement measure is, what this means for the cost drivers, and finally how much money can be saved through the improvement measure. This can be accomplished in a technical impact diagram. Usually one impact diagram is needed to gauge the impact for each measure.

Performance indicators are measures in the upper quadrants of figure 3. They have been described in the cause-and-effect diagram. Improvement measures do not alter the general connection between these measures, they rather change the input and the output of the measures. This is called the effect function in figure 4. The so called resource function measures the economic impact of the improvement measures in resource terms (e.g. engineering man hours) and this can easily be converted to money by the so called cost function illustrated in figure 4.



Measure M8/12:

Aim: Increase ref. plant nearness in customers' spec. and in offer

Resource: Make use of ref. plants Pre-acquisition and bid preparation

Procedure: Make pre-acquisition and concept definition part of co-ordination process

		As is	target	delta
Benchmark:	1. Ref. plant nearness of spec.	35%	60%	
	2. Ref. plant nearness of bid	40%	80%	
	3. Engineering h order proc.	*a*	0.5*a*	
	4. Engineering h bid prep.	*b*	0.5*b*	
Basis: ?	Potential:			...

Figure 4: The technical impact diagram captures the monetary impact of each measure

2.3 Design of the cockpits

The cockpits designed to monitor complex processes differ from classic financial controlling and benchmark systems such as the Du-Pont benchmark system. First, each cockpit is designed to meet the requirements of its respective process and its development over time. Second, the content of the cockpit is based on the process targets, the profound understanding of cause-and-effect diagrams and the improvement measures. The cockpits offer the possibility of a simple integration into a company-wide balanced scorecard system, as they fully cover the process monitoring part required by the balanced scorecard.

We discuss the design of cockpits using our example of the steam power plant producer. The core process displayed in the cockpit above is the reference plant development process. For each of the three core processes we designed two cockpits: a project cockpit (e.g. controlling the design of reference plant *x* by project manager *y*) and a process cockpit (e.g. controlling the overall core process of reference plant development for the general manager responsible for this process).

Figure 5 shows the final project cockpit. The three outlined areas concern the project manager's aim (top), process efficiency (lower left), and reference plant effectiveness (lower right).

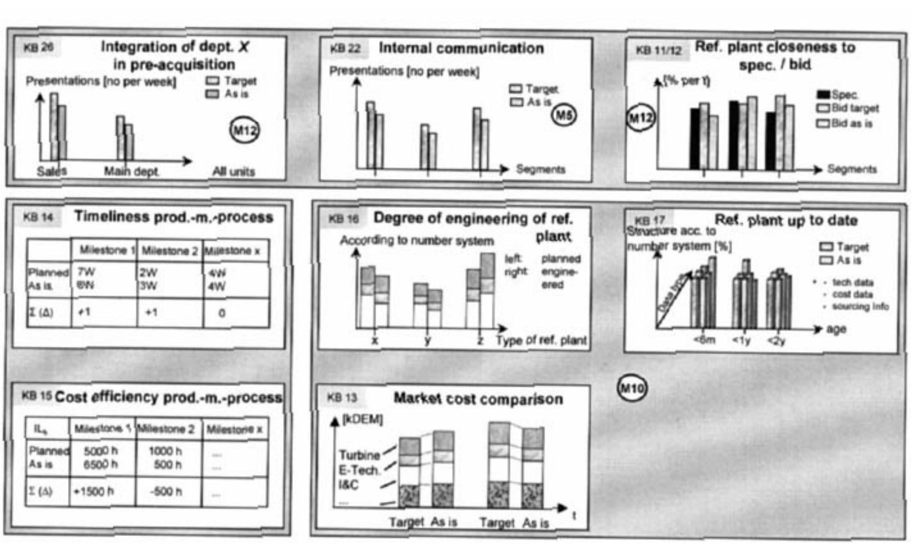


Figure 5: The cockpit

3 Implementation

More often than not it is quite a long and arduous way from devising a coherent concept to actually implement it in its full glory. Three steps will move our coherent concept into the realm of reality.

3.1 Technical implementation of index numbers

This step provides the required index numbers for the process cockpits devised in the earlier steps. Analysis of the data already collected and the IT systems in which they are available gives an outline of the task ahead. Each index number has to be measured for an appropriate period of time which has to be defined for each one. At the same time, the source for each index number has to be defined, including the person responsible for generating and updating the relevant data.

An IT concept has to define how to maximise the use of existing IT tools and where new software or even hardware is needed. This is followed by programming and the actual collection of the relevant data. This cycle is best implemented in a prototype phase allowing for the detail adjustments usually needed.

3.2 Implementation of process improvement measures

This step drives the implementation of the identified process improvement measures. These were originally designed as a remedy to the existing weaknesses in the core processes. Their implementation is crucial for the overall improvement of performance.

Implementation of these measures is usually done in a project organisation, a detailed description of which is beyond the scope of this paper. Index numbers can be devised and included in the cockpits for the most critical improvement measures. Such index numbers are removed after successful implementation of the measures they map. Their inclusion in the cockpit secures full management attention and support – two critical success factors for all such projects.

3.3 Adaptation of management reporting and incentive systems

In order to create a powerful instrument for managing core processes, management reporting and incentive systems have to be adapted accordingly.

Adaptation of management reporting ranges from including the new cockpits in the pages of an existing report to the total re-design of the whole reporting system. Adapting incentive systems means including target values for index numbers into management target systems and linking their attainment to monetary consequences.

4 Conclusion

In our example, the power plant producer restructured his three core processes. In order to ensure effectiveness, appropriate measures have been imposed based on cause-and-effect diagrams. For each improvement measure the team set up an impact diagram to quantify and control its effects. The results of cause-and-effect diagrams and impact diagrams form the basis for the cockpit charts that were integrated into the company's management reporting system.

In this example the so far hidden rules within and the hidden links between the processes became transparent and, therefore, manageable. The power plant producer was enabled to design his processes optimally and to tie them together so that no process would suffer from the one in front or make the one behind it suffer. With the possibility of quantifying each of the improvement measures the foundation was laid for continuous and lasting improvement.

The project started with a pilot project in a selected department. For this pilot department alone, cost reductions of £2m p.a. had been reaped. Based on this result, the decision was taken to roll-out the project to the entire business unit concerned with steam power plants. Due to the long gestation period of such plants and the ongoing character of the project, definitive results for the whole business unit have not yet been determined.

* * *

References

Schuh, G., Gruenenfelder, M., Hofer, A. (2000). 'Managing Complexity In Your Product Range – Reduce Complexity through Strategic Product Definition and Product Family Management.' in McCarthy, I., Rakatobe-Joel, T. *Complexity and Complex Systems in Industry.* Warwick: Warwick University Publishing.

Schuh, G. (1999). *Change Management.* Aachen: Shaker.

Further Reading

Genter, A. (1994). *Entwurf eines Kennzahlensystems zur Effektivitaets- und Effizienzsteigerung von Entwicklungsprojekten.* Muenchen: Oldenburg.

Mueller, M. (1999). *Prozessorientierte Veraenderungsprojekte. Fallbeispiele des Unternehmenswandels.* Ph.D. thesis, University of St. Gallen.

Mueller, A. (1996). *Ansaetze eines ganzheitlichen Controllings.* Munich, Vienna: Oldenburg.

Probst, G., Gomez, P. (1991). *Vernetztes Denken.* Wiesbaden: Gabler.

Corresponding author:

Michael Gruenenfelder, Dufourstrasse 73, CH-9000 St. Gallen, Switzerland. Mobile: +49-172-9827081, email: gruenenfelder.keller@bluewin.ch.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

IMPROVING THE PRODUCT BY ACCELERATING THE PROCESS

J Jarrett and P J Clarkson

Keywords: design management, design strategy, complexity management

1 Introduction

For the half-century following the Second World War, *technology* has been seen as the key to achieving the aim of improving complex products such as turbines. Throughout the 1950s and 1960s great strides were made in the quest for such performance attributes as more power, efficiency and reliability of products through innovative activities like materials research and mechanical development. Into the 1970s and 1980s these have been augmented by the increasing use of computational modelling which has yielded further performance gains. Accordingly, over this period, engineers have often considered the 'goodness' of a design to be primarily a function of its performance.

It is known that the performance improvements due to successive technological advances are now levelling off, and that the law of diminishing returns is increasingly being felt [1]: any further gains to be had by these methods are increasingly marginal, costly and time consuming. This effect combined with the end of the Cold War (resulting in major cuts in defence research and procurement budgets which have historically been big drivers in this area) and worldwide economic recession in the early 1990s has resulted in a paradigm shift: performance-based 'goodness' benefits must now be traded against the *time* and *cost* penalties which they entail. In short, only those performance gains which are *commercially* beneficial are now appropriate.

2 Background

The new paradigm, which relates performance-based goodness to time in order to achieve commercially optimum results, represents a shift away from focussing on the performance-optimised design. It is therefore necessary to establish how the commercial optimum is determined. This may be achieved by considering, in the context of goodness against time, both how the design develops and how the market responds to such a product. Clearly, this response will have a very strong dependence on the market in question.

The Boeing Company presented a qualitative view of its market as a series of contours (figure 1) [2]. A simple, diminishing returns based model of the progression of goodness with design time can be superimposed over these contours. This curve asymptotes to a level determined by the technology employed.

The commercial optimum is thus identified as the point of maximum market share: the point at which the slope of the design curve matches the local slope of the market field. This point is evidently short of the performance maximum and established desired design end point.

Thus if the design process were to continue beyond the commercial optimum, even if the product performed better it would be detrimental to the company's market position with serious repercussions for its profitability.

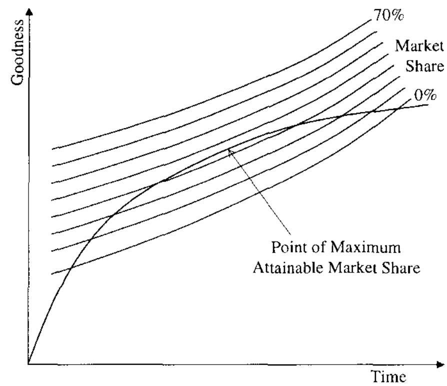


Figure 1. Locating the point of maximum market share [1]

Evidently those companies which move from the performance optimum towards the commercial optimum first will have a significant advantage over their competitors. An example is that of Boeing with the 747 aircraft: produced in three years when the industry norm was five it has owned the large long haul market for a quarter of a century. Clearly once all competitors have established the commercial optimum position the primary way to gain further advantage is to move that company's commercial optimum point. Since the company cannot directly influence the shape of the market, it must turn to altering the shape of its design curve. As new technology is decreasingly effective at achieving this aim, the natural move is to deploy existing technologies more effectively in order to accelerate the process. The effect of successfully achieving this is indicated in figure 2.

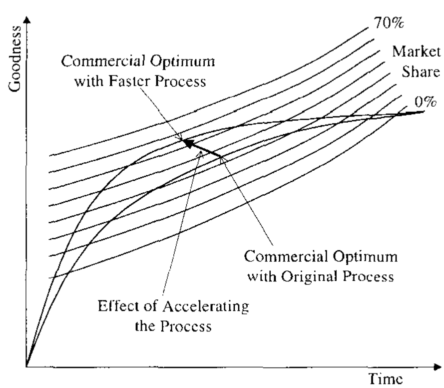


Figure 2. The effect of accelerating the process.

It can be seen from figure 2 that accelerating the process has the effect of not only shortening the design time but also resulting in a "better" product in terms of performance. These two effects combine to yield a significant increase in market share. Thus figure 2 provides the rationale for accelerating the design process; alas, it gives little indication of *how* this may be achieved. Accordingly it is necessary to examine the design process more closely in order to reveal sufficient detail to suggest methods by which acceleration may be affected.

3 The Surge-Stagnate Model

In order to establish a method to accelerate the design process it is necessary to examine the process more closely. A detailed study of a typical turbomachinery product development to resolve its characteristic features and factors controlling its shape has been undertaken.

It has been found that a sufficiently well documented area for examination is the first 50 years of evolution of the steam turbine. Charles Parsons filed his first patents for the steam turbine in April 1884 and constructed his initial prototype machine (No.1) during that and the following year. This machine drove a high speed dynamo and produced 6ehp for a saturated steam consumption of 200lb/kWh. This equates to a thermal efficiency of approximately 1.5%. Yet in just three years he more than trebled the efficiency of his original design. However, the significant detail in the history of this work is that for most of the intervening time, when Parsons experimented with No.1 trying myriad alternative designs to assess their efficiency, there was no real change in performance. Indeed, between 1885 and 1888 he wrestled with the problem of how to reduce tip leakage flows which experiment had shown were significantly reducing the performance of the machine.

He first replaced the straight 45 degree cut blades with conically-holed rings. This produced no gain in efficiency, yet increased the manufacturing cost. For the "Earl Percy" dynamo turbine of 1885, he introduced shrouded blades which made no impression on the leakage flows and accordingly did nothing to improve the efficiency [3]. In 1886 he tried sloping the blade roots to deflect the flow from the leakage paths, but the difference to the standard blading was inappreciable.

An experimental 16kW machine of 1887 had complex and expensive baffle shrouds incorporated. The corresponding reduction in leakage was found to be "somewhat counterbalanced" by increased skin friction, and was thus did not proceed further than the experimental rig. He next tried testing a "vortex flow" variant of the baffle shrouding as well as serrated-against-plane surface techniques in water-based experiments, again to no avail. Thus Parsons struggled for three years, yet crucially made no discernible improvement to his machine. However, what he did gain was a great wealth of experience with regard to his turbine, and a far better understanding of the mechanics of the flows therein.

It had become apparent to him that the shape of the steam jet leaving his blade rows was far from satisfactory, and that his parallel-sided annulus was inappropriate for the steam, which was already increasing by a factor of six through No.1 [4]. Thus the changes he made in 1888 were to bend the blades (which as noted above had previously been straight cut) and to flare the annulus to permit the steam to expand without driving up the axial velocity. The result was that virtually overnight the steam consumption fell from 200lb/kWh to 58lb/kWh. Thus he spent 3 years iterating his original design to little apparent avail, but the *understanding of the flow that he thereby gained* enabled a near four-fold step improvement in efficiency in his next design.

In the longer term, this effect is repeated. If the continued development of non-compounded steam turbines (ie generically the same as Parsons' No.1) is tracked over the subsequent 40 years, two further similar cycles are revealed. These are clearly visible in figure 3.

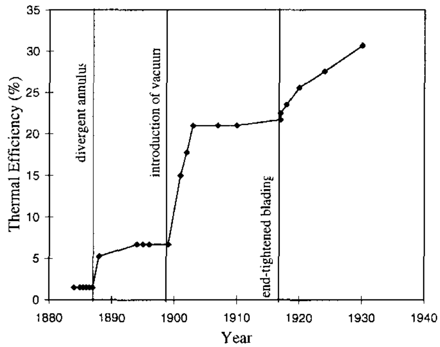


Figure 3. Thermal efficiency of non-compounded steam turbines 1884-1930.

The first of these occurred at the turn of the century due to the introduction of exhaust vacuum at the Neptune Bank Station of the Newcastle-upon-Tyne Electric Supply Company (NESCO), the previous machines having been non-condensing. Parsons had actually started work on such designs as early as 1889, but due to his leaving the Clarke-Chapman Partnership the same year it was not constructed for another 12 years as he lost the rights to his patents.

Optimum vacuum conditions were soon established and little progress was made until an effective method of reducing tip leakage (end-tightened blading) was finally introduced around the start of the Great War [5]. Once again this was a radical departure from previous designs in that the tip leakage flows were reduced by axial clearance control rather than radial (a method which persisted in single-cylinder designs till the 1970s).

Much of the plateaux regions are evidence of years of physical prototyping which may today be done, in a virtual sense using computers, within a single design process. Accordingly, it may be reasoned that the simple diminishing returns controlled design curve can be resolved into a series of such cycles, as shown in figure 4.

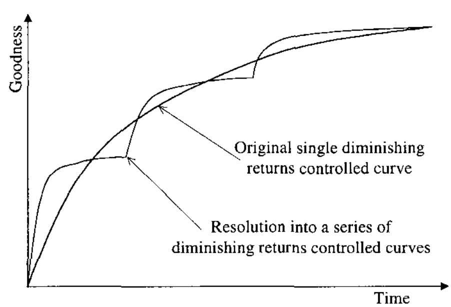


Figure 4. Hypothetical resolution of the design process characteristic.

This new model of the design process is a significant component of this work, and is described as the Surge-Stagnate Model (SSM) [6]. The SSM encapsulates what was learnt from the Parsons case study. It contains stagnations which are reasoned to be caused by iterating the design in order to gain in its understanding. In this time, little direct progress is likely but by doing so, pertinent changes may be made to the design which result in a surge. The model is summarised in figure 5.

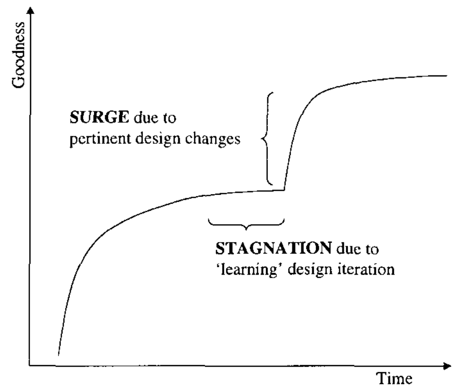


Figure 5. The Surge-Stagnate Model.

4 Quasi-Optimisation

Having developed the Surge-Stagnate Model, thought may now be given as to how the design process may be accelerated. An electrical analogy may be employed. It is evident that each surge-stagnate cycle may be characterised by an amplitude and a time period. It is therefore clear that the process may be accelerated either by amplification of the surges or by shortening the stagnations, as shown in figure 6.

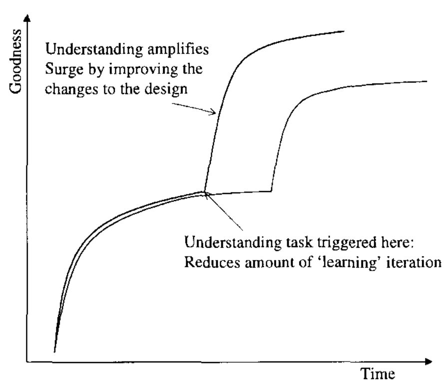


Figure 6. The desired effects of Quasi-Optimisation.

From the evidence of the Parsons Study, it is clear that both of these aims may be obtained by facilitating the generation of *understanding* by the use of an 'understanding task' which provides sensitivity information in the form of local partial differentials between performance parameters and user-variables. By gaining more understanding earlier in the stagnation, the designer may make a more pertinent change (amplifying the surge) earlier (shortening the stagnation). This method we describe as Quasi-Optimisation (QO), to convey the fact that it is required to be capable of operating on an incomplete design. The application of QO is particularly suited to design processes that are computationally intensive, thus enabling the generation of understanding using a sensitivity analysis. However, it should be noted that such approaches are also possible in the absence of a computational model [7].

5 Results

In order to test these concepts, a working model of the SSM has been implemented for the design of steam turbines based on the Signposting methodology developed at the Cambridge Engineering Design Centre [8]. Individual tasks in the design process have been identified through literature survey and consultation with industry. These tasks have then been placed in categories such as Estimate, Calculate and Select. The tool has been implemented in Fortran, primarily due to its use in several of the industry standard Computational Fluid Dynamic (CFD) packages, thus minimising interfacing problems. The tool consists of a library of tasks, and a dynamic signposting engine which supports the designer by indicating (via the characteristic Signposting traffic light symbology) which tasks are suitable at any particular point in time, dependent on the current state of the emerging design.

In this way, the designer generates information about the design space en route. Design goodness is evaluated by comparison of working values of performance parameters with their specified values and weighted by the accuracy with which they are known. Once a working value for at least one performance parameter has been produced, QO may be deployed. The designer may (but is not obliged if he has sufficient understanding or experience) request the software to interrogate the design space about the current design definition, generating sensitivities of performance parameters to changes in input variables. The resultant analysis prompts the designer to progress the design in the most profitable direction. Thus the designer may save time by not having to learn this information by a lengthy process of iteration and instead may make appropriate changes to the design at the first attempt.

A laboratory experiment, involving seven designers with no prior knowledge of steam turbine design, was undertaken following the inclusion of a switchable Understanding Mode. Each designer was required to design a turbine to a commercial specification with and without assistance. In this way it was possible to assess the effect of QO on the progress of the design and a typical result is illustrated in figure 7. It is immediately apparent that Surge-Stagnation cycles are very clear in both curves. In addition, the result with the Understanding Mode enabled is consistently faster than its unassisted equivalent to achieve some goodness levels. For example, 70% goodness is achieved in 450 seconds with the Understanding Mode and in 1350 seconds without. This improvement was not found to be dependent upon the order the designer undertook the experiments.

More insight into how QO is achieving this acceleration affect may be gained by averaging the data from twenty hours of testing as shown in figure 8. It is clear that both the anticipated acceleration mechanisms are occurring. QO increases the effectiveness (in terms of goodness creation) of an industry standard throughflow design code 'SLEQ' by a third, evident by amplification of its associated surge from a 15 to a 20 percentage point increase. It is concluded that this is a downstream response to the use of QO at the preliminary design level.

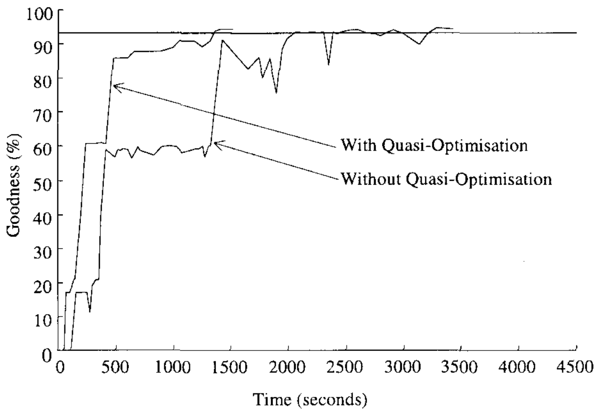


Figure 7. A typical comparative laboratory test.

It is also clear from figure 8 that the stagnation downstream of the invocation of SLEQ is halved compared with the unassisted equivalent. This is a response to QO indicating to the designer via sensitivity data not only how much to change a parameter and in which direction, but also which parameter it is most effective to concentrate on.

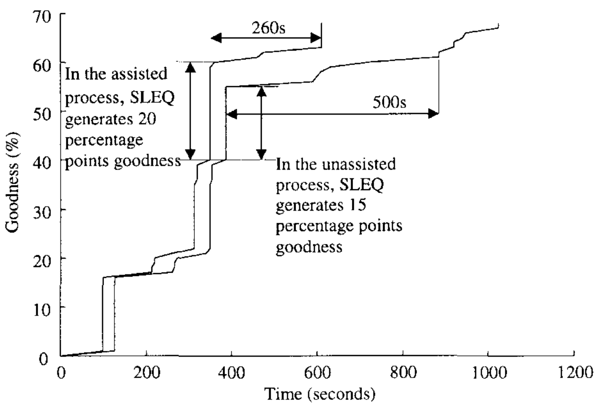


Figure 8. Amplification effect of QO on SLEQ-induced surge.

By comparison with an industrial design, to the same specification using exactly the same technology, the average steam turbine produced by completed Quasi-Optimised processes in the testing programme were 1.3 percentage points more isentropically efficient. Crucially, this performance increase was accompanied by a design time reduction *of an order of magnitude*. This is a clear indication of strong acceleration.

Clearly the steam turbine case design study, conducted under laboratory conditions, cannot validate the existence of SSM and the use of QO. However, it does indicate that a design process characterised by SSM can be accelerated using QO. Further work is now under way to apply these ideas within an industrial context.

6 Conclusions

It is evident that market share and therefore industrial profitability may be increased by accelerating the design process. This paper has presented the Surge-Stagnate Model which provides a framework in which to examine the factors controlling process speed and Quasi-Optimisation as a method to achieve the required process acceleration.

The presented results of a laboratory test of an implemented system confirm that the SSM is a valid model of the Steam Turbine design process and that QO produces an increase of 1.3 percentage points isentropic efficiency whilst reducing the design time by an order of magnitude compared with an industrial case study.

The methods described in this paper and tested under laboratory conditions are now being implemented in an industrial Engineering Design environment in a project funded by Rolls-Royce plc.

References

[1] Dryell, D R and Freeman, C W (1989), "Trends in Design in Turbines for Aero Engines," 2nd. Parsons International Turbine Conference. Published in "Materials Development in Turbo-Machinery Design," pp38-45, Dublin.

[2] Rubbert, P E (1994), "CFD and the Changing World of Airplane Design," AIAA Wright Brothers Lecture, Sept.

[3] Richardson, A (1911), "The Evolution of the Parsons Steam Turbine, Engineering," London.

[4] Bolter, J R (1994), "Sir Charles Parsons and Electrical Power Generation – a Turbine Designer's Perspective," Parsons Memorial Lecture, Proceedings of the Institution of Mechanical Engineers, **208**: pp159-176.

[5] Hills, R L (1989), "Power from Steam: A History of the Stationary Steam Engine," Cambridge University Press.

[6] Jarrett, J P and Clarkson, P J (2000), "The Surge-Stagnate Model for Complex Design," Engineering Design Conference, London, pp771-778.

[7] Ullman, D G, Herling, D and D'Ambrosio, B (1997), "What To Do Next: Using the Problem Status to Determine the Course of Action," Research in Engineering Design, **9**(4): pp214-227.

[8] Clarkson, P J and Hamilton, J R (2000), "'Signposting', A Parameter-driven Task-Based Model of the Design Process," Research in Engineering Design, **12**(1): pp18-38.

Jerome P Jarrett
Engineering Design Centre
University of Cambridge
Trumpington Street
Cambridge CB2 1PZ
United Kingdom
Phone: +44 (0)1223 332 673
Fax: +44 (0)1223 332 662
E-mail: jpj1001@eng.cam.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

COMPLEXITY IN PLANNING DESIGN PROCESSES

C Earl, J Johnson, and C M Eckert

Keywords: Systematic product development, process modelling, complexity management

1 Introduction

Planning design processes requires balancing a company's resources across different product developments and exploiting similarities between products. Each product development has a network of processes. Similarities between processes on different products give connectivities between products. There are several dimensions for these connectivities ranging from common components to shared design knowledge. Multidimensional networks provide a structural description of complexity for design processes. Design planning is described by flows or 'traffic' on this network.

2 How are separate product developments connected?

Optimizing logistics, manufacturing, design and human resources simultaneously poses a challenge for many companies; and requires careful planning. There are conflicting objectives: on the one hand manufacturing and human resources should be cost efficient, on the other hand the company innovates, invests and rejuvenates to keep a competitive advantage. To help achieve a balance companies maximise the use of existing manufacturing processes and equipment, parts of existing designs, standard design procedures and the knowledge of their employees. Resources are shared across different products that are under development at any one time. This results in a complex network of shared activities, knowledge, parts and processes.

Design companies deal with the complexity of individual products as well as the complexity of designing and manufacturing many different products. An aircraft with 10 000's of parts and 100 000's person hours design time is an extreme example. Few designs are fully modular, so that a change to one part can lead to changes in many other parts of the system [1,2]. Suh [3] reviewed complexity in terms of uncertainties. Design processes use resources such as technical experts, in house company knowledge and computer models to reduce uncertainties. Critically, results from previous similar designs will reduce uncertainty and thus design complexity. However, these are not always 'freely' available as the complex connections among products need to be established and maintained. This structural complexity of processes and relations is represented here by a multidimensional network [4].

Connections between products arise through similarities between the processes used in their development. Similarities occur in many ways ranging from design principles to component details. By definition a new product is not identical to anything that exists. The planner assesses what is the same as in other products, what is similar and what requires a new solution. An analysis of the different types of similarity [5] helps a planner to estimate resources required and to identify potential conflicts or trade-offs.

3 Modelling design processes and their connections

McGovern et al [6] describe the complexity of product design, manufacture and supply for customized or one-of-a-kind products. Complex networks of processes exist both within companies and throughout their supply chains [7]. Each product development is highly constrained by available resources and expertise. A critical factor is making effective use of previous and concurrent product developments [6]. Similar observations were made in a study about customization in the aerospace industry [1] where a lack of understanding about the connectivities between different parts of the aircraft lead to inefficient processes. This is because changes were not planned with respect to their impact on other design problems.

3.1 Structure of design processes

Staged design process models formulated in terms of idea generation, conceptual design, embodiment and detailed design (for example [8]) provide guidance for developing milestones, but do not address specific design activities or product properties. For planning purposes these models are too high-level. Design processes at the more detailed level are modelled by activity or task networks and associated algebraic representations of relationships and connectivities between activities. Eppinger et al [9] apply design structure matrices (DSM) to create a static task model for complex processes. Clarkson and Hamilton describe signposting [10] which is a dynamic task model. In this signposting model, tasks are connected through the assignment of parameters. A confidence measure on parameter values describes the current state of tasks and is used to 'signpost' those tasks that can be undertaken at any one time. Theoretically, parameter confidence represents a significant feature of the process model because it provides a mechanism for iteration. Both signposting and DSM generate a network of tasks. In this paper we will use the term process to denote the design activities or tasks in product development. A process may be a well specified single task such as mould design for a component or a diverse design task on a major subassembly such as a power train or a control system.

Design processes can be decomposed into hierarchies of tasks or subprocesses in much the same way that component and subassembly hierarchies describe products. Hierarchies are not unique, the same elements can be grouped in many different ways. This causes problems in many engineering companies, because different people with different concerns prefer different groupings. For example assembly planning will group by the order in which parts will be put together. Manufacturing will group by similar processes or suppliers. Designers will group by components. Process planners will group by teams and differentiate between suppliers and in-house design. It also leads to the (artificial) 'beancounting problem' when classes intersect, and planners (unnecessarily) insist that numbers must be assigned to on class or the other, but not both [11]. it is common for planners to react to this by imposing artificial disconnections, which may obscure opportunities and dangers.

Basic product information is typically derived from a bill of materials, which represents the units of purchasing. These units can vary in size and importance, e.g. a whole engine and a simple component such as a screw may both be purchased items. This distorts the hierarchical representation as both are represented as leaf nodes in the tree-like product structure of the bill of materials. Developing product hierarchies, including heterarchies in which things can belong to more than one class, and applying them across different parts of a business is the ideal of knowledge management. Relations across products and other business activities are defined by similarities. The next section identifies several different types of similarity relevant to modelling connections among processes.

3.2 Types of similarity

Products and processes share and reuse what is similar. Earl et al [5] examine similarity according to levels or layers (table 1) in a particular process. Similarity can propagate downwards through the layers from knowledge to components. If similar solution principles are applied on the product level, e.g. a similar power train system is used in more than one vehicle, then product development is likely to specify a similar manufacturing process and share some components. On the other hand a very similar component on the product level might have a totally different manufacturing process while sharing many solution principles, for example changing from a cast product to a pressing process. Equally, similarity can propagate upwards provided that the two products have similar requirements. For example if two car headlamps are produced by a similar manufacturing process, they are likely to be similar products.

Layer	Similarity
Components	Identify
Manufacturing	Similar product → similar process
Product	Identical systems, solution principles
Process	Procedures, product design
Knowledge	Expertise

Table 1 Layers and Similarity

Therefore, *a priori* independence of the layers is assumed and each layer is assessed individually. However, an efficient plan will reflect coherence across multiple layers. Overlap in reuse between them is likely to be the most parsimonious approach, e.g. using a similar manufacturing process for a similar product part is likely to be cheaper then applying a new or different manufacturing process. A cost effective way to design is to limit change and innovation to well contained areas, while keeping others as constant as possible.

3.3 Structural complexity of processes

Design and product development display several features of complex systems. For example deterministic chaos is apparent when what seem like insignificant changes to a specification lead to considerable changes to final design and associated costs to bring it to market. The product's behaviour may be difficult to predict because there may be little information about possible solutions or a high degree of dependence among design features. This corresponds to the complexity described by Suh [3]. A fortuitous change might simplify a process, but an innocent isolated alteration, to please a client, may bring added costs.

Many descriptions of complexity in design deal with single product developments. Two main themes emerge. The first concentrates on variety and structure of processes [12] whilst the second examines information and the functional structure of the product [3]. From the perspective of axiomatic design, Suh [3] presented uncertainty as a key definer of complexity. Time-independent and time-dependent complexity were distinguished. Our model applies this distinction, considering a static (time-independent) structure of processes with a dynamic (time-dependent) traffic of product developments.

Predictable behaviour of design processes depends on available resources. With several products under development, processes will be shared and experience from previous designs analysed for the new products. With limited resources queues will build up. This case is familiar in manufacturing, where analytic information theory measures of complexity such as Frizelle and Woodcock [13], Deshmukh et al [14] and Calinescu et al [15] are based on operational uncertainties. Complexities in designing and its associated decision processes are described by similar entropy maximising measures by Tribus [16], March [17] and Suh [18]. These indicate how tasks are spread across available design processes or resources. This complexity of behaviour reflects a complexity of the underlying structure of processes.

Structural complexity is described by Johnson [4] who concentrates on hierarchies of elements and relations. System behaviour is represented by 'traffic' on a multidimensional network of hierarchical objects. These objects might be processes in a product development. Mathematically traffic is represented by numbers attached to structures . Changes are transmitted through the network via shared processes which are used on different product developments. This might entail a process used simultaneously by two products or the reuse of information from a similar process on a previous product. The connectivities in the network enable experience, knowledge and capability are transmitted between products.

4 Planning multiple product developments

Design process models such as signposting or design structure matrices indicate the complexity of relations among tasks [10,19]. These representations are not simple networks based on precedence (input-output) but need to express the dynamic multidimensional relations between tasks. For instance design parameters determined by one task will affect the determination of parameters in another task. Activities and their relations in a single product development form a 'product network' [6]. We concentrate here on the complexity of multiple product developments which use common processes, simultaneously, or sequentially. In part this is prompted by the observation in the information-theoretic analysis of manufacturing complexity (eg [15]) that it is the mix of product flows on manufacturing facilities which presents complex behaviour.

4.1 Establishing connectivities among products

If an activity is shared or substantially reused then the products are connected by this similar activity. Product networks may be connected by several similar activities representing the 'dimension' of the connection. Dimensions of connection can be defined according to the numbers and strengths of similarities. Different connectivity patterns emerge at different dimensions.

The connectivities yield a network of activities across the product networks. They represent areas in which common information is used on separate activities or where knowledge acquired in one activity is used in another. Each activity generates its own design knowledge. Although knowledge management is not the subject here, the model suggests a 'personalization' rather than a 'codification' strategy [20]. This latter seems appropriate for standardised, low cost solutions with the former representing a more flexible approach allowing careful interpretation of previous experience. It appears problematic to mix the two strategies, at least for knowledge management in consulting. This distinction applies rather neatly to product developments. Direct connections to other processes indicates personalization whilst codification refers to abstractions and product databases.

Connectivities between processes on different products generates a network which links products according to similar processes. Roughly this can be pictured as a 'knowledge' network across the products (figure 1). Although knowledge is only one of the layers of process at which connections are made, the term 'knowledge' network seems appropriate to reflect communication between product developments.

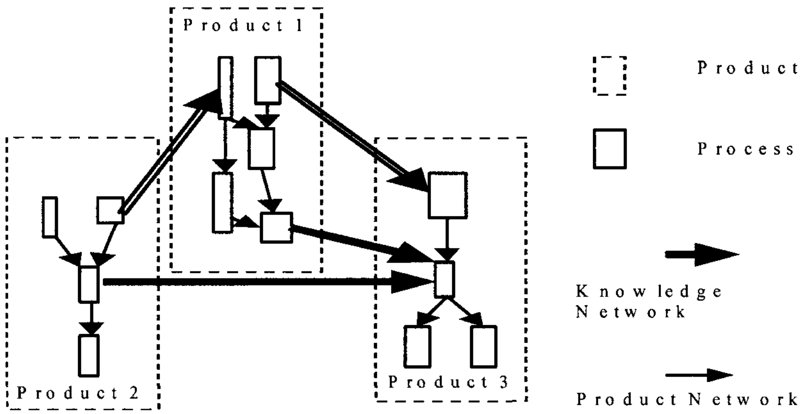


Figure 1 Product and knowledge networks

However, this visualisation misses the dimensions of similarity. These require a multidimensional network which can also represent higher dimensional connectivities. For example a large set of activities, similar in many products, might represent central design issues. Peripheral, weak connections can also be identified. The argument here is that these hierarchical types can be discriminated and not that they should be ignored. There are parallels to the application of design structure matrices and general classification methods which are used to inform decompositions of process rather than prescribe them.

4.2 Modelling connectivities by multidimensional networks

Suppose a product development uses identifiable activities $\{a_1, a_2 \ldots\ldots a_n\} = \mathbf{a}$ in a product network $N(\mathbf{a})$. Other products have activities $\{b_1, b_2 \ldots\ldots b_m\} = \mathbf{b}$ and $\{c_1, c_2 \ldots\ldots c_l\} = \mathbf{c}$. Similarity is defined by relations between activities. For example similar knowledge in a_i and b_j (or possibly across many activities in both **a** and **b**) is represented by $K(a_i, b_j)$. The similar knowledge relation may extend to $K(b_j, c_k)$. Because this is a similarity or tolerance relation [21] it is not transitive so $K(a_i, c_k)$ may not hold.

The elementary relations such as K induce composite higher level similarities. For example, if P is a relation indicating a shared component part, then composite similarity is defined as requiring both K and P. More generally, a composite similarity of strength 3 requires just three similarities, irrespective of type. A group or polyhedron [4] of activities is defined, through similarity as well as the hierarchical groupings of activities in product networks, provides the representation of multidimensional connectivity. Connections are then made between disparate activities because they belong to groups which are connected though common members in a chain of connections. These multidimensional network models have been applied in several design areas, ranging from vision systems to road traffic planning.

In order to model flows in this polyhedral structure Johnson [4] assigns functions to elementary activities or groups (polyhedra) whose values change over time. Values on an activity depend on values at connected activities. Changes in the values of functions give flows between activities (and between polyhedra at higher levels in the hierarchy). The underlying multidimensional network indicates possibilities to be exploited in a plan of design activity, the changing values of the functions, or 'traffic' in Johnson's terminology represents system behaviour.

4.3 Complexity – structure plus traffic

Planning the design of a product, means identifying required processes and then sequencing and scheduling them. Similarity analysis provides connections to other products some of which are still being planned, some under development and some already completed.

A plan is a feasible flow on the multidimensional knowledge network. In information theoretic models of complexity, product uncertainties in meeting requirements are a measure of complexity of a design. The analysis of these uncertainties attempts to identify the sources of knowledge and capabilities available. For example if a very similar product has been made successfully then there should be little uncertainty; provided that design data is available. The structure of the knowledge network should reveal the uncertainty and thus complexities of the design. However, this 'in principle' needs converting to an 'in practice'. The complexity of the design will depend on how resources are used and knowledge accessed. An ideal plan will reflect the information theoretic complexity. However, a plan which takes account of contentions on limited resources, possibly offset by collaborations on shared resources, requires a realistic assessment of design complexity. This is provided by the structural model of the multidimensional knowledge network.

The information theoretic measures of complexity do two things. First they set the ideal design complexity and secondly they can assess the overall complexity of a design 'system' engaged in many products. The latter gives an overall description of the system behaviour and indicates where the interventions of plans can reduce complexity. The former indicates what a plan attempts to achieve.

5 Examples

In order to illustrate the connectivities among products, two examples of design processes from different sectors are outlined. A recent study of the design of a sports car showed that similar solutions were shared with other designs but with insufficient planning to capitalise on these similarities. The new car is designed to a higher specification than the successful lead product of the company. The new design intended to reuse the maximum number of components. However, now, half way through development designers only expect to be able to reuse about 40 % of minor components. The new product will also include components from cars of other manufacturers, such as highlights because these are expensive to develop. The manufacturing process will be the same to that of the existing car, which had included design innovations to increase manufacturing efficiency. The new design used the engine and powertrain of another European car manufacturer but developed it further. It will be used by both companies. A additional dimension is that the whole industry is aiming to standardize processes across all their products based on the common quality procedures developed between US car manufacturers and their suppliers. Knowledge is shared across products in-house by people moving from on project to another. Many new people were taken on in a recent expansion further increasing the complexity of the knowledge network.

A second example is from a company designing and manufacturing equipment for power generation plant on an engineer-to-order basis. The company recently outsourced most manufacture, concentrating activity on design in a global market. Planning design has become more important. In ETO, key uncertainties are time and resources. Uncertainties (quantified by variances) can accumulate in alarming ways resulting in high probabilities of late delivery. A number of strategies are being explored to reduce uncertainties, including standardizing products. Although this runs counter to other market expectations, especially for a design led company, of more customized products with increased variety. The multidimensional network can simulate the mix of products and changes to knowledge required. Initial mapping of product developments and their connectivities appears to reduce uncertainties. In the longer term this can yield not only operational benefits but also reassessment of product strategy through analysing complexity in the mix of products and associated design processes.. The latter is key goal for this company as it assumes its new design-led strategy.

6 Conclusions

A model of the complexity in design processes arising from the interactions among multiple product developments is presented. The 'one-off' characteristics and uncertainties in product development make it a complex process in its own right. However, the mix and interaction of different products presents a problem of planning resources to undertake these complex processes. This interaction cannot be avoided through decomposition and dependency reduction since it provides the mechanism through which knowledge and experience are used.

Connectivities between processes enable the flow of knowledge, experience and product data. The complex structure of multiple product developments is described by a multidimensional network This hierarchical network is based on different types and strengths of connectivities between design activities. Information based measures provide an ideal design complexity. Design planning under conditions of limited and costly resources can benefit from modelling behaviour as flows or 'traffic' on a multidimensional network of connectivities among product developments

References

[1] Eckert, C M, Zanker, W, Clarkson, P J 2001, "A better understanding of change in the development process", to be presented ICED 2001, Glasgow UK

[2] Lindemann, U Reichwald R 1998 *Integriertes Änderungsmanagement*, Springer, Berlin.

[3] Suh N P 1999 "A Theory of Complexity, Periodicity and the Design Axioms" *Research in Engineering Design* **11** 116-133

[4] Johnson 1995 "The Multidimensional Networks of Complex Systems" *in Networks in Action*, J Casti, R Thord, D Batten (eds), Berlin, Springer 49-79

[5] Earl C, Eckert C, Johnson J 2001 "Complexity of planning design processes" ASME Design Theory and Methodology Conference, Paper, DTM-21697, Pittsburgh 9-12 September

[6] McGovern T, Earl C F, and Hicks C 2000 "Product and Knowledge Networks in Manufacture and Supply of Engineer-to-Order Products" in *Proceedings 9th International IPSERA Conference,* London, Ontario, CA, June 2000

[7] Hicks C, McGovern T, Earl C F 2000 "Supply Chain Management: a strategic issue in engineer to order manufacturing" *Int Journal of Production Economics* **65** 179-190

[8] Cross N 1989 *Engineering Design Methods* John Wiley, Chichester, UK

[9] Eppinger S D, D E Whitney, R P Smith, and D A Gebala 1994 "A Model-Based Method for Organizing Tasks in Product Development" *Research in Engineering Design* **6** 1-13

[10] Clarkson P J and Hamilton J R 2000 "Signposting: a parameter-driven task-based model of the design process" *Research in Engineering Design* **12**(1) 18-38.

[11] Johnson J 2001 "Coherence Theory in the Science of Complexity" Proceedigns *Proceedings Complexity and Complex Systems in Industry* ed McCarthy I P and Rakotobe-Joel T, Warwick University, Warwick,373-386

[12] Ehrlenspiel 1995 *Integrierte Produktenwicklung* Carl Hauser Verlag, Munich

[13] Frizelle G, Woodcock E 1995 "Measuring complexity as an aid to developing operational strategy" *Int J Operations and Production Management* **15**(5) 26-39

[14] Deshmukh A V Talavage JJ, Barash MM "Complexity in Manufacturing Systems" *IIE Transactions* **30** 645-655

[15] Calinescu A, Efstathiou J, Sivadasan S, Schirn J Huaccho Huatuco L 2000 "Complexity in Manufacturing: an Information Theoretic Approach" *Proceedings Complexity and Complex Systems in Industry* ed McCarthy I P and Rakotobe-Joel T, Warwick University, Warwick, 30-44

[16] Tribus M 1969 *Rational Descriptions, Decisons and Designs* Pergamon, New York

[17] March L 1976 "The Logic of Design and the Question of Value" in *Architecture of Form* CUP, Cambridge 1-40 (reprinted as Logic of Design in N Cross (ed) *Developments in Design Methodology*, Wiley, Chichester 1984)

[18] Suh N P 1990 *Principles of Design* Oxford University Press, New York

[19] Krishnan V, Eppinger S, Whitney D 1997 "A model based framework to overlap product development activities" *Management Science* **43**(4) 437-451

[20] Hansen M T, Nohria N, Tierney T "What's your strategy for managing knowledge?" *Harvard Business Review* March –April 1999

[21] Schreider J A 1975 *Equality, Resemblance and Order* Mir, Moscow

C F Earl
Department of Design and Innovation, Open University, Milton Keynes, MK7 6AA, UK
Tel (44) 1908 652398, fax: (44)1908 654052, Email: c.f.earl@open.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A COMPLEXITY INDEX FOR THE DESIGN PROCESS

S Sinha, A I Thomson, and B Kumar

Keywords: Competitiveness, complexity management, design management, human resources, product planning

Abstract

Design projects are typically plagued with schedule and cost overruns. A factor in these overruns can be attributed to projects being more complex than originally anticipated. Although research has been carried out in this area there is currently no method for measuring the complexity of the design process. In order to estimate costs and schedule projects more accurately it is essential that complexity of the design process can be measured. Presently, design process complexity is highly subjective. The objective of this research is to quantify the complexity of the design process through the establishment of a 'Complexity Index' for the design process, thus removing the subjective aspect.

1.0 Introduction

1.1 Motivation

The complexity of design is the cause of many management problems in industrial companies [1]. A transition from the qualitative understanding of complexity to a quantitative understanding would be a highly desirable and necessary step towards the understanding of overruns of design projects [2]. Design projects are typically plagued with schedule and cost overruns ranging between 41% - 258% and 97% - 151% respectively [3,4]. A factor in these overruns can be attributed to projects being more complex than originally anticipated. Although research has been carried out in this area there is currently no method for measuring the complexity of the design process. In order to schedule projects and estimate costs more accurately it is essential that complexity of the design process can be measured. Measurement is vital for controlling any process because it is difficult to manage what cannot be measured [5]. Presently, design process complexity is highly subjective. The objective of this research is to quantify the complexity of the design process through the establishment of a "complexity index" for the design process, thus removing the subjective aspect. This research has been inspired by work carried out by Salingaros in assigning complexity to architectural buildings by quantifying qualitative issues using scientific principles from other domains of science [6]. As complexity of a design process has qualitative factors it was felt that some methodology may be applied to quantify its complexity within a certain context. Wiendahl calls for increased efforts to make complexity quantifiable [7]. Only when this measurability is achieved can fresh approaches be developed for reducing complexity in production systems and so produce a systematic reduction in complexity.

1.2 Background

The Oxford English Dictionary defines complexity as "the state or quality of being intricate, complicated or complex". Complexity and complex terms are normally used in everyday situations to describe a characteristic, which it is not yet possible to quantify precisely. Scientists and Mathematicians consider a system to be 'complex' when it consists of a multitude of interacting elements. One problem with the definition of complexity is that there is no single concept of complexity that can adequately capture our intuitive notion of what the word ought to mean. Several different kinds of complexity may have to be defined [8].

Hayes, refers to design complexity as being 'a function of the number of components in a given product [9] and the number of different components and parts required to build the product line as a whole'. However Kim points out that there could be a system with fewer parts, which could be extremely complex and more difficult to manage than a system having hundreds, perhaps thousands of parts [10]. Complexity has also been defined in the context of product development and design management [11] as one in which 'a great many independent agents are interacting with each other in a great many ways'. A product development process will see interdependencies between components as well as departments and people.

Measuring complexity in design has long been a concern for managers and researchers [1]. The earliest attempts to mitigate the effects of complexity involved trying to simplify the process. This has revolved around "taylorist" approaches to scientific management; reducing tasks to minimum standardised elements to allow a clear view of planning requirements [1]. However, the taylorist approach shows limitations when adopted on a large scale. According to mathematician Von Neumann complexity could be numerically measured like any other system by relating it to a quantifiable parameter, such as the length of a programme or the magnitude of a 'cost' in money or time [12].

1.3 The design process and its complexity

The engineering design process [13,14 &15] is the means by which ideas and requirements are converted into the information from which products and technical systems can be made. It is fundamental to all industry, and must respond to changing expectations and the effect of many indirect influences [16]. It has also been established [7] that a linear relationship does not always exist between the product and process. There are some firms, which manufacture 'simple' products using complex product installations such as manufacturers of injection moulded plastic parts. Products such as castings, panels and moulded parts are usually not complex in themselves, but the injection moulding facilities used to produce them are in fact extremely 'complicated' system as regards technical features like productivity, convertibility or energy consumption. On the other hand, there are other companies, which manufacture complex products using simple production facilities.

Singh, comments on the need of measuring complexity within a context rather than measuring it on an absolute scale [17]. Although is it possible to rate one product or technology as more complex than another closely related product or technology, it is difficult to establish context independant measures of complexity. Others point out that one can expect the degree of complexity to vary with the skills and the insight of the task doer [18]. Similarly, Frost emphasises the influence of skills and knowledge in dealing with complex tasks [19]. All of these issues fall within the category of design context and are being addressed within the scope of this project.

2.0 Aims and objectives

The overall aim of this project is to numerically quantify the complexity of the design process within specific design contexts. This will be addressed through the following objectives:

- identifying the designers perspective about complexity, its manipulation and relationship with other design parameters;
- creating a model of a design process that enables the manipulation of knowledge about complexity.
- defining the relationship that exists between product and process complexity;
- establishing the parameters, which contribute to the complexity of the design process and the relationship that exists between them within specific design contexts;

3.0 Research methodology

This research is being carried out by addressing a number of stages:

- **Literature review:** review of current literature in complexity, design, design complexity and related areas;

- **Scenario investigations and identification of CGF's:** in order to model the complexity of the design process a number of "design process scenarios" are being investigated. This involves in-depth analysis of the design process to identify the factors which contribute to complexity;

- **Contextual analysis and modelling:** this stage involves the identification of the generic constituent parts of 'design context'. The relationship that exists between the Complexity Generating Factors will then be determined for specific instances of design context;

- **Implementation:** Computer modelling of design process complexity is being facilitated through the adoption of a Product Analysis module of Tech Optimiser 3 [20]

- **Testing and Evaluation**: once a model for design process complexity has been created it will be evaluated using case studies.

4.0 Findings

Work to date has focused on three distinct areas:

- Literature study;
- Identifying Complexity Generating Factors (CGF's);
- Contextual analysis and modelling.

4.1 Literature Study

A review of relevant literature has identified the following key observations:

- Although research has been carried out in the area of identifying the parameters which drive complexity, no complete set of complexity drivers have been identified;
- Little is known about the relationship which exists between the parameters which drive complexity;
- Very little has been established in developing a relationship between the complexity of the design process and the complexity of the final product;
- Design complexity is strongly context dependant.

4.2 Identification of CGF's

It is proposed that complexity of a design process can be defined by a number of parameters. These parameters being generators of complexity known as 'Complexity Generating Factors' (CGF's). This concept is supported by one of the earliest investigators of complexity, mathematician Von Neumann who suggested that complexity could be numerically measured like any other identity with reference to parameters [12]. A number of key complexity generating factors have been identified as part of the work carried out so far, including:

- inherent product complexity;
- process complexity;
- team co-operation and communication;
- computer and network complexity;
- a maze of specifications including international regulations and safety;
- creative design;
- customers requirements;
- size of the design;
- structure of the team;
- design tools;
- design constraints;
- dependencies within the design activities;
- geographical locations.

Some commonality exists between the CGF's for example customer requirements are related to design constraints.

Once all CGF's have been identified it is necessary to determine the relationship that exists between them. This relationship however, is strongly dependant upon the design context. In one context, a CGF may have a major impact on the overall complexity whilst in another it may become insignificant. For instance team co-operation and communication may play a predominant role in the complexity of a globally distributed design context but less significant where the designers are co-located.

4.3 Contextual Analysis

The fact that design complexity is strongly context dependant has been stated in previous sections. Therefore, when modelling the complexity of the design process it is first essential to determine the context. Within the scope of this work it is proposed that a design context consists of four related areas these being work, time, motivational and social contexts. Figure 1, illustrates the constituent parts of each context.

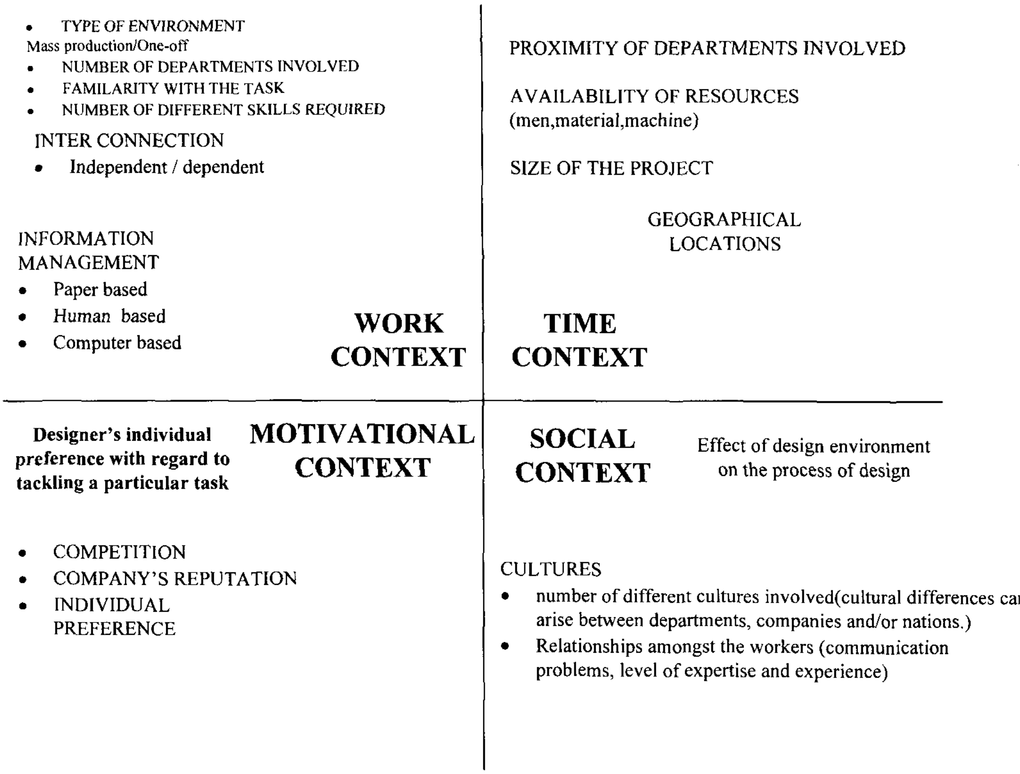


Figure 1. The Constituent Parts of a Design Context

Presently, the relationship that exists between the CGF's for specific instances of this design context are being investigated and modelled hierarchically.

5.0 Conclusions

In conclusion, there is a need to understand and quantify the complexity of the design process in order that design project schedules and costs can be estimated more accurately. Extensive research has been carried out in the area of design complexity, however:

- no complete set of complexity drivers have been identified;

- little is known about the relationship which exists between the parameters which drive complexity;
- very little has been established in developing a relationship between the complexity of the design process and the complexity of the final product;

This project is addressing these issues through the aims and objectives stated above. To date, a number of complexity drivers have been identified and the relationship that exists between them is being established and modelled based on a context dependant basis.

References

[1] Frizelle G. and Gregory M.J., 2000. 'Complexity and the impact of introduction new products', Int.Conf. on Complexity and complex Systems in industry, University of Warwick(UK)19-20 September,2000, page 248

[2] Calinescu, A., Efstathiou, J., Sivadasan, S., Schirn,J. and Huaccho Huatuco, L. 2000. 'Complexity in Manufacturing: An Information Theoretic Approach', Int.Conf. on Complexity and complex Systems in industry, University of Warwick(UK)19-20 September,2000, page 30

[3] Norris, K.P. 1971. 'The accuracy of project cost and duration estimates in industrial R&D', R&D Management, 2, pp 25-36

[4] Murmann, P.A. 1994. 'Expected development time reductions in the German Mechanical Industry', Journal Product Innovation Management, 11, pp 236-252

[5] DeMarco, T.1982. Controlling Software Projects, Yourdon Press, New York(1982)

[6] Salingaros Nikos A. 1997. 'Life and Complexity in Architecture From Thermodynamic Analogy', Physics Essays, Volume 10, pp 165-173

[7] Wiendahl, H.P. and Scholtissek, P. 1994. 'Management and Control of complexity in Manufacturing', Annals of the International Institution for production engineering research (CIRP), 1994 Vol.43/2 pp 533-540

[8] Stamm Von B., 2000. 'The effects of context and complexity in new product development', Int.Conf. on Complexity and complex Systems in industry, University of Warwick(UK)19-20 September,2000, page 63

[9] Hayes, R.H., Wheelright, S.C. and Clark, K.B. 1988. 'Dyanamic Manufacturing, New York: The Free Press

[10] Kim,D.H.1993. 'The link between individual and organizational learning', Sloan Management review, 35, pp 37-50

[11] Waldrop, M.M. 1992. *Complexity,* London:Penguin Science

[12] Gidado K.I. 1996. 'Project Complexity: The focal point of construction production planning', Construction Management and Economics, Vol.14, pp 213-225

[13] Pugh

[14] Paul & Beitz

[15] Ullrich & Eppinger

[16] Wallace, K.M. and Hales, C.1987. 'Detailed Analysis of an Engineering Design Project', Int. Conf. On Engineering Design ICED 87, Boston, MA 17-20 August 1987, pp 94-101

[17] Singh, K. 1997. 'The impact of technological complexity and interfirm cooperation on business survival', Academy of Management Journal, pp 339-367

[18] Campbell, D.J. 1988. 'Task Complexity: a review and analysis', Academy of Management Review 13, pp 40-52

[19] Frost P.and Mohoney, T. 1976. 'Goal – setting and the task process: an interactive influence on individual performance', Organizational Behaviour and Human performance, 17, pp 328- 350

[20] Invention Machine Corporation, www.invention-machine.com

This page intentionally left blank

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

REDUCING COMPLEXITY OF MODELLING IN LARGE DELIVERY PROJECTS

H Laurikkala, E Puustinen, E Pajarre, and K Tanskanen

Keywords: Complexity management, design strategy, competitiveness, computer supported co-operative work.

1 Introduction

In large one-of-a kind projects the costs and the schedule target are usually overrun. In managing the whole project lifecycle, the project strategy and start-up phases are crucial. The mistakes made in preliminary planning will cumulate to all later project phases. (Figure 1) In addition to these factors, the unexpected risks, the external influence and requirements and changes during the project make the project difficult to manage and increase the uncertainty aspect. [1] This all can be summed up to one word: complexity.

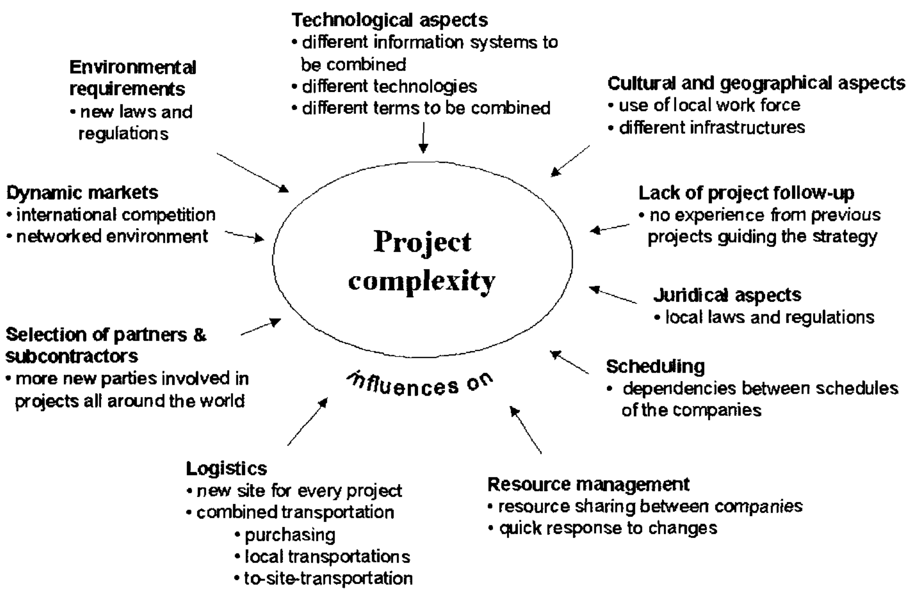


Figure 1. Factors affecting project complexity in the preliminary planning phase

When competing in the field of large, one-of-a-kind projects, the answer in many cases is not the R&D, design process or the efficient manufacturing, assembly and testing, because the companies are technically equal. The differences come usually from better service and better quality. The companies need help in managing the whole lifecycle of a project.

In order to decrease and manage the complexity of the whole supply chain of the project, the companies have to improve their risk management. This can only be done by concentrating on managing the essential project information – by modelling the whole project lifecycle. An effective information system is needed for modelling the real world complexity and simulating alternative project strategies before the project is started.

1.1 Aims and objectives

The aim of this paper is to describe how and why to reduce complexity of modelling large industrial projects. The paper is based on the experiences in modelling and simulating complex project deliveries. Both the challenge of creating a strategic information system for project contractors and the experiences of the companies using it are discussed.

1.2 Background and motivation

Companies operating with large projects have to face multiple problematic situations; the projects last for several years, the amount of contractors and subcontractors is high and the projects are carried out around the world. To increase the co-operation between four Finnish power plant contractors, the Prosit P9 project (1996-99) was started with the idea of developing a distributed and networked strategic information system called Simo. The project was carried out in Tampere University of Technology and continues now as VERSO project in close co-operation with indutrial partners working in the field of power plant contracting.

1.3 Research methods

The project started with a strategic study covering the future networking strategies of the case companies. It also included a SWOT-analysis about project management. A state-of-the-art study about existing related projects, product business and project modelling tools was done with the result that there were no existing tools for distributed large project management of a virtual organisation.

Case study was the most important method used in building a working tool for project management. The software is currently being tested in real projects of the case companies. After receiving results of the use, the group of companies is to be expanded to be used in larger projects, such as pulp and paper mills.

1.4 Related work

From the complexity point of view, modelling and product development in concurrent engineering are discussed e.g. by Belecheanu [2], Salminen [3] and Szczerbicki [4].

There are numerous process modelling tools. Flowcharting tools like Wizdom Systems, ProcessWorks! and DataWorks! and SyteCentre Business Process management from Symix Computer Systems are capable of depicting much of an enterprise's operations. The user can break down the day-to-day operations into functions and activities, define the process model, work-flows and their connections to Enterprise Resource Planning (ERP), and external information systems. [5]

The trend in flowcharting software is to pile on functionality, which makes the software more complicated and increases the complexity.

Enterprise simulation tools like Symix's Virtual Planning or KBSI's Witness need the basic data of capacities, flows, rates, cycle times and start and end dates, usually coming from an ERP system.

Workflow modelling, e.g. Holosofx E-process Suite [6], for Business Process Re-engineering (BPR), allows organisation and business process modelling, simulation and analysis. SAP AG [7] is an ERP system for product lifecycle management. It is meant for integrated product development and concurrent engineering.

These tools are planned for company-wise use, i.e. they do not support the networked co-operation of the companies in the distributed use, neither are they specialised for one-of-a kind products of complex projects.

2 Designing a modelling tool for industrial use

The starting point of the design process is the target (i.e. whom the tool is designed for). When trying to reduce the complexity of modelling, one has to start from designing the modelling method. If the tool is meant for everyday-use of industrial companies, the development of the method has to be done in co-operation with industrial partners. Only the users of the tool can decide whether the method is suitable for their purposes.

The reason for striving towards less complexity is that in the industry the added value does not come from technically detailed and complex methods or systems but from the advantages the companies can gain by using the system they understand. The system has to fulfil the following requirements:

- it has to be visual, and not difficult to use
- it has to include the most important phases from the whole project, such as contracting, defining the scopes, scheduling etc.
- it has to allow the data transfer to and from companies' own information systems in order to prevent the nugatory work caused by data entry from one system to another

The experiences gained from industrial use of the test version of Simo software have shown clearly that designing the graphical user interface as user friendly as possible is of utmost importance when taking a new tool into use. Former modelling methods used were difficult to use and did not allow any data transfer to and from companies' own systems. For more information, see [8].

In Simo software, a special emphasis has been placed on the possibility to vary the level of detail in modelling. Thus it is possible to concentrate on the most critical issues of the project (e.g. selecting subcontractors or defining scopes) whereas less important information can be modelled roughly or left out altogether.

3 Simo software

Simo is a Java-based, distributed, networked information system, operating in Internet environment. It consists of business, resource and product modelling, which form the base for the project modelling, where the contracts and schedules are made. The results of the simulation can be viewed in a separate analysis tool. Simo was launched into company-wise test use in the beginning of the year 2000. The Finnish power companies have made their own business, resource and product models, simulated them and analysed the results. The distributed use of Simo is to begin, when all the companies are familiar with the software. (Figure 2)

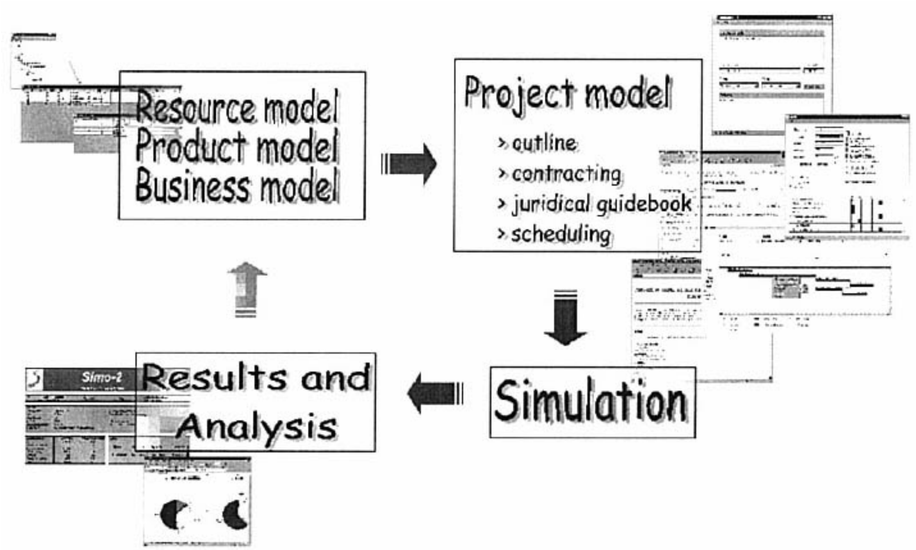


Figure 2. Using Simo software

3.1 Where can Simo be used?

There are separate systems for project management and planning, logistics, production management and product data management. However, there are no commercial systems, which simultaneously are specialized in large projects, cover the whole project lifecycle, enable both strategic and detailed views to the project, are compatible with companies' IT (e.g. MS Project and Excel), are enhanced with juridical guidebook, are distributed supporting consortium wide co-operation & networking and enable simulation (making trial and learning operation mode possible). Simo software is designed to fulfil all these needs.

Simo software is meant for managing complex, large one-of-a kind projects (power plant projects, pulp and paper mill projects, construction projects, shipbuilding projects etc). These kinds of projects contain huge amount of problematic situations: they last typically for three to five years, containing many changes and uncertainties during the projects. The consortium can have up to 10 participants each of them having many subcontractors each of them having many subcontractors. The projects are carried out all over the world, and the sites as well as partners and subcontractors vary from project to project.

Though, there are some limitations of the use. Simo is not planned for managing mass production or simple short-term company-wise projects, because it supports both the start-up phase with a lot of design and the networked co-operation of contracting companies.

The advantages of using Simo are numerous. For more information, see [9,10]. In a strategic level, Simo can be used for testing alternative and new ways to operate, it supports networking and co-operation between companies and it helps marketing in the project-planning phase. In an operative level, Simo can be used for finding out current practises, documenting, improving and unifying them and it can also be used as a tool for improving project follow-up.

4 Reducing complexity by modelling at different levels

Traditional business modelling systems allow modelling either at a strategic or a detailed level. This means that each business process has to be modelled in an equally detailed level even though they were not equally important for the company or well known by the modeller.

In Simo it is possible to model the most interesting and the most important processes at more detailed level than the less important ones. With modelling at different levels the amount of unnecessary guesswork can be reduced to minimum. (Figure 3)

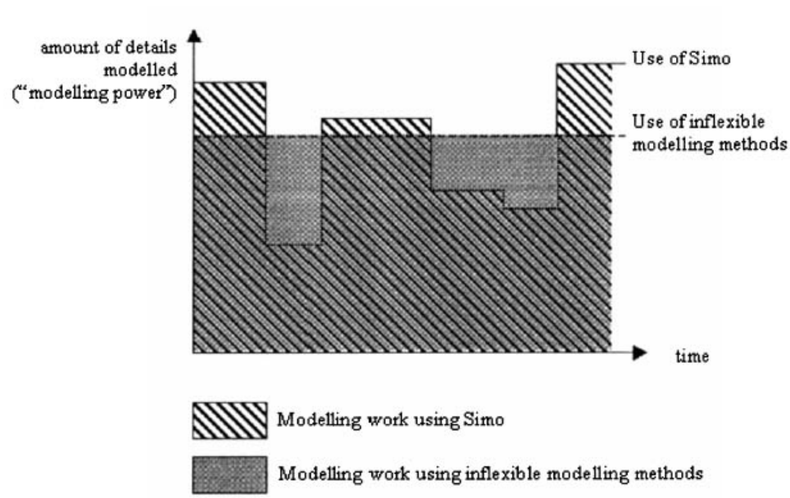


Figure 3. Varying the level of detail in modelling

A lot of variation for modelling levels can be reached with purely semantic means. On the implementation of the Simo tool, one of the key issues has been to provide also syntactic support for modelling in different ways for different needs. In the following, modelling of resources, products, business and projects is briefly described. For more detailed description of Simo modelling methods, see [8, 9].

Resource modelling of Simo includes information about companies (name, location and description), their facilities (name, location and description) and resources within facilities (amount and costs). It is worthwhile to enter all the aforementioned information for the model of one's own company but not for all the models of the subcontractors. The resource model of the subcontractor can contain only the subcontractor's name, location and, if the modeller wants, additional information about the subcontractor (e.g. quality and success in previous projects).

On product and business modelling phase the structure of the product (e.g. power plant or pulp and paper mill) is defined. This includes parts (physical parts or purchasing packages) and subparts in a hierarchical tree-like structure. Each part contains operations and documents that are done to the part during the delivery process. For these operations, possible suppliers are listed. In the simple modelling (e.g. modelling of subcontractors and partners), only delivery times and prices are defined for operations. In a detailed case (e.g. modelling of one's own business), operations are defined as activity chains with specified duration in work hours using resources modelled in a resource modelling phase.

Project models in Simo contain project outline, contracts between different partners, scopes of deliveries, and scheduling at the operation level. In simple models the contract structure can be flat, i.e. a company purchases supplies from Virtual Supplier and delivers the plant to Virtual Customer. Virtual Supplier is an imaginary company, which can supply any product or operation any time for any price. Similarly, Virtual Customer represents a company, which itself does not provide any services, and acts as a customer only. They do not have to be modelled in a resource-modelling phase. In a more detailed case the supply chain can be modelled and analysed more thoroughly by choosing alternative suppliers from different locations for different simulation runs.

It is not necessary to have detailed information from all the companies in the supply chain. And in the preliminary planning phase of a project it is not even known which companies are to participate. For early phase modelling it is possible to have template companies like "*A turbine supplier*" with generic product, price and delivery time information depending on the business area. When it is time to compare possible suppliers or the supplier is already chosen, a generic company can be switched to a specific one, like "*International Turbines Ltd.*".

Also scheduling can be modelled at different levels. A rough schedule relies on the pre-defined dependencies (modelled in a product- and business-modelling phase) between operations and uses default values for resource use. An elaborate schedule can contain extra dependencies, fixed operation starting dates and thorough control of resource use.

It has to be born in mind that however detailed and sophisticated the models and the simulator are, the results are of no use unless properly and efficiently analysed. Simo includes tailored company-specific Results and Analysis tools in MS Excel and MS Project, allowing data transfer to and from company's other information systems.

5 Industrial experiences of use

Industrial use of Simo software started in the beginning of the year 2000 in four large Finnish companies working in the field of power plant contracting. It was started by modelling of resource, product and business models. At first, the experiences of modelling were not promising because of the methods and tools used. [8, 9] The graphical user interface of one of the tools was too complicated and included too many windows. It was almost impossible for the user to get a clear view of the total data he had entered. The windows did not provide the user with the information about what he was modelling, so he had to scroll back to the starting point through many windows if he was not sure which operation or activity he was modelling.

It was impossible to transfer any existing information to or from the tool, everything had to be modelled again. This was the most important reason for the rejection of the tool.

Because there were no existing modelling methods or tools suitable for modelling large, one-of-a-kind projects, the modelling of resource, product and business models was implemented in Simo software tool. The users were interviewed about their modelling experiences and the new method was designed to fulfill their needs and to avoid complexity. The companies have been satisfied with the modelling with Simo software tool ever since (the development still continues). The new tool is easier to use, the start-up period is shorter and the modelling work is quicker. All the information is easily available and printable and the exchange to and from existing systems is simple. The companies have been able to model their current resources, product structures and business processes for the basis of project modelling. They have also found some of their weaknesses in documenting their present situation and the most important of all, they have been able to improve their situation by efficient modelling.

Project modelling has been a part of Simo software from the beginning. At first it was quite simple, including only the most important features of contracting, choosing the scope and scheduling. It has been developed cooperatively with the users and the work still continues. The project modelling is to be diversified in order to fullfil the needs of the users: the tool will support multi-project management, it will help in risk management by allowing the modelling of unexpexted changes, subcontractor selection will be supported by additional features, etc.

In a typical power plant project, a resource model contains information about up to 100 companies (own company, partners and subcontractors), few of which are modelled at the detailed level. Product and business models contain up to 100 parts, from one to six operations for each part, up to 15 documents for each part and up to ten activities for each operation and document.

Since most of the power plant projects last from three to five years, no new projects have been modelled or simulated with Simo yet. Though, it has been used for modelling numerous old projects in order to see whether the modelling method and the level of detail are correct. The use for new projects has started and so far, the results have been promising.

6 Conclusions

Simo software and modelling practices are the major results of the Prosit project. Simo is a software tool for planning, management and follow-up of large projects. It consists of a product, resource and business modelling part, a project modelling part and a simulator in Simo and of separate results and analysis tools in MS Excel and MS Project.

Distributed product development includes a lot of complexity. Most authors seem to approve of the fact and to concentrate on developing information systems trying to cover too many aspects of the whole process. Though, the experiences from the Prosit project show that in the industry the concentration should be based on reducing the complexity of the modelling phase. This makes management of the whole project supply chain easier.

The first step when starting to develop a modelling and simulation tool for industrial purposes should be getting familiar with the business environment and the processes of the companies. It is not possible to attain new developments unless the current state of operations is thoroughly found out. It is better to start with a simple, well-functioning version of an information system and to develop it step by step than to jump into a too advanced version at once.

Simo is a strategic information system for distributed project planning, management and follow-up. It is designed from the start to the end (from the meta-modelling part to the project modelling) with the idea of avoiding complexity. It is to help a project manager when starting a new project to avoid delays in start-up and to improve the information flow between different departments by providing them with a standardised way to operate.

Industrial experiences have shown that avoiding complexity has made the resulting tool a lot easier to use, shortened the start-up phase and sped up the real modelling work. The real benefits can only be gained when the industrial users can concentrate on the results, not on using the software itself.

References

[1] Mawby, D., Stupples, D.W., "Deliver Complex Projects Successfully by Managing Uncertainty", Proc. Complexity and Complex Systems in Industry, 19^{th} – 20^{th} September, Univ. of Warwick, UK, p. 155-166.

[2] Belecheanu, R., Pawar, K.S., Barson, R., Haque, B., "A Framework for Supporting Decision Making in Early Phases of Product Development in Concurrent Engineering"., The 15th International Conference on Production Research. 9th-12th August 1999, Limerick, Ireland.

[3] Salminen, V., Yassine, A. & Riitahuhta, A.: "Strategic Management of Complexity in Distributed Product Development", NordDesign 2000, 24-25 August, Copenhagen, 2000.

[4] Szczerbicki, E. & Reidsema, C.: "Modelling Support in Concurrent Engineering Environment", World Manufacturing Congress, WMC 1999.

[5] Gould, L.S. "Designing a Better Business", Automotive Manufacturing and Production, August 2000, pp. 62-65.

[6] Holodofx Inc. Product Description / E-Process suite – 2000. 2000.

[7] SAP AG. Collaborative engineering and Project management, 1999.

[8] Laurikkala, H., Tanskanen, K., Pajarre, E.: "Simo-2, a software for product structuring for large project contractors", NordDesign 2000, August 24-25, 2000, Copenhagen, Denmark.

[9] Laurikkala, H., Pajarre, E.: "Supply Chain Management with a Strategic Modelling and Simulation Tool for Virtual Supplies", Int. Conference on Advances in Production Management Systems, APMS'99, September 6-10, 1999, Berlin, Germany.

[10] Laurikkala, H., Tanskanen, K.: "Management of Large Projects in Virtual Organisations", 6th Int. Conf. on Concurrent Enterprising, June 28-30, 2000, Toulouse, France.

Heli Laurikkala
Tampere University of Technology, Machine Design, P.O. Box 589, 33101 Tampere, Finland,
Tel: +358 3 365 3665, Fax: +358 3 365 2307, E-mail: heli.laurikkala@tut.fi

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

SUCCESSFUL STRATEGIES FOR COMPLEX DESIGN IN EARLY PHASES OF THE PRODUCT DEVELOPMENT PROCESS – AN EMPIRICAL STUDY

B Bender, U Kammerer, F P Pietzcker, W Hacker, and L T M Blessing

Keywords: empirical study, descriptive models of the design process, design heuristics

1 Introduction, Aims and Objectives

This paper describes a two year empirical study investigating efficacy of established design methodology in the phases of conceptualisation and rough embodiment design. The study aims at verification and validation of the results of a field study into the *efficacy and learnability of design methodology*. In a *longitudinal study* the development of individual design strategies of design students is being examined, followed by a *cross-sectional study* into typical procedures used by experienced designers.

First results of the longitudinal study are presented here. These lead to a proposal for re-interpretation of conceptual design as being more an iteration loop of decomposition and aggregation rather than a hierarchically Top-Down process. This might be seen as a contribution to further development of design methodology and design education. The cross-sectional study is still running and will be finished by the end of 2001.

2 Hypothesis: Re-interpretation as a Flexible and Optional Heuristic Improves Efficacy and Acceptance of Design Methodology

Following conclusions from earlier studies (e.g. [5], [2]) it is assumed that the initial interpretation of design methodology as a linear-sequential prescription for design activities is the main obstacle to the efficacy and acceptance of design methodology in practice. A re-interpretation of design methodology as a flexible and optional heuristic leads to much better support of individual design performance and to a higher level of user satisfaction.

If those findings are correct, then it is to be expected that

- explicit design methodology education (DME) <u>will not instantly support</u> design performance of novices but may even reduce it;
- explicit DME <u>will support</u> design performance after having been used and internalised, so strategies and methods can be applied flexibly and task-oriented instead of being carried out as schematic procedures.

- experienced designers without explicit DME but with high design performance, use implicitly acquired procedures from design methodology, normally without being able to make these explicit.

To validate these assumptions, different individual design styles and strategies are examined, as well as the assumed influences of DME and professional practice on design success and on the development of design procedures.

3 The Nature of the Study

A two-phase study has been set up:

- A **longitudinal study**, investigating the development of individual design styles and strategies by following the progress of engineering design students with and without explicit DME;

- A **cross-sectional study**, investigating the individual design styles and strategies of experienced designers with and without explicit DME.

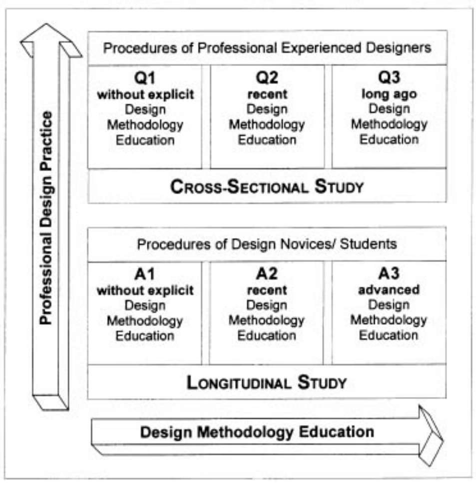


Figure 1. Design of the two-phase study

Within the longitudinal study (see Table 1) mechanical engineering students with different levels of explicit DME - identified by examining educational and professional background – each were confronted in a laboratory environment with a *conceptual design task* and an *embodiment design task* at three predetermined stages of their study. The conceptual design tasks consisted of a verbal description of a design problem without any visualisation, and had to be finished within one hour. The embodiment design tasks comprised a verbal description, a sketch of the working structure and technical data. This task had to be finished within 3.5 hours.

Both the conceptual and the embodiment design tasks had been chosen out of common areas of basic engineering design and were characterised by very similar requirements and constraints. The homogeneity of all tasks with respect to the criteria *conflicting aims*, *complexity*, *transparency*, *degrees of freedom* and *required knowledge* has been validated by design experts using a method for systematic analysis of design problems and tasks [7].

Table 1. Schedule of the longitudinal study

'A1-Phase'	'A2-Phase'		'A3-Phase'	
end of 4th term	**end of 6th term**		**end of 7th term**	
test group	**test group**	**control group**	**test group**	**control group**
design students at the end of their basic design course	design students having taken part in a one-year Design Methodology lecture	design students not having taken part in the Design Methodology lecture	design students having taken part in the 6 month Design Methodology project	design students not having taken part in the Design Methodology project

For observation and analysis of the individual procedures for both tasks, the intended strategy, the actually realised (observed) strategy and the retrospectively recognised (recapitulated) strategy of each participant were observed and categorised using appropriate empirical research methods (for further details see [1])

Within the cross-sectional study, which is currently running, experienced designers with different levels of explicit DME are confronted with a novel design task for which a rough embodiment design is to be produced. The results of this study are to be compared to the results of earlier studies on the design behaviour of experienced designers

4 Success and Failure: The Evaluation of Design Performance

Design success indicates the realisation of optimum design quality under given certain constraints. The design of the study ensures maximum comparability and constant design constraints. For the assessment of design quality, a formalised method based on strategies of value analysis has been developed to achieve maximum evaluation reliability and reproducibility. Design quality was measured in three different sub-categories:

- quality of the documented *concept variants*
- quality of the documented *evaluation of concept variants*
- quality of the documented *embodiment designs/ assembly drawings*

For each of these sub-categories a multi-criteria set of objectives had been formulated containing those features that describe design success in the specific sub-category (Table 2). "Key-questions" enable access to the different partial objectives and avoid misunderstandings between evaluating experts to increase inter-assessor agreement. Multi-criteria evaluation charts based on value analysis, allowed the evaluation of design performance in every subcategory using the objectives and criteria mentioned above, their relative weights and ordinal value scales. To further increase evaluation reliability a team of three design experts carried out the assessment.

Table 2. Objectives/ criteria and "key-questions" for the assessment of design performance

quality of concept variants	
criterion	**key-question**
quantity	How many concept variants have been found/ elaborated?
independence	On how many different working principles are the concept variants based?
transparency	Are the main features of the concept variants discernible and assessable?
function	Is the stipulated overall-function fulfilled?
simplicity	Can the elaborated concepts be realised with reasonable effort?

quality of the test participants' evaluation of concept variants	
criterion	**key-question**
results of evaluation	Are the results and conclusions accurate?
transparency of evaluation	Is the evaluation comprehensible and reproducible?
systematic evaluation process	Have all necessary steps of an appropriate evaluation process been carried out?
technical and economical criteria	Have common technical and economical evaluation criteria been considered?

embodiment design/ assembly drawing	
criterion	**key-question**
function	Is the stipulated function fulfilled?
layout/ embodiment	Is the overall layout durable and have relevant guidelines of embodiment design been attended?
manufacturing	Is manufacturing possible?
assembly	Is assembly possible?
safety/ reliability	Does the overall design prevent danger and failure?
simplicity	Is the overall design simple?
clarity	Are all states of the system discernable and unambiguous?
completeness	Is the overall design complete?

5 Analysis of Individual Design Procedures Using Action Regulation Theory and Descriptive Statistics.

For analysis and categorisation of observed design procedures, an often used approach is the analysis of the number and magnitude of deviations of an observed design strategy from a postulated reference strategy, see e.g. [2]. Commonly used reference processes are design methodologies such as the German Guideline VDI 2221. The longitudinal study described here does not only focus on the hypothesis-driven validation of typical design procedures. Two approaches are combined.

The more *data-driven* approach uses descriptive statistics to examine correlations of DME and design performance. This may lead to the identification of new design procedures. The results of this analysis are presented in the following paragraph.

The more *hypothesis-driven* approach defines basic design operations derived from common design theory. These are used to postulate different plausible and expected procedures based on concepts of Cognitive Psychology, Action Regulation Theory (e.g.[3]), and Design Methodology. For more details see [1]. This part of analysis has not been finished yet.

6 First Results

To investigate the influence of the different stages of explicit DME on design performance, the mean values of the credit points each participant obtained for the three categories of design quality have been tested. Due to the decreasing number of participants in the observed phases, initially observed differences between the means were often not statistically significant. However, they still represent a distinct tendency, worth further research. In this section we clearly distinguish between significant results and tendencies.

A maximum of 60 credit points could be obtained for the embodiment design task. Participants achieving less than 25 credit points were eliminated from all three categories, based on the assumption that achieving a design quality below a certain level is a sign of a fundamental design incapability and therefore the design procedure followed by these people should not be compared with the procedure of other designers.

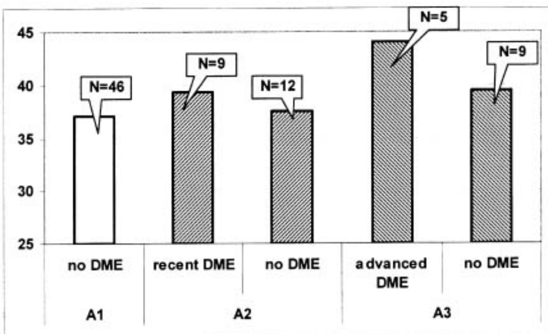


Figure 2. longitudinal trend of quality in embodiment task (means)

As shown in Figure 2, recent DME had no significant positive influence on the embodiment design quality, but a strong tendency can be observed towards a positive influence of advanced DME (A1/A3DME significant: p=0,050; A2 without DME/A3DME tendency: p=0,141; A3DME/A3 without DME tendency: p=0,133). Within the group without DME, no significant change in embodiment design quality could be identified (A1/A2 *and* A3 without DME insignificant: p=0,716). These findings match the hypotheses very well.

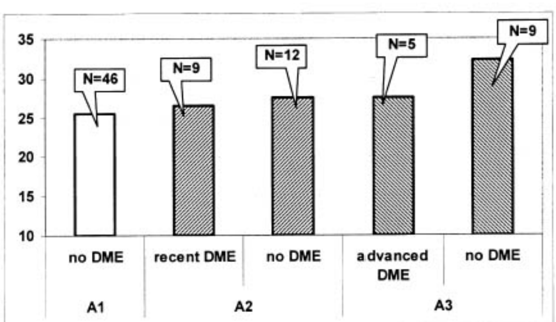


Figure 3. longitudinal trend of quality in conceptual task (means)

For the quality of concept variants also a maximum of 60 credit points could be obtained. Very good solutions were expected to achieve more than 30 credit points. As shown in Figure 3 no positive influence of DME could be identified (A1/A2DME insignificant:0,933; A1/A3DME insignificant p=0,849). To the contrary, a tendency towards lower quality in the DME groups might rather be assumed. These findings contradict the hypotheses, but match the often mentioned experiences from teaching and practice indicating discomfort with strictly hierarchical conceptualisation.

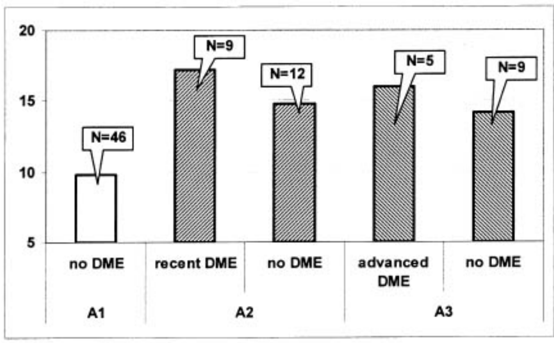


Figure 4. longitudinal trend of quality in concept evaluation (means)

The quality of concept evaluation could maximally result in 40 credit points. As shown in Figure 4 - and as expected – the quality of evaluation of inexperienced novices (A1) is low. The evaluation performance for A2 and A3 is much better for all students, irrespective of DME background, although DME shows a light tendency towards a more positive effect on evaluation competency (differences of means within A2/A3 insignificant: p>0,4; A1/A2DME *and* A3DME significant: p=0,036; A1/A2DME tendency: p=0,071 but A1/A3DME insignificant: p=0,323). This tendency is underlined by a very strong correlation of a *systematic evaluation process* (p=0,000, ρ=0,758) on the one hand and the consideration of *technical and economical criteria* (p=0,000, ρ=0,676) on the other with the *results of evaluation* over all participants (see Table 2). These findings match the hypotheses to a good degree.

In summary, DME and particularly advanced – partly internalised – DME, has a distinct positive effect on design success for embodiment design tasks. For conceptual design, DME shows a tendency to improve concept evaluation. As mentioned above, it is assumed that participants with advanced DME apply strategies and methods flexibly and task-oriented, rather than follow schematic procedures. This is to be validated in the in-depth analysis of the individual design procedures.

DME showed a tendency towards a negative influence on the capability of solving conceptual tasks. A first in-depth analysis of the conceptual procedures of the DME students revealed a sometimes almost obsessive focus on early abstraction and an obvious suppression of early spontaneous conceptual ideas. As a consequence, some self-evident solutions for embodiment-determining functions were neglected or ignored in favour of exotic ones and of the early variation and optimisation of auxiliary functions. It is proposed - and indicated e.g. by another study into naive design procedures still running at TU Dresden - that this way of applying design methodology is artificial and finally frustrating.

An approach in which conceptualisation is viewed as a feedback-driven iterative process rather than a strictly hierarchically and top-down process, matches the considerations of Cognitive Psychology and Action Regulation Theory much better. In compliance with the Task-Episode-Accumulation approach [8] and the recognition that design activity tends to be opportunistic for the purpose of "cognitive cost-effectiveness" [9], the re-interpretation of the conceptual design process as an iteration loop of *decomposition* and *aggregation* is proposed (see Figure 5). The importance of iteration within the conceptual design phase has been pointed out in design methodology literature (e.g. [4], p.139). However, this has not been reflected in the process models nor in the appropriate integration of the many solution finding methods into these process models

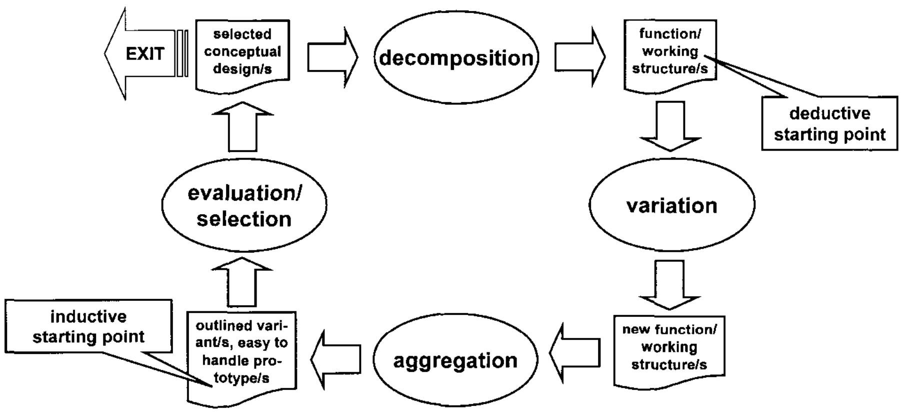


Figure 5. re-interpretation of conceptual design process model as a feedback proven iteration loop

Decomposition and aggregation refer to a logical and hierarchical breakdown using different levels of abstraction and representation. Conceptualisation can start at two points. At the *inductive* starting point designers may start with one or more concrete, but probably ill-defined, concept variants, based on an intuitive search for solutions – e.g. using methods like the gallery method ([4] p.78). The resulting variants can then be refined by following the other steps in the loop. At the *deductive* starting point, the designer starts at a higher level of abstraction, as proposed in classical design methodology. No matter which starting point is chosen, the whole loop has to be carried out according to the requirements of the actual task, to prevent early fixation and neglecting innovative solutions.

The creation of new solutions should not only rely upon variation and experimentation at the abstract levels of function and working principle. The aggregation phase is crucial for solution finding. Outlining a solution by sketching or impromptu-modelling does not only illustrate an idea, but is an important creation technique in its own right, supporting the mental interaction of internal and external activity [6]. Here the proposed integration of intuitive solution finding methods into the process model is taking place.

7 Conclusion

The empirical study described in this paper aims at the identification and determination of successful and less successful design procedures in order to improve individual design performance in the phases of conceptualisation and rough embodiment design.

As a first result, the systematic and hierarchical approach of design methodology still seems to support design approaches that take advantage of hierarchical strategies. This seems to hold for direct design steps, such as rough embodiment design, and indirect design steps, such as the evaluation of conceptual solutions. For design steps taking advantage of spontaneous association and intuitive creation, strictly hierarchical strategies seem to be artificial and inappropriate. Following the consideration that the interpretation of design methodology as a linear-sequential prescription does not encourage the intended improvement of individual design performance, it is assumed that a re-interpretation of design methodology as an optional heuristic, to be used as a toolbox, flexibly and well adapted to the potentials and restrictions of each design situation, will lead to increased efficacy and acceptance of design methodology.

References

[1] Bender, Bernd, Kammerer, Uwe, Pietzcker, Frank, Blessing, Lucienne Hacker, Winfried, „A Systematic Approach to Observation, Analysis and Categorisation of Design Heuristics", International Conference on Engineering Design ICED 01, Glasgow 2001

[2] Fricke, G., Konstruieren als flexibler Problemlöseprozess – Empirische Untersuchung über erfolgreiche Strategien und methodische Vorgehensweisen beim Konstruieren, VDI-Fortschrittsberichte, Reihe 1, Nr. 227, Düsseldorf 1993

[3] Hacker, Winfried, Konstruktives Entwickeln als Tätigkeit – Versuch einer Reinterpretation des Entwurfsdenkens (design problem solving), Zeitschrift für Sprache & Kognition 18 (3/4), pp. 88-97, Bern 1999

[4] Pahl, G. and Beitz, W.: Engineering Design – A systematic Approach, 2Rev.Ed., London 1996

[5] Rückert, Carsten, „Untersuchungen zur Konstruktionsmethodik – Ausbildung und Anwendung", VDI-Fortschrittsberichte, Reihe 1, Nr. 293, Düsseldorf 1997

[6] Sachse, Pierre, Unterstützung des entwerfenden Problemlösens im Konstruktionsprozess durch Prototyping, in: Sachse, Pierre, Specker, Adrian: Design Thinking, pp. 68-145, Zürich 1999

[7] Schroda, Frauke, Die Analyse der Anforderungsstruktur konstruktiv-schöpferischer Probleme, in: Sachse, Pierre, Specker, Adrian: Design Thinking, pp. 8-65, Zürich 1999

[8] Ullman, D.G., Dietterich, T.G. and Stauffer, L.A.: A model of the mechanical design process based on empirical data, Artificial Intelligence in Engineering Design and Manufacturing 2, pp. 33 – 52, 1988

[9] Visser, W., Organisation of Design Activities: Opportunistic, with hierarchical episodes, Interacting with Computers 6, pp. 239-274, Oxford 1994

Dipl.-Ing. Bernd Bender, Dipl.-Ing. Uwe Kammerer, Prof. Dr. Ir. Lucienne Blessing
Technical University of Berlin, Dept. of Mechanical Engineering and Transport Systems, Engineering Design and Methodology, Sekr. H 10, Strasse des 17. Juni 135, D-10623 Berlin, Germany, Tel.: ++4930 314-24309, Fax: ++4930 314-26481, E-mail: bernd.bender@ktem.tu-berlin.de

Dipl.-Psych. Frank Pietzcker, Prof. Dr. rer. nat. habil. Winfried Hacker
Technische Universität Dresden, Fakultät Mathematik und Naturwissenschaften, Allgemeine Psychologie, D-01062 Dresden, Tel: ++49351 463-6890, Fax: ++49351 463-3522, E-Mail: pietzcker@psy1.psych.tu-dresden.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A METHODOLOGY FOR COMPUTATIONAL DESIGN TOOL RESEARCH

R H Bracewell, K Shea, P M Langdon, L T M Blessing, and P J Clarkson

Keywords: computer-aided design; systematic product development; knowledge-based engineering; empirical study; user evaluation

1 Introduction

A difficulty with research in computational design methods is in evaluating the capabilities and merit of the fundamental research output beyond initial benchmark tests that compare to theoretical examples. While this is a necessary step in method development, issues involving integration of computational methods within the design process are often overlooked. This need for better integration of tools throughout method development, was highlighted as an important issue for the future of design research at the UK DTI/EPSRC 1999 Workshop [1] as well as by Vergeest et al. [2] in the context of novel design tools. Fast feedback cycles from experimental use in education and practice could lead to the development of methods that are both theoretically capable and suited to achieving these capabilities within varying design processes. Thus, as researchers, we need the necessary tool set and support to rapidly prototype research systems without being unnecessarily hindered by implementation details and fast changes in computer technology and standards.

The development of computational design methods and tools requires knowledge of current advances in computer science, software engineering and hardware technology as well as familiarity with the application domain. One strategy for developing computational design methods is to split the tasks where the engineer, or domain expert, conducts the methodological research on paper and then hires a computer scientist to implement a proof-of-concept system. However, with today's programming job market, most research projects can not attract the necessary talent to implement the frequently complex software specifications and domain details. In addition, working out a method on paper can lead to difficulties in laying out diagrams of complex knowledge structures, making and propagating changes to data representations, and navigating the complex information spaces involved. These difficulties can make it very hard to address crucial issues of representation and logic in the early stages of method development. The result can be that a partial and computationally flawed specification is supplied to the programmer, leading to an implementation that most likely will not adequately reflect the method intentions and merits.

There is a common belief that software implementation of advanced design tools merely requires the application of mainstream software engineering techniques and thus is not a subject with which design researchers need be concerned. However, computational design research often entails a software design task that is highly specialised, involving the method complexity associated with numerical techniques combined with the human-computer interface issues of traditional CAD software. Additionally, it may be necessary to select and integrate advanced AI techniques with a range of continuously changing mainstream engineering software modules and packages. A welcome indication that the importance and distinctiveness of this subject is starting to be recognised, is the introduction of courses teaching the fundamentals of Cae tool development at EPFL [3], CMU, Berkeley and elsewhere. While a thorough grasp of the fundamental principles taught in such courses is vital, constant awareness of the possibilities offered by the latest developments in information technology (IT) is also essential. However this demands a great breadth of experience and more investment of time than can generally be afforded by most academics, programmers and students involved in design research projects. In the related context of developing complex software product families, Meyer and Lehnerd [4] emphasise the critical importance of awareness of new generations of software tools, operating systems, and hardware as they often provide solutions to difficult problems and barriers confronting a software team.

The aim of this paper is to propose a practical methodology for Computer-aided engineering Design (CaeD) tool research, that is intended to enable the development of useable computational design tools early on in research projects. A distinctive feature is the incorporation of CaeDRe, a coherent, modular, up to date and extensible development environment providing a choice of high level languages, tools, reusable software components, integration mechanisms and code-hardening mechanisms. CaeDRe is described in more detail in Bracewell and Shea [5].

2 Defining a Methodology for Computational Design Tool Research

The methodology presented stems from the unification of three complementary approaches developed by the authors over years of practical experience in CaeD research. The first element is the design research methodology developed at the Cambridge EDC by Blessing and Chakrabarti [6], from which a four stage process has been adopted; see Figure 1. In the first stage, *Criteria*, measurable success criteria for the design tool are identified, such as "reduced time to market" together with chains of causal influences linking back to overall business objectives such as "increased profit". The second stage, *Description I*, analyses the existing design process to discover relations between the measurable criteria and the design process thus identifying where application of a design tool could lead to improvements. In the third stage, *Prescription*, insights gained in *Description I* are used to create a storyboard for an improved design process that could result from using the new design tool. For computer-based tools, this storyboard creates a starting point for specifying and implementing a prototype system. Finally in Description II the design tool is tested experimentally to determine whether it works as intended and whether it actually impacts the measurable success criteria. While this is one sequence that can be taken through the methodology, the case study presented in this paper will illustrate that it is not necessary to follow these stages in one specific order.

Software implementation issues were not a primary focus of the previous methodology as it was intended to be applicable to the development of both computer-based and manual design tools. Extending it to provide support on software design and the use of up-to-date computer science techniques as well as empirical methods for method evaluation throughout the research project has resulted in the CaeD tool research methodology shown in Figure 2. Software engineering issues are now addressed by incorporating a second approach, the Cae tool development methodology defined and taught at EPFL [3]. The development process is divided into five activities: (1) task definition, (2) choice of representations, (3) choice of methods, (4) definition of visualisation, interaction and distribution strategies and (5) theoretical and experimental validation. If the choices of knowledge level and implementation level representations are explicitly separated, and a distinction is made between non-controlled and controlled experimental testing, the process fits neatly into the general methodology shown in Figure 1.

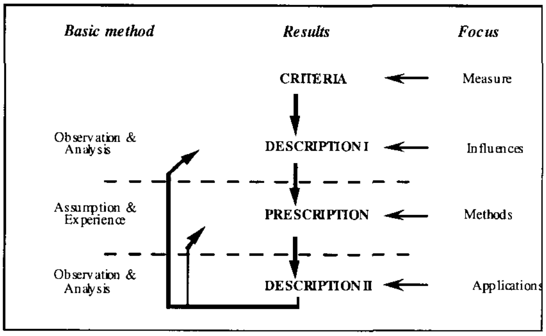


Figure 1. Design Research Methodology (Blessing, Chakrabarti and Wallace, 1998)

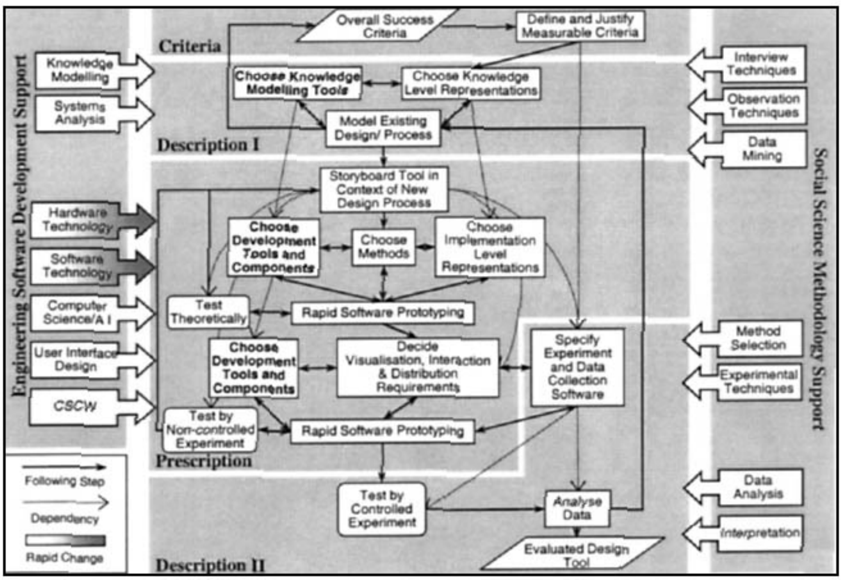


Figure 2. Computer-Aided Engineering Design (CaeD) Research Methodology

There are two main forms of method validation in CaeD design: (1) theoretical validation, i.e. comparisons to known benchmark problems and (2) experimental validation, i.e. testing with user groups. With some CaeD tools, such as production systems, method development often stops after comparisons to benchmark problems but does not continue to be validated empirically. The researcher usually performs the benchmark tests themselves thus, attention to visualisation, user interaction and distribution requirements as well as seamless integration mechanisms with other packages are not a priority. In some cases, for example automated routine design tools, there are explicit and commonly agreed criteria to assess the merit of a method, and thus theoretical validation may be sufficient. However, as we move to methods that are intended to enable design exploration and innovation, such claims can only be made in a theoretical sense rather than a practical one. While theoretical validation is a necessary step in any method development, it is difficult to assess the effectiveness of a computational method in a practical sense without creating a reasonable prototype system.

The software design support necessary to create robust, functional prototypes, as well as integrate them into conventional and experimental design systems is provided by the third element of the methodology. This is the use of an integrated design research platform. The concept was initiated in the mid '90s at several design research centres all needing to integrate multiple computational design research projects into a unified whole. The off-the-shelf software development environments available at the time proved inadequate thus requiring awkward, ad hoc connections with other large, monolithic Cae and AI packages. In order to circumvent these problems, software platforms were designed to exploit the growing body of high quality, open source software becoming available and provide the capability of easily inserting individual research projects as extension modules. Examples of software platforms developed as part of large design research projects are SEED [7], n-dim [8], the Framework for Knowledge Intensive Engineering [9], DIICAD Entwurf [10] and the Schemebuilder Development System [11]. The last example, designed and assembled by the first author, was then developed further, exploiting recent advances in software technology, to create the Integrated Functional Modelling (IFM) Development System [12].

While use of the IFM Development System has proved highly effective in the functional modelling research domain, its monolithic nature restricted application to other projects within the research centre. This was due to the fact that the system philosophy was to provide a single preferred solution covering a wide range of high level functional requirements in a tightly co-ordinated whole. In strongly coupled research projects such as Schemebuilder and IFM, this worked well. However, in more loosely related but still potentially complementary projects often steered by different investigators with diverse software backgrounds, preferences and industrial collaborators this model is not appropriate.

A solution lies in the product platform concept. A product platform, as defined by Meyer and Lehnerd [4], is "a set of sub-systems and interfaces that form a common structure, from which a stream of derivative products can be efficiently developed and produced". Companies in diverse domains have successfully adopted product platforms, for example Hewlett-Packard for ink-jet printers, Black & Decker for power tools, and Visio Corp. for diagramming software. CaeDRe results from reworking the IFM Development System as such a product platform for computational design tool research where the "products" are modular, prototype tools that are sufficiently functional and robust for evaluation by industry. Using the product platform approach emphasises that it is not the choice of individual functional solutions that is vital but rather importance should be placed on definition of interface specifications to form a flexible but coherent architecture. Facilitating integration of research tools can remove the practical obstacles that frequently exist in evaluating separate but complementary design tools in combination. The design and structure of CaeDRe is described in detail in [5].

The use of CaeDRe in the methodology is shown in Figure 2 by the three bold "**Choose Tools/Components**" tasks, one for each of the three main iterative cycles in the research process. These are respectively the modelling of the existing design and/or design process, the definition and implementation of the method and finally the transformation of the fundamental method into a usable prototype tool with suitable user interface. A further insight gained from the work on the Schemebuilder Development System was the realisation that, from Human Computer Interface (HCI) considerations, the integrated software platform also provided ideal facilities for automated data collection and analysis for the empirical evaluation of prototype design tools [13]. These considerations are incorporated in the methodology in the "Specify Experiment and Data Collection Software" task.

The side bars of Figure 2 show the incorporation of current software development and social science techniques that are necessary to the successful development of CaeD methods and tools. While in most cases provision of appropriate on-line guidance is sufficient, this should be emphasised with human expertise.

3 Case Study: Transforming Structural Shape Annealing into eifForm

The CaeDRe platform, a key element in the presented methodology, has recently been applied to transform a proof-of-concept implementation of the structural shape annealing method into a useable prototype system called eifForm. eifForm is an experimental computer-aided design system for structural synthesis aimed at providing architects and structural engineers with an enhanced means for lateral exploration of design possibilities. The generative method, called structural shape annealing, is composed of the following modules: a structural shape grammar, a structural analysis integration mechanism, simulated annealing optimisation and design performance evaluation algorithms. The method enables automated generation of innovative, performance driven free-form structures, providing both structural solutions and conceptual stimuli [14].

To illustrate the CaeD methodology presented in Section 2, the development history of structural shape annealing will now be placed within this context (Figure 3). Bold numbers relate to activities in the figure. The project began with a goal of creating a method that provided means for lateral exploration of planar truss structures using a grammatical formalism [15] as the knowledge level representation **1**. The conceptual design process for truss structures was modelled **2** considering a wide range of behavioural and performance issues based on design criteria discussed by Billington [16]. To allow for future extensions to three-dimensional trusses, the implementation level representation was chosen **3** as a modified winged-edge data structure inspired by work done by Heisserman and Woodbury [17]. Simulated annealing was chosen as the optimisation method **4** as a non-gradient based method was required for the representation and objective function desired [18]. Additional methods for rule selection and application were also developed. In the choice of development tools and components **5,** Unix-like systems were chosen as the OS platform, with ANSI C the development language, Emacs the coding environment, Geomview for 3D visualisation and FElt for finite element analysis. As theoretical testing was done by the researcher, visualisation, user interaction and distribution requirements were not explicitly addressed at this point. This implementation **6** only included utilities for translating final designs to selected analysis, CAD and visualisation file formats. Further details can be found in [5].

Until now, several papers have been published providing theoretical validation **7** of the method through comparisons to benchmark structural optimisation problems, for example single-layer space frames [14]. Further work successfully applied the method to the redesign of full-scale transmission tower for an energy company [19], with an overall success criterion **8** of reducing life-cycle costs of transmission lines. The company knew that costs were strongly related to structural mass, so reduction of mass was agreed as an easily quantified measurable criterion **9**. Tests have also been done **7** that compare theoretical capabilities for generating innovative solutions in comparison to human designers [20]. The next stage in method and system development is evaluating the capability for *achievable* innovation by designers using the system directly.

The proof-of-concept implementation was tested by non-controlled experiment **10** in collaboration with the Design Studio of the Future (DSoF) at MIT. In the studio, the system was used to explore generative design methods in the context of creating structural form and to assess the potential for enhancing design exploration and innovation. The expanded storyboard that resulted **11** is a digital design process+, i.e. non-digital mediums are always necessary, where the method is a key module within a process that incorporates many other current technologies.

The process starts with creating more elaborate and purposeful models of design scenarios through integration with CAD packages, digital topography maps, and material databases. Design generation using structural shape annealing is then invoked as either a fully automated or interactive process. The final designs are then transferred to visualisation and analysis tools, as the resulting designs often lie beyond the experience of the designer requiring attention to proper interpretation. This interpretation often leads to refinement of the input models for further design generation. Finally, selected designs are transferred to CAD/CAM tools for making physical models, using rapid prototyping.

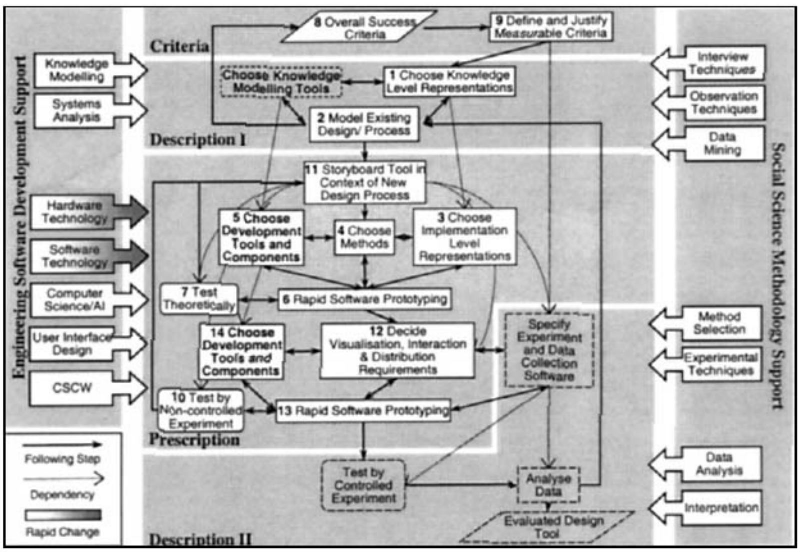


Figure 3. The CaeD Methodology as applied to eifForm

It was found, not surprisingly, that the level of implementation of the proof-of-concept system hindered the students' exploration of the theoretical method capabilities. Effective use of structural shape annealing involves expression of design intent through definition of geometry, structural behaviour and search models as well as proper interpretation and visualisation of generated designs. It is only through this process that designs that meet intent and are constructable will be generated. Important elements in creating an innovative design environment that will enable designers to access the theoretical capabilities of the method are user interaction and seamless integration with up to date digital technologies. As with most computational systems involving some level of automation, e.g. numerical analysis and rendering software, proper modelling will lead to more effective use. Equally important is transparency of the underlying method to foster user understanding of the algorithms. These key elements for effective use all lie within the categories of visualisation, user interaction and distribution **12** that to now have not been formally addressed.

Further development of eifForm **13** will now be carried out using the methodology presented towards creation of a system for use in education and practice. The choices of development tools and components using the CaeDRe platform **14** are described in [5]. Its application in transforming structural shape annealing into eifForm is intended to create a modular, maintainable, extendable and customisable system for additional method testing in both non-controlled and controlled experiments. It is hoped that fast feedback cycles from experimental use will produce method and system enhancements that will lead to a computational design environment for exploration of innovative structural solutions.

4 Conclusions

This paper described a promising new methodology for practical, high level support of CaeD tool development within a research group setting. The methodology aims to provide a systematic process for producing evaluation ready prototype systems targeted at improving current and future designs and design processes. It is intended that using the methodology will lead to development of more appropriate methods suited to adoption in practice. The degree of merit of the methodology will only be truly established by comparing the capacity for integration and evaluation of tools produced using it with those derived from alternative approaches.

Acknowledgements

The authors would like to thank the UK EPSRC for providing funding for this work.

References

[1] Culley, S.J., *Future Issues In Design Research(FIDR) Workshop*, Final Report, EPSRC/DTI, 1999.

[2] Vergeest, J.S.M., Kleinhuis, S., Wiegers, T., Opiyo, E.Z., Lennings, A.F. and Horvath, I., "On Industrial Evaluation of Novel Design Tools", in *International Conference on Engineering Design, ICED 99*, Munich, v3 pp1857-1860, 1999.

[3] Raphael, B., Shea, K. and Smith, I.F.C., "A task and software independent CAE course", in AICIVIL-COMP99: *The fifth international conference on the applications of AI to Civil and Structural Engineering*, Oxford, England, 1999.

[4] Meyer, M.H. and Lehnerd, A.P., *The Power of Product Platforms*, The Free Press, New York, 1997.

[5] Bracewell, R.H. and Shea, K. CaeDRe: A Product Platform To Support Creation And Evaluation Of Advanced Computer Aided Engineering Tools, Ibid., 2001.

[6] Blessing, L.T.M., Chakrabarti, A. and Wallace, K.M., "An Overview of Descriptive Studies in Relation to a General Design Research Methodology", in *Designers - The Key to Successful Product Development*, ed. Frankenberger, E., Badke-Schaub, P. and Birkhofer, H., pp42-56, Springer, 1998.

[7] Flemming, U. and Woodbury, R., "Software Environment to Support Early Phases in Building Design (SEED): Overview", *Journal of Architectural Engineering*, v1, n4, Dec 1995, pp147-152, 1995.

[8] Subrahmanian, E., Reich, Y., Konda, S L, Dutoit, A, Cunningham, D, Patrick, R, Thomas, M, Westerberg, A W, "The n-dim Approach to Creating Design Support Systems", in *ASME DETC,* Sacramento, California, 1997.

[9] Tomiyama, T., Kiriyama, T. and Umeda, Y., "Toward Knowledge Intensive Engineering", in *Int. Workshop on Engineering Design*, Lancaster, pp319-337, 1994.

[10] Grabowski, H. and Lossack, R.S., "Knowledge Based Design of Complex Products by the Concept of Design Working Spaces", in *IFIP TC5 WG 5.2 International Conference on Knowledge Intensive CAD*, Pittsburgh, PA, USA, pp79-98, 1996.

[11] Bracewell, R.H., Bradley, D.A., Chaplin, R.V., Langdon, P.M. and Sharpe, J.E.E., "Schemebuilder: A design aid for the conceptual stages of product design", in *9th Int'l Conf on Engineering Design ICED'93*, The Hague, Heurista, pp1311-1318, 1993.

[12] Bracewell, R.H. and Johnson, A.L., "From Embodiment Generation to Virtual Prototyping", in *International Conference on Engineering Design (ICED)*, Munich, Germany, WDK, v2, pp685-690, 1999.

[13] Langdon, P.M. and Cheung, H.F., "A Hypertext Documentation System for Schemebuilder: The 'Mechatronics' Book", in *Joint Hungarian-British Mechatronics Conference*, Budapest, 1994.

[14] Shea K., and J. Cagan, “Innovative Dome Design: Applying Geodesic Patterns with Shape Annealing,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **11**:379-394, 1997.

[15] Stiny, G., “Introduction to Shape and Shape Grammars,” *Environment and Planning B*, **7**:343-351, 1980.

[16] Billington, D.P., *The Tower and the Bridge: The New Art of Structural Engineering*, Basic Books, New York, 1983.

[17] Heisserman, J., and Woodbury R, “Geometric Design With Boundary Solid Grammars,” in *Formal Design Methods for CAD (B-18)*, J.S. Gero and E. Tyugu, eds., Elsevier Science B. V., North-Holland, pp. 85-105, 1994.

[18] Swartz W., and C. Sechen, "New Algorithms for the Placement and Routing of Macro Cells", *IEEE proceedings: Cat No. 90CH2924-9, IEEE Conference on Computer-Aided Design*, Santa Clara, CA, November 11-15, pp. 336-339, 1990.

[19] Shea, K., and I. Smith, “Applying Shape Annealing to Full-Scale Transmission Tower Re-Design”, *Proceedings of DETC99: 1999 ASME Design Engineering Technical Conferences*, September 1999, Las Vegas, NV, DETC99/DAC-8681, pp. 1-9, 1999.

[20] Shea K., and J. Cagan, “The Design of Novel Roof Trusses with Shape Annealing: Assessing the Ability of a Computational Method in Aiding Structural Designers with Varying Design Intent”, *Design Studies*, **20**(1):3-23, 1999.

Dr. Rob Bracewell
Engineering Design Centre, Department of Engineering, Cambridge University, Trumpington Street, Cambridge CB2 1PZ, UK, Phone: +44 1223 766962, Fax: +44 870 1337199, Email: rhb24@eng.cam.ac.uk, URL: http://www-edc.eng.cam.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

ESTABLISHING CASUAL LINKS BETWEEN SUCCESS CHARACTERISTICS AND PROJECT OUTCOME

C Baumgärtner and L T M Blessing

Keywords: Empirical studies, success characteristics, causality, research methods.

1 Introduction

The overall aim of this paper is to contribute to the methods used in design research. A way is proposed to guide the search for causality in empirical research, when data on a large number of variables has been captured, but the precise order in time is unknown. The method has been successfully applied in an empirical study of more than 60 design projects in the automotive industry [1].

Much has changed in design research over the last 15 years. The prescriptive research approach to support designers is increasingly being complemented by empirical approaches. Characteristics that co-occur with successful projects are identified and used to develop design support. How the causality between characteristics and successful outcome has been determined is often not clear. It seems that causality between the characteristics and success is assumed when covariance has been established. This paper focuses on finding possible causal links between project characteristics and project outcome

Research Step	Approach
1. Define success in design projects	• Literature research • Experts interviews • Reasoning
2. Classify projects in sample according to success	• Analysis of projects according to success definition
3. Identify project characteristics that show covariance with success	• Definition of dimensions of characteristics • Analysis of projects according to dimensions • Establish covariance statistically
4. Establish causality between project success and covarying characteristics	• Check covarying characteristics according to rules of causality described in this paper • Check causal links amongst covarying characteristics
5. Propose recommendation to achieve covarying characteristics	• Literature research • Expert interviews • Reasoning
6. Check recommendation in reality	• Case study

... Area discussed in this paper

Figure 1. Methodological framework

2. Underlying methodological framework

The underlying methodological framework is shown in Figure 1. The development of this framework was based on the need to determine causality. Much empirical research focuses on identifying the characteristics or elements of successful projects. In early empirical research [2], [3], [4], the characteristics of the projects (characteristics of the designers, approach of designer, general characteristics of the projects) were linked to project success based on co-variance. The authors argue that a more careful identification of causality is needed [5]. The reason is not only a scientific one, but also a practical one. The overall aim of design research is to arrive at recommendations for designers that can make the design process more successful. Hence, the reasons for a successful design process must be identified. Identifying the characteristics that covary with success is insufficient. Only the identification of the causal links between the covarying characteristics and success enables a researcher to recommend designers about the characteristics that can and should be controlled and about the most effective and efficient way of doing so. The focus of this paper is on step 4 of the methodology (for more details see [1]).

3 Causality

Causality is defined as the link between cause and effect: the cause being the reason why an effect exists. Literature on social research [6] is very careful in differentiating between causality and causal interference. Causal interference is the causal link derived from data. Causality is the theoretical concept that is independent of the data but can be identified through investigating the data. Hence, in order to establish recommendations, the causality of findings has to be understood. The cause-effect relationship in which this research is interested is between the 'outcome variable' (in design research most often success) and an 'explanatory variable' such as any characteristic of a project, for example, technical problems during the project. In this case one could recommend avoiding technical problems during the project in order to achieve successful projects. As will be shown in the following sections, such straight-forward recommendations cannot be given.

The general concept of causality is often questioned. Immanuel Kant postulated that causality is only a thinking model that allows us to collect experience instead of unrelated data, but that it is not a natural phenomena. In natural sciences, in this century, causality could not be maintained anymore at micro-level [7]: Quantum mechanics and atoms in their half-life time do not follow any causality anymore. However, at macro-level causality has its justification in order to build up experience – as Kant already stated. This is how the concept of causality is used in this research.

3.1 Prerequisites for causality

Existing literature describes principles of causality [8]. Frankfort-Nachamias [9] demands that three issues be fulfilled in order to be able to speak of true causality:

- covariation: two phenomena or variables vary together;
- time order: the cause has to occur prior to the effect;
- exclusion of spuriousness: the observed covariation cannot be explained by a third variable;

This paper describes the research process after covariation has been established (see figure 1). This means, that those characteristics have been identified that are linked to success, resulting in statements such as "projects with a high level of project goal definition and project planing are significantly more often successful" or "projects with many organisational problems are significantly less often successful". Previous work [5] proposes simple ways of establishing covariation.

Time Order

The time order between characteristics and project success is easy to identify in design research if the data is collected in a way that includes time order. In our study, success was defined as success of the project as a whole, which meant that it related to the perception of success after the project had ended. All other characteristics that were captured, described the project and were therefore prior in time to the perception of success. Hence all of these are possible causes. In order to identify causality more precisely, causality *amongst* these characteristics has to be determined. This would be relatively easy if for each single project the relative order in time of the occurrence of these characteristics could have been established. This was not the case in our data set, which was based on survey and interview data that had already been processed.

Spuriousness

In many occasions two characteristics covary and seem to be directly linked – however, the causal link occurs through a third variable. Two main mechanisms of spuriousness have been observed in our research:

- Two characteristics are output variables of the same third explanatory variable that has not been considered. We use a simple example to illustrate this mechanism. It was found that the damage after a fire covaries with the number of people in the fire brigade fighting the fire. One could conclude from the data that the damage is caused by the number of fire fighters, when the size of the fire, that causes both, has not been considered (see figure 2). The number of fire fighters is not the cause of the damage, and any measure addressing the number of fire fighters would not have the expected effect, in this case the effect could be disastrous.

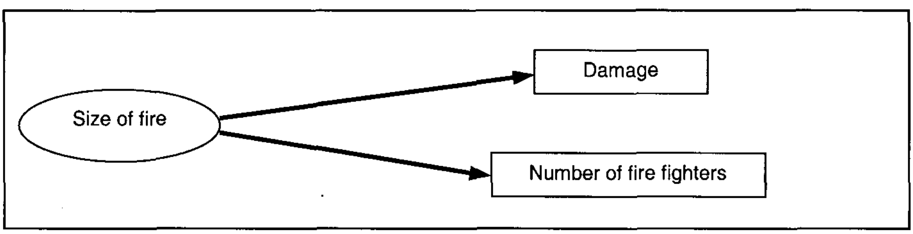


Figure 2. First example of a third variable explaining the relationship between two others

- A variable is the effect of one variable and at the same time the cause of another variable. That is, networks of variables, rather than chains of variables are possible. For example, a football team has to play with one player less, early on in the game. In the second half the team is exhausted, they are less often in control of the ball and allow a goal without scoring. Several causal links are possible. It could be each of these causes, as well as their combination that finally causes the effect (see figure 3). In developing measures to improve the situation, a focus on one variable only, may not have the desired effect.

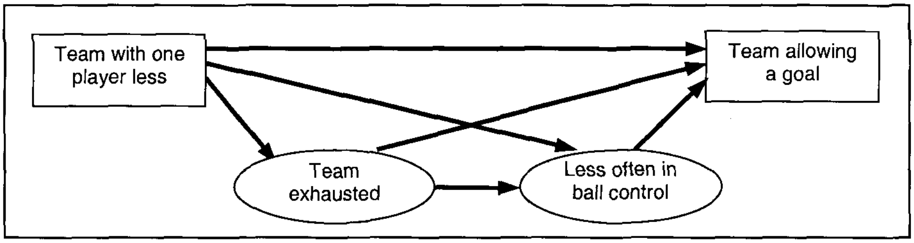


Figure 3. Second example of a third variable explaining the relationship between two others

In our study, success was found to be linked significantly to more than 20 of the variables that were measured and an initial check revealed that several of these covaried with each other. Even the time-consuming process of checking every combination of characteristics for covariance, would not have provided insight into the expected network of causes and effects that leads to success. The data lacked the time order needed to fulfil the second prerequisite for causality. This lead to the development of a method for identifying at least the possible causal links.

4 Categorisation for time order and spuriousness

In order to determine possible causality from the available data, the characteristics that were found to covary with success were first categorised in the following two ways.

First, using categories determining an approximate time order of the characteristics:

- *general characteristics* that are not linked to time nor change during the project, and hence are the earliest in time, such as the technical equipment of the company in which the project is carried out;
- *pre-project characteristics* that were determined prior to project start, such as resource allocation or design requirements from top management;
- *project characteristics* that related to the period during which the project was carried out, such as use of the original design specification as guideline throughout the project;
- *project-end characteristics* that related to the time when the project ended and afterwards, such as whether the project ended on time.

Second, using categories describing the possibility to influence the characteristics:

- *no-influence characteristics* that have to be accepted when carrying out the project, such as characteristics of a previous product;
- *trade-off characteristics* that can sometimes be influenced, but influencing these involves trade-offs, such as the time frame of the project;

- *influenceable characteristics* that can be influenced predominantly by the people carrying out the work, such as level of project goal definition and project planning;
- *result characteristics* that are the result of other characteristics and cannot be influenced directly, such as technical problems.

Table 1. Examples of characteristics that were found to covary with success, categorised according to time order and influenceability, the **bold** characteristics are discussed in the text

	No influence	Trade-Off	Influenceable	Result
General	• Design requirements from top management • Characteristics of predecessor product • High level of experience in the company in the area	• Technical equipment of the company • Time frame		
Pre-project		• Resource allocation for the design • Feasibility study carried out before kick-off	• Involvement of the project manager the project goal definition • **Level of project goal definition and project planning**	
Project		• Integration of other groups (such as purchasing or production) in the design process	• Use of original design specification as guideline during the project	• **Organisational/ technical problems during the project** • Need for change of project goals • Cancellation of the design project • Project end on time
Project-end				• Production costs of the project • Product performance • Agreement of product with design specification

The next step is to draw a cross-table (see Table 1). This table allows a clear picture of possible causal links and thereby guide an effective search for covariance between characteristics, that may reveal hitherto hidden links. For instance, in the case of the link mentioned above, "projects with many organisational problems are less often successful", the cross-table helps to recognise that organisational problems are a ´result characteristic´ occurring during the 'project' and thereby placing it relative to the other characteristics that covary with success. Search can now focus on those characteristics that covary with ´organisational problems´ and that are before this characteristic in time. One obvious option is to check whether there is covariation with ´level of project goal definition and project planning´, as it can be assumed that a low level of this characteristic might cause organisational problem. If this is indeed the case, it could be this level of project goal definition and project planning that actually explains the level of success, rather than the organisational problems, or it could be that both characteristics contribute. Different measures for improvement are required for each of these cases. Important in this respect is also the other axis in the cross-table. Those characteristics that cannot be influenced, can still be a cause but have to be excluded as the focus for improvement. Details in the original data and further identification of possible causal links are necessary to confirm the causal link as a possible explanation of success.

5 Conclusions

Descriptive design research should not only describe characteristics and their co-occurrence with successful project outcome, but attempt to explain the observed links to allow the development of effective recommendations for designers. For data in which time order cannot be establised in detail and which covers a large number of variables, a categorisation scheme has been proposed that can help identify possible causal links. In the study carried out, this method helped to focus the analysis of the many characteristics and to draw conclusions. In addition, the method helped to formulate recommendations for carrying out the type of projects under investigation [10].

References

[1] Baumgärtner, C., "Collaboration between Engineering Consultants and their Clients: Characteristics of Success", University of Cambridge: PhD thesis, 1999.

[2] Blessing, L. "A Process-Based Approach to Computer-Supported Engineering Design", Universiteit Twente, PhD thesis, 1994.

[3] Dylla, N., "Denk- und Handlungsabläufe beim Konstruieren", Technische Universität, München: PhD thesis, 1990.

[4] Fricke, G., "Konstruieren als flexibler Problemslöseprozess - Empirische Untersuchung über erfolgreiche Strategien und methodische Vorgehensweisen beim Konstruieren", Technische Hochschule Darmstadt: PhD thesis, 1993.

[5] Baumgärtner, C. and Blessing, L., "Applicability of Success Criteria in Design Research", CD-ROM Proc. International Congress of Project Engineering, Lleída, Spain, October 2000.

[6] King, G., Keohane, R.O. and Verba, S., "Designing Social Enquiry - Scientific Inference in Qualitative Research", Princeton University Press, Princeton, 1994.

[7] Zeilinger, A., "The Quantum Centennial", Nature, 408, 2000 pp. 639-641.

[8] Cook, D. T. and D. T. Campbell, "Quasi-Experimentation – Design & Analysis Issus for Field Settings", Houghton Mifflin Company, Boston, 1994.

[9] Frankfort-Nachamias, C and D. Nachamias, "Research Methods in the Social Sciences", Arnold, London, 1996.

[10] Baumgärtner, C. and Blessing L., "Characteristics of Successful Collaborations between Engineering Consultants and Clients in the Automotive Industry", Proc. ICED'99, Munich, 1999.

Dr. Christoph Baumgärtner
Bain & Company Inc., Karlsplatz 1, D-80335 München, Germany, Tel.: +49-173-399-0097, Fax: +49-89-5123-1113, email Christoph.baumgaertner@bain.com

Prof. Dr.-Ing. Lucienne Blessing
Konstruktionstechnik und Entwicklungsmethodik, Technische Universität Berlin, Sekr. H10, Straße des 17. Juni 135, D-10623 Berlin, Germany, Tel: +49-30-314-23341, Fax: +49-30-314-26481, Email: Lucienne.Blessing@ktem.tu-berlin.de, URL: http://www.ktem.tu-berlin.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

A SYSTEMATIC APPROACH TO OBSERVATIONS, ANALYSIS, AND CATEGORIZATION OF DESIGN HEURISTICS

B Bender, U Kammerer, F P Pietzcker, W Hacker, and L T M Blessing

Keywords: descriptive models of the design process, design heuristics, protocol analysis

1 Introduction

Improvement of methods and tools supporting the design process requires detailed understanding of successful design strategies. Therefore observation, analysis and interpretation of design styles and strategies within empirical design research are of high importance.

In this paper, the main objectives of empirical analyses of design procedures will be determined, potentials and limitations of different empirical approaches will be discussed and, based on results of earlier research, a systematic approach to observation, analysis and interpretation of design procedures within a laboratory environment is proposed.

This approach has been developed for and tested with a descriptive study into design strategies of novices and professional designers with the aim of investigating the applicability of classical design methods in the conceptual and early embodiment design phases. The study has been described in [1]. This paper therefore focuses on laboratory studies into design procedures, and gives an overview of the concept and the approach. Details can be found in [2].

2 Main Objectives

Some empirical studies into design are based on the hypothesis that design procedures will improve the result of the design. There has been a number of empirical studies examining this relationship, establishing a tradition of cooperation between engineering design sciences and psychology in Germany. These empirical studies focused on observation, analysis and interpretation of design procedures, aiming at the identification of successful design strategies.

Their main goals were:

- a reliable *documentation* of observed design procedures;
- reliable and valid *identification* and *classification* of types of design procedures;
- correlation of different design procedures with
  - *design success*, described in terms of *design quality*;
  - *mental characteristics* of the individual designer;

- *educational* and *professional background characteristics*;
- *external influences* (such as working environment).

Well-established methods exist for data collection and analysis of professional experience and mental characteristics (e.g. intelligence or special imagery) of designers. The same is true for the evaluation of design quality using e.g. value analysis or the related strategy described in VDI 2225. However, although there have been a number of empirical studies into design procedures, many different and few proven methods exist for valid observation, documentation, analysis and interpretation of the strategies and procedures of designers. This paper focuses on methods for the analysis of design procedures and their link with design quality. The objective is to contribute to a more systematic approach for analysing design activity.

3 Observation and Documentation of Design Strategies

To identify different design strategies one of the following approaches can be used.

1. The **hypothesis-driven** approach starts from a hypothetically deduced set of design strategies and categorizes the observed strategies according to the hypothetical set.
2. The **data-driven** approach starts from empirically identified, specific design procedures and tries to categorize these with respect to their procedural characteristics. Next the quality of the results for each group of strategies is assessed and statistically compared to find out whether significant relationships between types of strategies and results exist.

In both cases a pre-definition of *basic design operations* is needed. In addition, the hypothesis-driven approach usually includes a pre-definition of an appropriate *structure* of these operations. Observation and documentation of the process can then be carried out by recording the *transitions* between basic operations. Using suitable empirical methods, not only the *actually realised* design strategy, but also the strategies which were *intended* and anticipatively reported by the participants, and the *retrospectively reported* (*recapitulated*) strategies can be analysed. Appropriate methods for observation of actual procedures are e.g. video observation, photo observation, protocol analysis, thinking aloud techniques, computer-based protocols (an overview is given in [3], p.241). The intended and the retrospectively recognised design strategies can be examined by card-sorting-techniques, interviews, questionnaires, maps as visualisation of the design procedure, etc..

For further data-processing the visualisation using *process charts or matrices* and *transition matrices* has been established.

In process charts and matrices a postulated order of basic operations is entered along the vertical axis, and time or time intervals are entered along the horizontal axis (or vice versa). A design procedure according to a postulated type usually follows the main diagonal of the chart or matrix.

In transition matrices ([5], pp.56-58) the postulated order of basic operations is entered along both axes. The cells of the matrix contain the number of times a transition from a specific operation on the horizontal axis to one on the vertical axis has taken place. A design procedure according to a postulated type follows a line *one step above* the main diagonal.

4 Analysis and Interpretation of Design Strategies

Even though many empirical studies aim at identifying successful and less successful design strategies, further analysis and interpretation of the data based on these matrices is carried out in many different ways, reducing comparability and reliability of the results. To avoid these problems, a systematisation of the analysis procedure is proposed. In general, the following approaches for analysing process and transition matrices can be observed:

- *absolute analysis* of the matrix structure;
- *relative analysis* of distances by processing number and magnitude of transitions using descriptive statistics;
- *quantification* of deviations from postulated procedures using numerical indicators;
- *qualitative-visual analysis* of the absolute and relative matrix structure.

The results of these fundamental approaches are of different *breadth*, *depth*, *precision* and *reproducibility*. From this, one can conclude that: each method has its *specific area of applicability*, the *prerequisites* for the application of each method have to be identified and considered to ensure validity of the results, and a suitable *combination of methods* should be employed to cover the research objectives.

5 A Systematic Approach

Based on the above, a systematic approach to the planning of empirical studies into design procedures is proposed, containing expected *basic design operations*, a hypothesis-driven *determination of design strategies*, a set of *observation methods* for capturing design procedures and a set of *methods for analysis and interpretation* of design procedures. The earlier mentioned empirical study is used as an example [1].

5.1 Expected Basic Design Operations

Following established design methodologies and results of other empirical studies, a set of expected basic operations in conceptual design and early embodiment design stages can be formulated.

Table 1. Basic design operations in the conceptual and embodiment design stages (after [7], p. 140)

conceptual design	**hypothetical basic operations**
information	• identify task and requirements
definition	• abstract to identify the essential problems • establish function structures
creation	• search for working principles • combine working principles into working structures • outline principle solutions (e.g. sketches) • select suitable combinations/ principle solutions • firm up into principle solution variants
evaluation/ decision	• evaluate variants against technical and economic criteria • definition of principle solution (concept)

embodiment design	hypothetical basic operations
preliminary layout	• comprehend task and requirements • comprehend concept/ principle solution • identify embodiment-determining requirements • clarify spatial constraints • identify embodiment-determining main function carriers • develop preliminary layouts and form designs for main function carriers • select suitable preliminary layouts • develop preliminary layouts and form designs for remaining main function carriers
detailed layout	• search for solutions to auxiliary functions • develop detailed layouts and form designs for main function carriers • develop detailed layouts and form designs for the auxiliary function carriers and complete overall layout • evaluate against technical and economical criteria
completion and checks	• optimise and complete form designs • check for errors and disturbing factors • complete overall layout • definition of overall embodiment design

Not all of these operations are necessarily expected to be observed in every design process nor are these expected to be carried out in the postulated order. In literature slightly different sets of basic design operations might have been proposed (an overview can be found in [3], p.240), but generally speaking the set given in Table 1 can be considered as comprehensive and typical for those used in design studies.

5.2 A Hypothesis-driven Determination of Design Strategies According to Cognitive Psychology and Action Regulation Theory

The mental characteristics of engineering designers are usually described in terms of design thinking, or more precisely design problem solving. On the one hand, this description is helpful, because important characteristics of the involved thinking process are stressed. These are for instance:

- In contrast to generic laboratory tasks that do not require specific knowledge, design thinking deals with knowledge-rich tasks.

- Design thinking involves processing of a chain of effects: the designer has to generate spatial objects, that implement specific functions, and that, finally, will have intended effects.

- Design thinking has to cope with highly complex requirements, which often cannot be accomplished in an optimal manner: The designer ideally has to imagine all possible solutions as an exhaustive combination of their characteristics in order to select the optimal one. However, this is impossible since human mental capacity is strictly limited.

On the other hand, a description of engineering design in terms of design problem solving is doubtful. Firstly, it limits the focus to cognitive problem solving, neglecting the typically dynamic nature of design problems, even though the rather static initial conception of problem solving in Cognitive Psychology has changed much since its inception in the early 70s. Furthermore, the decisive aspect of knowledge application is overlooked, the role of mental capacity (working memory) and "cognitive cost effectiveness" [9] is not discussed, and the external kinds of processing like sketching, writing, impromptu-modelling, debating or gesturing are not considered either. Secondly, the approach of design problem solving reduces the complex and context-dependent activity of engineering design to the level of isolated mental processes: even the sum of all mental processes involved may not explain the characteristics of a successful goal-oriental engineering design activity.

Our proposal is to describe engineering design in terms of Action Regulation Theory as a goal-oriented mental working activity [6]. Following this approach, four types of design procedures can be developed as hypotheses: hierarchically phase-oriented, hierarchically object-oriented (or subproblem-oriented), opportunistic and associative.

Following proposals of methodologies such as e.g. the German VDI 2221, a procedure can be characterised as organised sequentially and hierarchically at the same time. A hierarchical organisation involves the systematic decomposition of the overall task into partial goals, that can be decomposed into smaller and smaller subgoals. These subgoals are accomplished sequentially following a predetermined plan. The process thus involves a combination of hierarchical top-down mental decomposition and sequential goal accomplishment. An overall task can be decomposed systematically in terms of the necessary design activities (e.g. clarification of the task, development of solution principles, selection of an optimal method) or in terms of systems and sub-systems to be designed. The type of decomposition determines the first two types of design procedures. The other two types follow other approaches.

1. A design activity is executed for all subsystems before the next activity takes place, which is again executed for all subsystems, and so on. This kind of procedure can be called *hierarchically phase-oriented* (following e.g.[5] p.89, see **A** in Figure 4).
2. All design activities are executed for one sub-system before they are executed for the next sub-system, and so on. This kind of procedure is called *hierarchically object-oriented* or *hierarchically sub-problem-oriented* ([5], p.89, see **B** in Figure 4).
3. The third procedure is a knowledge-driven *opportunistic and associative* procedure (see **C** in Figure 4). Instead of starting with a complete and systematic decomposition of the task, the designer may begin with a preliminary decomposition and then start elaborating a part of the system. A possible reason is that the designer remembered a suitable solution for this part and took this opportunity as a starting point. Next, the designer may improve the preliminary decomposition with the help of the solution he just realised. This procedure shows a strong similarity to the TEA (task-episode-accumulation) model ([8]). This is a mixture of a top-down, more or less systematic decomposition with an opportunistic, local and bottom-up, proceeding. It is less guided by a total goal but more by the association between the step that has just been finished and the opportunity to do the *same* step again on a *different* (sub-)problem. Since engineering design normally consists of knowledge-driven and reasoning-driven steps, it seems likely that this procedure can be identified.
4. From the point of view of action regulation, a trial and error-like *muddling through* procedure may be identified (see **D** in Figure 4), which involves a more or less unsystematic trying to cope with different parts of the system and different design activities.

5.3 Observation Methods for Capturing Design Procedures

For the observation of design procedures of a large number of participants (N=81), with up to 35 participants in one session, working parallel on individual tasks, we used photo-documentation, the time-dependent use of differently coloured pencils and self-protocols.

Data on the intended and the recapitulated procedures was collected by means of questionnaires and card-sorting. The data was transferred to process matrices (Figure 1) by a team of three design experts. In this way, not only changes in basic design operations could be identified but also potentially associated changes between sub-systems (e.g. between a gear and a clutch, visualised by different hatching in Figure 1). For observation in the currently running second phase of the study (single test-design with only one participant in each test) the method of participant observation is used, supported by a software tool for real-time data-capture. The software tool automatically converts the data into process charts.

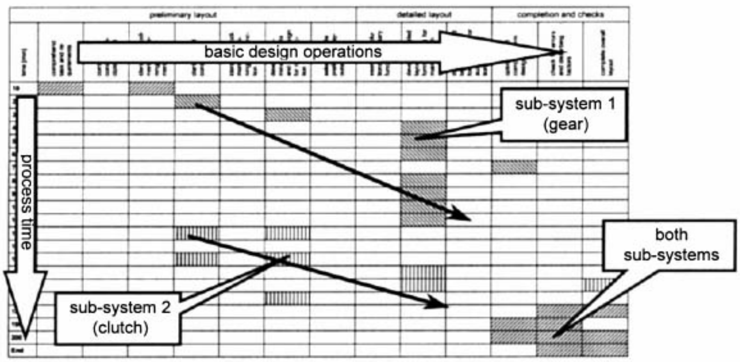


Figure 1. Example of a process matrix

For further processing, the process matrices were transferred into transition matrices. These allow the joint visualisation of *number* and *size* of observed transitions between two design operations (see the example C4(row)⇒D2(column) in Figure 2). Transition matrices are of particular advantage for visualisation of long processes because, in contrast to process matrixes, there is no expending time-axis.

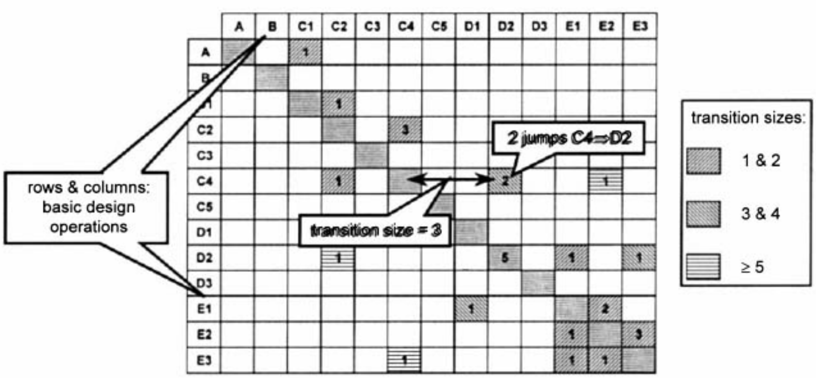


Figure 2. Example of a Transition matrix (set up after [5])

These methods give very good first advice for the interpretation of the recorded data and are easy to support by software. However, for the valid identification of different design procedures according to the mentioned hypotheses, further analysis of the data is needed.

5.4 Methods for Analysis and Interpretation of Design Procedures

For further processing of process and transition matrices, first a frequency analysis based on the observed transition sizes was carried out. The number of 'hits' within cells representing a predefined class of transition-sizes (visualised as cells with equal hatching in Figure 2) were summed and plotted in a bar chart. This provided a very good first indication of different design procedures (compare e.g. participant 61 to 85 in Figure 3).

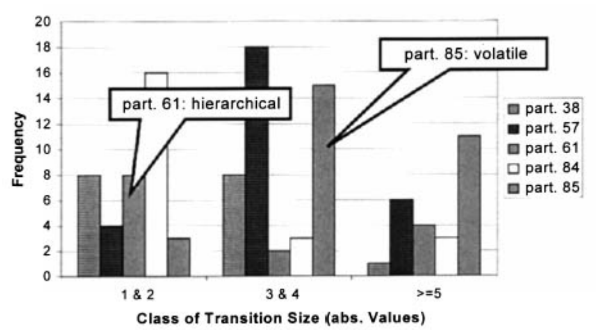


Figure 3. Example of a bar chart visualising the different frequencies of transition-sizes of 5 participants

To validate the hypothetically determined design procedures, a distinction was made between activity transitions that, at the same time, involved a sub-system transition, and activity transitions that did not involve a sub-system transition. The mean values of the transition sizes in both cases were determined and entered into a portfolio diagram (Figure 4) along different axes. This allowed a valid distinction of the procedures in the indicated sectors (A-D).

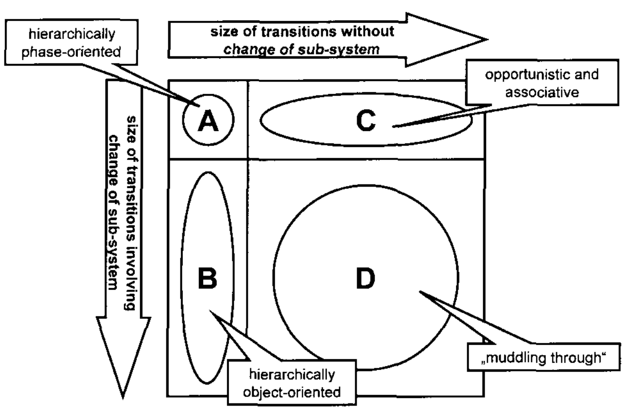


Figure 4. Portfolio-diagram identifying four different design procedures

6 Conclusion

This paper is intended as a contribution to the systematisation of empirical design research. Its main focus is on comparability and reproducibility to enable validation of methods and results. Aims and objectives of empirical studies into design procedures, methods for data capture, analysis and interpretation were discussed. As a result an approach for systematic conceptualisation of empirical studies into design procedures is proposed (documented in detail in [2]) which is to be considered as an element of a "general design research methodology"[4].

References

[1] Bender, Bernd, Kammerer, U., Pietzcker, F., Hacker, W., Blessing, L., Successful Strategies for Complex Design in Early Phases of the Product Development Process - an Empirical Study, International Conference on Engineering Design ICED 01, Glasgow 2001

[2] Bender, Bernd, Pietzcker, F., A Guideline for Observation, Analysis and Interpretation of Design Procedures and Design quality in Empirical Laboratory Studies, TUB-KTEM/ Technical Report 01-2001, Technical University of Berlin 2001

[3] Blessing, L.T.M., A Process Based Approach to Computer-Supported Engineering Design, Cambridge 1994

[4] Blessing, L.T.M., Chakrabarti, A. and Wallace, K.M.: An Overview of Descriptive Studies in Relation to a General Design Research Methodology, in: Frankenberger, E. et. al. (Eds): Designers The Key to Successful Product Development, pp.42-56, London 1998

[5] Fricke, G., Konstruieren als flexibler Problemlöseprozess – Empirische Untersuchung über erfolgreiche Strategien und methodische Vorgehensweisen beim Konstruieren, VDI-Fortschrittsberichte, Reihe 1, Nr. 227, Düsseldorf 1993

[6] Hacker, W.: Konstruktives Entwickeln als Tätigkeit – Versuch einer Reinterpretation des Entwurfsdenkens (design problem solving), Zeitschrift für Sprache & Kognition 18 (3/4), pp. 88-97, Bern 1999

[7] Pahl, G. and Beitz, W.: Engineering Design – A systematic Approach, 2Rev.Ed., London 1996

[8] Ullman, D.G., Dietterich, T.G. and Stauffer, L.A.: A model of the mechanical design process based on empirical data, Artificial Intelligence in Engineering Design and Manufacturing 2, pp. 33 – 52, 1988

[9] Visser, W., Organisation of Design Activities: Opportunistic, with hierarchical episodes, Interacting with Computers 6, pp. 239-274, Oxford 1994

Dipl.-Ing. Bernd Bender, Dipl.-Ing. Uwe Kammerer, Prof. Dr. Ir. Lucienne Blessing
Technical University of Berlin, Dept. of Mechanical Engineering and Transport Systems, Engineering Design and Methodology, Sekr. H 10, Strasse des 17. Juni 135, D-10623 Berlin, Germany, Tel.: ++4930 314-24309, Fax: ++4930 314-26481, E-mail: bernd.bender@ktem.tu-berlin.de

Dipl.-Psych. Frank Pietzcker, Prof. Dr. rer. nat. habil. Winfried Hacker
Technische Universität Dresden, Fakultät Mathematik und Naturwissenschaften, Allgemeine Psychologie, D-01062 Dresden, Tel: ++49351 463-6890, Fax: ++49351 463-3522, E-Mail: pietzcker@psy1.psych.tu-dresden.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DESIGN PROJECT MANAGEMENT IN PRACTICE – THE E-MAIL DIARY STUDY

D J H C Hendriks and K Dorst

Keywords: Interviews, questionnaires, project reports.

1 Introduction

The increasing complexity of the products and systems we design has led to increasingly complex design projects. As a result, design project management is rapidly developing into an independent field of expertise and even into a separate discipline [1]. Most of the people currently working in design project management are designers, who bring their design experience to bear upon the new, complicated design projects. Much of this knowledge is implicit in nature. In our research project we try to elicit this implicit design project management knowledge, and make it more explicit. These explicit heuristics can then form the basis for critical reflection, both upon the rules themselves, and by comparing them to the rules found in project management and design methodology literature. A clear overview of design project management heuristics can serve as the basis for training a new generation of design project managers.

Previous research indicated that implicit knowledge is an important basis for design project managers in managing their projects [2, 3]. Also Reid et al. [4] state: '... *Although project managers repeatedly refer to their expertise and knowledge, much of this knowledge is undocumented, tacit knowledge, which often is acquired by word-of-mouth, with people finding out who has relevant information on an informal basis, rather than by any formal process of identification. ...*'.

Our previous research project was based on retrospective interviews with experienced project managers working in design agencies. This research project resulted in an overview of problematic situations that can occur in managing design engineering projects. In our current research project we aim to describe the problematic situations in project management in much more detail. We aim to reveal implicit project management knowledge by eliciting the heuristics that design project managers use in decision making and problem solving.

In this paper we will first describe this research method in detail (§2), and then we will provide a first overview of the data we collected from eight design project managers working in well-known design agencies (§3).

2 Description of the e-mail diary research project

The definition of heuristic we use is based upon Snoek [5]: a heuristic is a goal-oriented way of thinking (rule of thumb) that is used to solve a problem of which the solution is already suspected.

In our research project we will use the following format:

Heuristic: IF … (problematic situation), THEN … (decision / solution) + conditions. An Example from daily life: IF I want to cross the street, THEN I look if there are any approaching cars. Condition: unless the traffic light is green.

We want to elicit these design project management heuristics by analysing the reports of personal project management experiences. To obtain a detailed description of design project management in practice, it is preferable to follow the design project managers while they are managing the projects instead of interviewing them afterwards. In our first research project we asked the project manager to keep one project in mind and to explain experiences in this project. The weakness of this technique is that there might be bias and incompleteness in the recollection of the project managers. In our current research project we would like to follow more projects real time. To do this we combined two research techniques in our "e-mail diary study": interviews and questionnaires. During a three-month period we closely followed project managers in managing their projects by means of weekly e-mail project diaries. Before and after this period we interviewed them. For an overview of the research approach see figure 1.

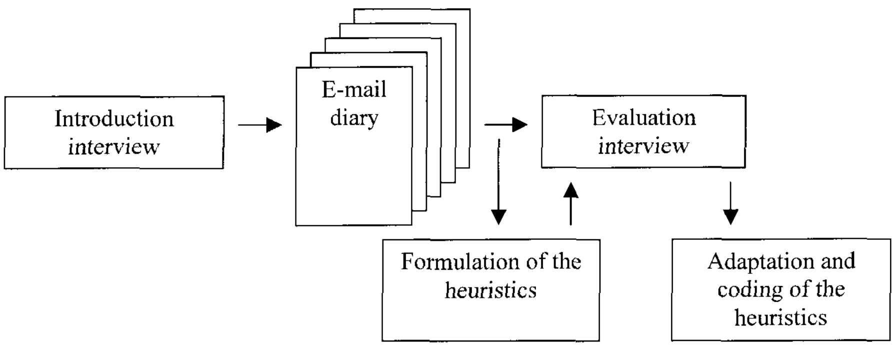


Figure 1. The "e-mail diary study" approach for one participant

Participants: Eight design project managers, working in large and well-known Dutch design agencies, participated in this research project. During the time of our study they managed a total of 41 design projects. They had between 3.5 - 12 years of design engineering experience, and 1 - 10 years of project management experience. The design projects we studied dealt with a variety of subjects: consumer products, industrial machinery, technical devices and web pages.

Introduction interview: In interviewing the project managers before the e-mail diary period, we aimed to explain our research project, to give instructions for filling in the e-mail diaries, and to obtain general information about the project manager and his projects.

Weekly e-mail diaries: After interviewing the project managers, they received weekly e-mail diaries during a period of three months. To encourage project managers to individually report their projects in order to reveal their experiences in managing projects, we developed a short questionnaire. This questionnaire consisted of a project diary for each project they handled at that moment. We asked them what problematic situations occurred that week and how they solved them.

Formulating heuristics: After receiving the diaries we transferred the texts into a database to get an overview for each project over the total research period. The interview texts and diary texts were segmented and out of these text segments heuristics were formulated.

Evaluation interviews: The evaluation interviews at the end of the e-mail diary period were held to obtain additional information and reactions of the project manager on the elicited heuristics. We completed these interviews with some questions about the research method, in order to evaluate this method from the perspective of the 'victim'. The heuristics were adapted or deleted in case the project manager didn't agree with them.

Coding of the heuristics: The final heuristics, 436 in total, were transferred into a database. For the coding of the heuristics we used a project management model that combines project management aspects we found in theory [e.g. 6, 7] and our previous research [2, 3].

Four codes were assigned to each heuristic. The first code represents a **phase** of the project, the second code represents a co-operation party, the third and fourth code represent a **controlling** aspect for the IF and the THEN part of the heuristic. For an overview of the coding categories see figure 2.

Phasing	Co-operation parties	Controlling aspects
1. Definition	1. Project manager	1. Time
2. Conceptualisation	2. Team	2. Money
3. Detailing	3. Design agency	3. Quality
4. Realisation	4. Client	4. Information
5. Evaluation	5. Supplier(s)	5. Organisation

Figure 2. Overview of the coding categories

To illustrate the way the heuristics were coded, let us look at heuristic 132 of project manager 2.

IF the client has many problems to get a model working, THEN we indicate that we, the design agency, can assist him.

The code is: 3435, this means that this heuristic was applied in the *detailing phase* (phasing: code 3) to solve a problematic situation that deals with the *client* (co-operation party: code 4). This problematic situation (IF-part of the heuristic) has to do with the *quality* of the product (first controlling aspect: code 3). The solution (THEN-part of the heuristic) has to do with the *organisation* of the project (second controlling aspect: code 5).

3 Heuristics in design engineering project management

The coding of the heuristics enables us to create an overview of the problems these design project leaders reported, and of the solutions they used. We will here present only some of the most basic overviews, to give the reader an idea of the data. We will here present the heuristics we found by *phase* of the design engineering project (definition, conceptualisation, detailing, realisation and evaluation phase), by *co-operation parties* (the project manager himself, his team, his design agency, the client and supplier(s)) and by the *controlling aspects* (time, money, quality, organisation and information).

3.1 phasing

It is important to study the project manager during an extended period and in his natural working environment. In our research project we studied 41 projects managed by eight different project managers during the different phases of their projects, so every design project management phase is represented in our data.

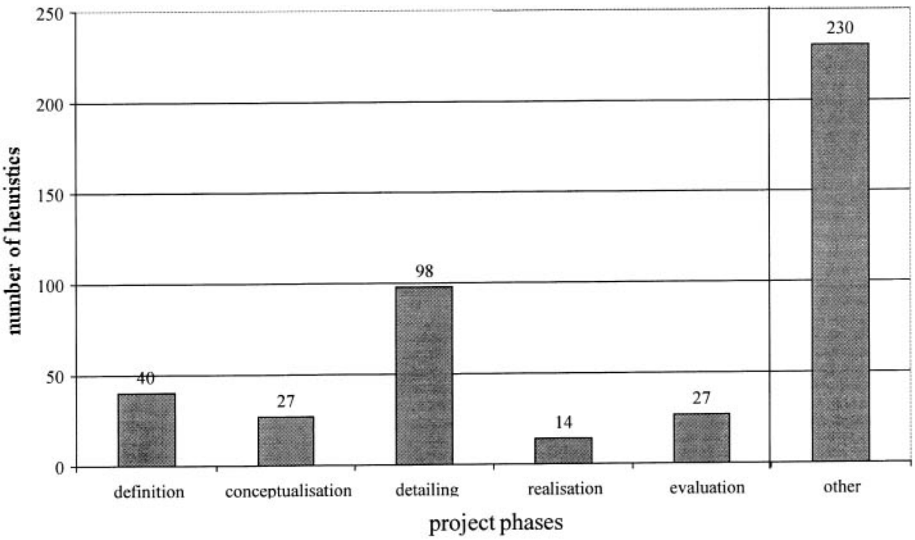


Figure 3. Phasing the project (n=436)

Figure 3 shows how the heuristics are distributed over the different project phases. Below some examples of the heuristics are given.

- **Definition phase (9%):** Example: IF any promises related to the budget are made by the client, THEN I already start cautiously although I know that I need official approval. (heuristic 58).

- **Conceptualisation phase (6%):** Example: IF one design proposal is much better than the others, THEN I consider presenting only that proposal. (heuristic 71).

- **Detailing phase (22%):** Example: IF there has been no time to examine all the technical risks, THEN it is wise to detail the model and only then start producing a series. (heuristic 52).

- **Realisation phase (3%):** Example: IF I feel that the client knows that manufacturing the parts is risky, but that he does not worry about it initially, THEN you have to convince the client that it is impossible within the given time. (heuristic 394).

- **Evaluation phase (6%):** Example: IF I want to prevent that the project will go on after sending the definitive design, THEN I announce a meeting to evaluate matters. (heuristic 294).

Only 3% of the heuristics was used during the *realisation phase*. This is not much, because the project team barely has a role in realising the project. But the problematic situations that do occur in this phase can be very serious, as we found in the diaries. Only 6% of the heuristics concerned the *evaluation phase*. This could be because evaluation is often skipped, through lack of time. Almost 53% of the heuristics can not be coded using the distinct phases. Many of these heuristics represent situations can not be allocated to a specific phase.

3.2 Co-operation parties

During a project the design project manager has to co-operate with the project team, the industrial design agency, the client and the supplier(s). Also the project manager himself has qualities which need to be managed. Figure 4 shows the heuristics distributed over the different co-operation parties. Again below some examples of heuristics are given.

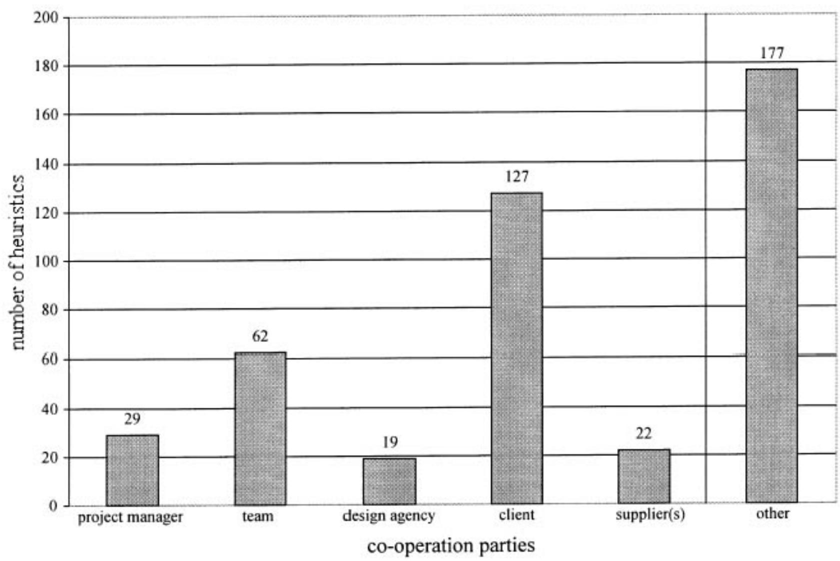


Figure 4. Co-operation parties (n=436)

- **Project manager (7%):** Example: IF I have to write a report and I need a deadline for it, THEN I plan some meetings and tell all parties when they will receive this report (heuristic 9).

- **Team (14%):** Example: IF a team member is on holiday, THEN I have to talk with him about the time he is available (heuristic 13).

- **Design agency (4%):** Example: IF a new project is valuable for our agency, THEN I will take overrunning of the budget for granted to obtain a better quality. (heuristic 267).

- **Client (29%):** Example: IF the client adds some new requirements which would lead to more design work, THEN I send the client an additional offer. (heuristic 452).

- **Supplier(s) (5%):** Example: IF a supplier says that he will overrun his planned time, THEN I have to call him and exert some pressure. (heuristic 446).

Many heuristics (29%) concern the co-operation with the *client*. Although this percentage is not completely surprisingly, literature and education do not give much attention to this subject. Almost 41% of the heuristics can not be classified into our coding categories. When we have a close look at these heuristics we see that the *content of the project* turns out to be an important issue of influence in design projects. Apparently the design task itself is also an important issue of influence on the decisions made by the project manager. Literature on project management advises against this involvement in the *content of a project* [7], as this can cause dilemmas and contradictions. We will search for some new categories within this group of heuristics.

3.3 Controlling aspects

An overview of the heuristics distributed over the controlling aspect is given in figure 5 controlling aspects. Again some examples of heuristics are described below.

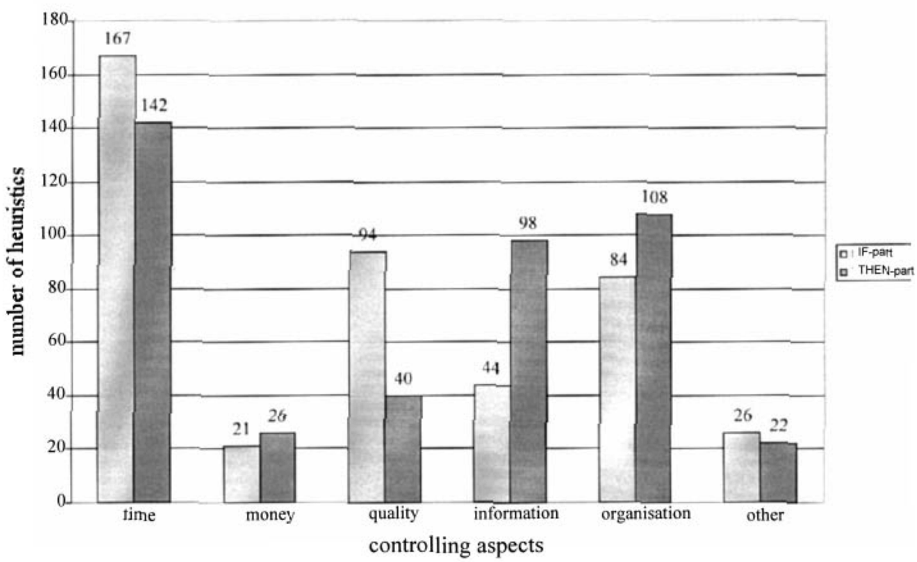


Figure 5. controlling aspects (n = 436)

- **Problem: quality, solution/decision: quality:** Example: IF it is not possible to solve a design problem another way because the mould has already been made, THEN you have to adapt the mould. (heuristic 441).

- **Problem: organisation, solution/decision: organisation:** Example: IF the budget is finished and more manpower is necessary to complete the drawings on time, THEN we present the reasons to the client and ask if we can charge for this. (heuristic 124).

- **Problem: information, solution/decision: organisation:** Example: IF I want to bring into line everybody's view of the project, THEN I attend the progress meeting with the client and the supplier. (heuristic 345).

It is not surprising that *time* causes many problematic situations (39% of the heuristics) and that *time* is also used often as solution or decision (34%). Apparently the *money* (which, with 5%, does not seem to cause many problematic situations) is seen as invariable, and *time* seems to be an easy solution. A project manager can use more time or more people. *Quality* is the controlling aspect that causes 22% of the problematic situations, but this controlling aspect is not often used as a solution strategy (only 9%). Obviously *quality* is an important aspect, but apparently a project manager does not make many concessions compromising the quality of the design. *Information* is a problem in 10% of the heuristics. It is more often used as a solution (22%). We assume that informing the client can sometimes alleviate the concerns of the client.

4 Conclusions

4.1 On the e-mail diary research method

This research method has proven to be very useful for studying participants during an extended period of time. As we had contact with these project managers by sending them the project diaries by e-mail, no time consuming visits were necessary, except for the interviews before and after the e-mail diary period. We gathered an enormous amount of data. One diary consists of the report of the experiences of one project in one week. We received 328 diaries that were the basis for the formulation of 436 heuristics.

Besides these advantages for the researchers, this research method also provides advantages for the project managers. They can decide when they have the time to fill in the diaries and they can use as much time as they need. Some project managers emphasised the usefulness of these diaries for their own reflection of moments.

We admit that there are also some matters of attention. The researcher has little control of receiving diaries. When a project manager does not send back his diaries or when diaries come in late, all the researcher can do is send a reminder. This makes for a very complicated and time-consuming effort.

4.2 On design project management

The standard project management literature does not address many of the problematic situations we found in our study. Design Methodology almost exclusively studies the design project for individual designers [8] and project management literature is basically aimed at the controlling of large projects in civil engineering and architecture. Project managers in the types of projects we studied complain about the project management literature being too general to be useful and practical in their specific discipline.

In coding the heuristics we had difficulties in classifying heuristics in the different phases and co-operation parties. Obviously not al the problematic situations can be classified in phases, and there can also be another co-operation party beside the parties we found in literature and previous research. It will be very interesting to take a closer look at these two groups of heuristics in order to complete the design project management model. As we already mentioned before, we assume that the majority of the heuristics in the group *other* in the co-operation diagram concerns the content of the project and that therefore many of the general assumptions and general rules for project management indeed do not apply to product design. For instance, design project managers have to manage the content of the design project, as well as the process (see also [9]). This has far-reaching consequences for the new courses in design project management that are currently being set up. Besides having a closer look at the *other* groups, we also will study the controlling strategies (the links between the controlling aspects of the IF and the THEN parts of the heuristics). For example: are *time* problems often solved with *time* solutions? Or does a design project manager use other controlling aspects?

References

[1] Buijs J. and Valkenburg, R., "Integrated New Product Development", Lemma, Utrecht, 2001.

[2] Hendriks, D. and De Wilde, J., "Project Management for New Product Development projects: an empirical study" Lindemann U, Birkhofer H, Meerkamm H, Vajna S (Eds) Proceedings of the International Conference on Engineering Design, Heurista, Zürich, 1999.

[3] Hendriks, D. and De Wilde, J., "Design Project Management", Scrivener, S. A. R., Ball, L. J. and Woodcock, A. (eds.), CoDesigning 2000 proceedings, Coventry, Coventry University, 2000, p. 67-72.

[4] Reid, F. J. M., Culverhouse, P. V. Jagodzinski, A. P., Parsons, R. and Burningham, C. "The management of electronics engineering design teams: linking tactics to changing conditions", Design Studies, 21, 2000, p75-97.

[5] Snoek, J. W., "Het denken van de neuroloog", Dijkhuizen van Zanten, Groningen, 1989.

[6] Roozenburg, N. F. M., and Eekels, J., "Product design: fundamentals and methods", John Wiley & Sons Ltd, West Sussex, 1995.

[7] Wijnen, G., Renes, W. and Storm, P., "Projectmatig werken", Marka, Utrecht, 1997.

[8] Dorst C. H., "Describing Design - a Comparison of Paradigms", PHD Thesis, Delft University of Technology, 1997.

[9] Valkenburg, R., "The Reflective Practice in product design teams", PHD Thesis, Delft University of Technology, 2000.

Danielle Hendriks and Kees Dorst
Faculty of Design, Construction and Production, Department of Product Innovation and Management, Delft University of Technology, Jaffalaan 9, 2628 BX Delft, The Netherlands, d.hendriks@io.tudelft.nl, phone: +31-15-2781403/3068, Fax: +31-15-2787662

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A NOVEL MODEL OF LEARNING IN DESIGN

S K Sim and A H B Duffy

Keywords: Design re-use, design learning loop, pragmatic scientism

1 Introduction

Learning in design is a phenomenon that has been observed in design practice by many researchers. The observation that designers learn is supported by protocol studies in design that experienced designers can reach satisfactory design solutions more effectively than novice/naive designers. That there was no comprehensive model or theory of learning in design to explain the phenomenon was identified by Sim [1]. Hence a need was raised to develop a comprehensive model of learning in design that can describe the phenomenon and therefore serve as a basis to develop effective and efficient design support system(s) [1].

2 Criteria for evaluating models of learning in design

Sim [1] posited a set of criteria to evaluate existing models of learning in design to establish their capability of describing the phenomenon.

Explanative capability: Explanations are necessarily abstract statements that identify general properties, characteristics, and underlying processes and mechanisms for all possible instances of a phenomena [2]. They can be either based on evidence or based on a set of deductions derived from the tenets of the theory or models. Hence, a model or theory is said to have explanative capability if it is able to explain every observable instance of the phenomenon through its systems of abstract statements.

Predictability: For models or theories to exhibit predictability, they are characterised by a consistent set of statements and an inferential structure that can be used for prediction of events or conditions on the basis of assumed existing conditions [3].

Operationalisation: To construct effective explanations, all the terms and concepts used to form a theory must be operationalised [2]. These operationalisations must make it possible, in practice, to identify and classify, unambiguously, particular examples and states of the phenomena covered by the theory. Before operationalisation can occur, all terms and concepts must be subjected to an operational definition. An operational definition of a term is one which defines the term in connection to some physical or computational operation used to measure or quantify the defined entity.

Comprehensiveness: Miller [4] observes that the goal of science is the development of both special theories of limited scope, middle-range theories concerned with a greater number of phenomena and general theories that unify or integrate special and middle-range theories that include a major segment of the total subject-matter of a field or of several fields. Hence, in evaluating the various theories/models that describe the cognitive behaviour of designers, a design theory will be considered as a special theory since it describes a single perspective of the cognitive behaviour of the designer; a theory of learning in design would be considered as a middle-range theory and that which describes the social cumulative cognitive process involving a team of designers would be considered as a general theory.

Applicability: The measure of applicability of a theory or model of design is the extent to which it has been applied to design of different artefacts or different types of design processes. It can be described as artefact independent or domain independent.

3 Review of models of learning in design

3.1 Existing models of learning in design

While there are numerous models or theories of design, only three models of learning in design have been identified. These are described below.

Exploration-based model: The exploration-based model of design [5] characterises design as an exploratory process in which there is a co-discovery of the nature of the design problem and its solution (see Figure 1). The co-discovery process explores the problem formulation in terms of a set of expected behaviours and the production of a set of technical specifications that will deliver the functionality described in the final requirement description (i.e. the design solution) through a collection of inter-related design activities. Underlying the exploration process is the knowledge change that occurs as the design solution evolves, the chronicles of which provides the knowledge of the design history.

The input knowledge is:

- *Initial design requirement* R_i: incomplete and inconsistent statement of the design requirement.
- *Design Knowledge Base* (DKB) which comprises of:
  Domain knowledge K_{dm}: knowledge of the possible design space to be explored.
  Design knowledge K_{dn}: procedural knowledge (e.g. design strategies, methods) about how the space can be explored.

The knowledge generated is the *Design Description Document* (DDD) that comprises of:

- *Final design requirement* R_f: a complete and consistent requirement description derived through the process of exploring the space of possible designs (SPD).
- *Final design specification*: a functional and structural description of the design solution.
- *Design exploration history* H_d: A record of knowledge acquired in the activities performed, the decisions made, and the rational behind each decision.

For the knowledge generated in DDD to be available for re-use during other design tasks, the DDD needs to be transferred to the DKB, built to support the class of problems concerned.

The relationship between the DKB and the DDDs resulting from a series of design tasks is one of abstraction and generalisation and is an example of post-design knowledge change.

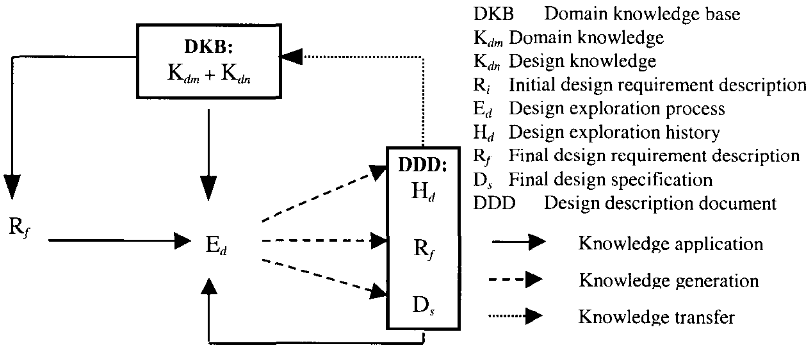


Figure 1. An exploration-based model of design [5]

Design Re-use Model: Duffy et al. [6] formulated the first engineering *Design Re-use Model* as illustrated in Figure 2. The knowledge components of the processes are:

- *Design requirements*: refers to a statement of a design need/desire.
- *Domain knowledge*: knowledge pertaining to a particular domain.
- *Re-use library*: a storage location holding reusable (compiled) design knowledge, information and data.
- *Domain model*: represents a designer's conceptualisation of a design domain that is applicable to the current design problem.
- *Evolved design model*: a statement of an evolved design.
- *Completed design model*: a statement of a fully designed new artefact that is believed to satisfy the design requirements.
- *Design by Re-use*: the process in which knowledge resources are searched and useful knowledge can be identified, retrieved, and applied to the new design.
- *Domain Exploration*: the process of searching, understanding, generalising, and in general, "rationalising" the domain to gain an understanding of the features of that domain from which reusable fragments of knowledge can be identified, extracted, and stored for subsequent use in design.
- *Design for Re-use*: a process carried out to generate designs for subsequent re-use.

In this model, the knowledge learnt comprised of the completed design model, the domain model and the re-use library. Through the *Domain Exploration* process, the completed design model together with the domain knowledge is transformed into reusable fragments (chunks) of knowledge for subsequent use in design. Through the *Design for Re-use* process, knowledge generated in the evolved design model is acquired for subsequent re-use. In both these processes there is an increase in knowledge of the design. The design re-use model has been implemented in two systems, NODES [7] and PERSPECT [8] to demonstrate the learning of knowledge and subsequent application of that knowledge in design.

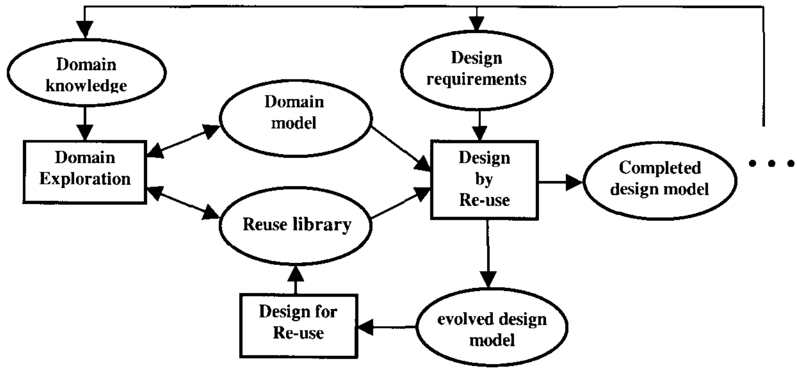


Figure 2: A design re-use model (Duffy et. al. [6])

The Design/Learning Loop: Duffy and Duffy [8] postulate how the activities of design and learning are coupled in Figure 3. The lower loop suggests in-situ learning and application of knowledge that occurs as the design solution is evolved from an initial stage, Stage: 1 to a design solution specification, Stage: N. In-situ learning occurs when knowledge learnt from the solution and its development is fed back to some store of experiential knowledge. Application of knowledge occurs when this knowledge is re-used to aid in the evolution to an acceptable design solution. Some of the learned knowledge will transform to long-term experiential knowledge to be re-used in later design scenarios whereas some of the transient knowledge learned will be used to assist in the evolution of the design. In addition to the lower loop, there is another loop that represents updates or modifications of the experiential knowledge depicting the designers' ability to explore and learn from their own knowledge.

Based on the Design/Learning Loop model, Duffy and Duffy offer a concept of computational learning that can ease the burden of knowledge elicitation and yet maintain (i.e. update and evolve) experiential knowledge depending on the knowledge needs of the designer. Hence, they introduce the concept of Shared Learning in which the designer and a computing system as a learning assistant co-learn when a designer learns new knowledge as a result of a learning assistant automatically learning and presenting previously implicit, and therefore unrepresented knowledge. Shared Learning is implemented in the system PERSPECT [8].

3.2 Evaluation of the models of learning in design

Explanative capability: The exploration-based model explains the design process as exploration of the space of possible designs in which an initial incomplete and inconsistent design requirement is modified into a final design specification. As a result of the exploration process, the knowledge gained in terms of what parts of the design space were explored, how and why decisions were made is recorded in the design history. The capturing of the design history, the final requirement description, and the associated design specification provides means by which knowledge can be learned retrospectively [1]. The design re-use model explains how knowledge that is generated during design can be stored for subsequent re-use. The recorded knowledge can be searched, abstracted into re-usable fragments and stored for subsequent use. It therefore explains an aspect of the phenomenon of learning in design in which knowledge is learned retrospectively. In addition to learning knowledge retrospectively, the Design/Learning loop model is able to explain how design and learning can be coupled (i.e. in-situ learning during the design process).

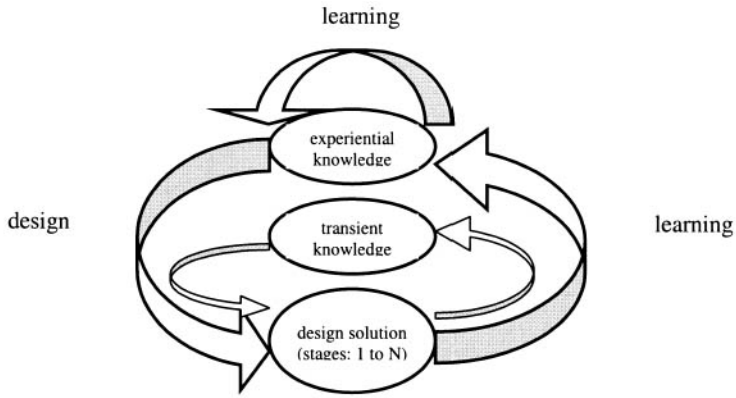


Figure 3: Design/Learning loop (Duffy & Duffy [8])

Prediction: Smithers et al. [9] observe that the exploration model lacks a formal description for it to be more predictive. The design re-use model does not feature any predictive capability in predicting when should learning take place or, given a design activity, what aspect of knowledge is relevant. The Design/Learning loop does not provide a formalism by which to predict different types of knowledge learned and how they are learned for different types of design activities. However, in the implementation of PERSPECT, the system is able to predict and therefore provide implicit knowledge available in explicit past design cases through generalisation.

Operationalisation: Due to the nature of exploration, Smithers et al. [9] expect the operationalisation of the model to be domain dependent in terms of the exploration activities. But, in assessing the exploration model they determined that it lacks the use of knowledge in the control of the design process. Two systems, NODES and PERSPECT were implemented to show how the re-use model can be operationalised. The Design/Learning loop does not define a definite set of operators by which the users of the model can identify with for the implementation of the model. Nevertheless PERSPECT was implemented to illustrate some aspects of learning in design.

Comprehensiveness: Although the exploration model is meant to describe the design process, it has elements that support retrospective learning for design re-use through abstraction and generalisation of knowledge from similar DDDs. It therefore has a limited mechanism for learning. The re-use model is an example of a specialised model of learning in design as its architecture is aimed at learning abstract or implicit knowledge from past designs. The Design/Loop covers the phenomenon in terms of in-situ and retrospective learning. However, it is not a general model of learning in design as it does not explain how knowledge is learnt in the context of a typical repertoire of the design activities encountered in design practice.

Applicability: The model of design exploration was evaluated within the domains of preliminary design of water turbine [5]. Although the re-use model has been applied to preliminary design (e.g. NODES and PERSPECT), it is conceivable that it can be used in other design scenarios (e.g. design for manufacture, etc.) in which knowledge needs for different perspectives of the design cannot be harnessed directly from past designs. The

Design/Leaning loop can be illustrated in the concept of Shared Learning in which the designer learns as the computational learning system learns [9].

3.3 Summary of evaluation

The evaluation of the three models is summarised in Table 1 below and suggests that:

- There was no general theory or model to describe the phenomenon of learning in design in a comprehensive manner.
- The existing models evaluated present some aspects of learning in design.
- For operationalisation to be successful, leading to the building of systems, there is a need for a clearly defined set of operators for design and learning.

Table 1: Evaluation of models of design and/or learning

	Models		
Criteria	*Exploration-based*	*Design re-use*	*Design/Learning loop*
Explanative capability	Some	Some	Some
Predictibility	Poor	Poor	Limited
Operationalisation	A limited set of design operators	A limited set of design/learning operators	A limited set of design/learning operators
Comprehensive-ness	Retrospective learning	Retrospective learning	Retrospective/In-situ learning
Applicability	Few	Few	Few

To explain the phenomenon of learning in design more comprehensively, the model ought to feature the following characteristics:

- A formalism that represents the repertoire of typical design activities in which knowledge change that occurs can be explained in terms of the input knowledge, output knowledge and the goal of the design activity.
- A formalism that depicts the learning activity in terms of elements which explain what design knowledge is learnt, why learning occurs, how learning takes place, when is learning triggered and in what manner these elements interact [10].
- The nature of interactions that occurs between the design and learning activities (i.e. to postulate hypotheses on the nature of these links and substantiate them by evaluation).

4 LinD - A model of learning in design

Developing a model or theory of learning in design requires the knowledge of two human cognitive activities, that of design and learning and their interactions. The traditional research methodology in deriving a model or theory that is objective, universal and context independent is that of scientism. But the research methodology of scientism requires that results from two or more experiments under similarly controlled conditions to be the same. In the context of evaluating the model of learning in design, the dependence on human designers to be the subjects of experimentation in evaluating the model would pose a stumbling block in achieving unequivocally the same results in two nominally identical experiments. This is because no two designers are alike in their design practice. Hence, although the principal research methodology adopted here is that of scientism, the method of evaluation is not through experimentation as is the norm for traditional scientific method, but through the protocol analysis of a case study of a designer at work [11]. Through the research methodology adopted, a model of learning in design called LinD has been developed.

LinD features formalisms for describing the cognitive activities of design and learning and three links that explain the interactions between these activities; epistemic, teleological and temporal link (see Figure 4). A mapping between a design activity and knowledge transformer ($A_d :: K_t$) shows the epistemic link between the design and learning activities.

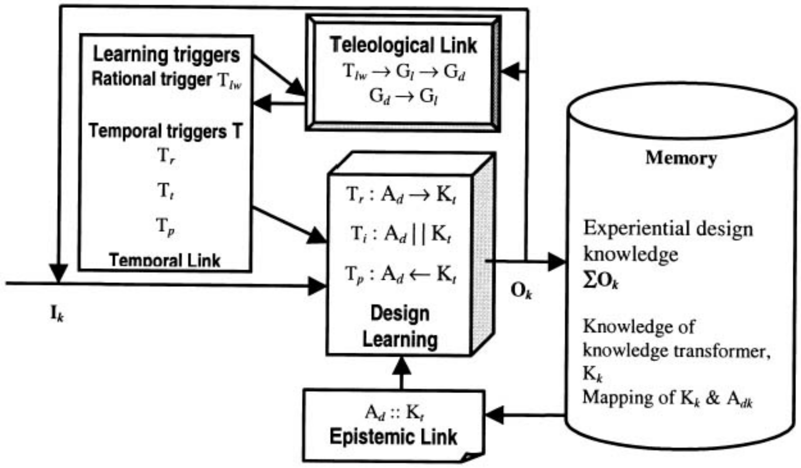


Figure 4 A model of learning in design, LinD [1]

Depending on the nature of the design activity, there are two possible ways by which design and learning goals interact that depict the inextricable link at the teleological level. Firstly, the learning goal (which is triggered by the rationale learning trigger, T_{lw}) precedes the design goal (i.e. $T_{lw,} \rightarrow G_l \rightarrow G_d$). Secondly, the design goal precedes the learning goal (i.e. $G_d \rightarrow G_l$). Depending on the learning goal, the output knowledge may act as the reason for further learning activity. Hence, the feedback of output knowledge, as input into the learning activity, depicts the iterative nature of the learning process.

For many of the design activities, the knowledge change (learning) occurs in-situ or provisionally. But for certain types of experiential knowledge learning is of utility value if it is learnt in retrospect. Hence the model features three types of temporal learning triggers; retrospective (T_r), in-situ (T_i) and provisional (T_p)). For the retrospective learning trigger, the learning activity succeeds the design activity ($A_d \rightarrow K_t$). For the provisional learning trigger, the learning activity precedes the design activity ($K_t \rightarrow A_d$). For the in-situ learning trigger, the learning activity occurs concurrently with the design activity ($A_d \mid\mid K_t$).

The output knowledge, O_k, is stored in the memory as experiential design knowledge that is basically an accumulation of output knowledge from the design task(s). Within the memory is also the knowledge of knowledge transformers, K_k, and the mapping between knowledge of knowledge transformers, K_k, and the repertoire of design activities, A_{dk}.

6 Conclusion

Through the research methodology of pragmatic scientism, a novel model of learning in design called LinD was derived. The model has explanative capability in explaining the phenomenon of learning in design, although its predictive capability is limited [1]. Nevertheless, it is envisaged that LinD provides a formal basis upon which effective and efficient design system(s) can be developed.

References

[1] Sim, S. K. (2000a). Modelling Learning in Design, Ph.D. Thesis, University of Strathclyde, Glasgow, United Kingdom.

[2] Smithers, T. (1996). On Knowledge level Theory of Design Process, in J. S. Gero and F. Sudweeks, *Artificial Intelligence in Design '96*, Kluwer Academic Publishers.

[3] Korn, J. (1996). Domain-independent Design Theory, *Journal of Engineering Design*, Vol. 7, No. 3. Journals Oxford Limited.

[4] Miller, G. G. (1995). *Living Systems*, University of Colorado Press, 1995.

[5] Smithers, T. and Troxell, W. (1990). Design is intelligent behaviour, but what's the formalism. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 4, 89-98.

[6] Duffy, S. M., Duffy, A. H. B. & MacCallum, K. J. (1995). A design re-use model, *International Conference of Engineering Design (ICED)*, Vol. 2, 490-495.

[7] Duffy, A. H. B., Persidis, A., & MacCallum, K. J. (1995a). NODES: A numerical and object based modelling system for conceptual engineering design, In *Knowledge Based Systems*, (Edmonds, E. A., Ed.). Elsevier Science Publishers Ltd., Amsterdam.

[8] Duffy, S. M. & Duffy, A. H. B., (1996). Sharing the learning activity using intelligent CAD, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1996, Vol. 10, 83-100, Cambridge University Press.

[9] Smithers, T., Tang, M. X., Tomes, N., Buck, P., Clarke, B., Lloyd, G., Poulter, K., Floyd, C., Hodgkin, E. (1992). Development of a knowledge-based design support system, *Knowledge-Based Systems*, Vol. 5, No. 1, 31-40, Butterworth-Heinemann Ltd.

[10] Sim, S. K. & Duffy, A. H. B. (1998b). A foundation for machine learning in design, *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 12, 193-209.

[11] Sim, S. K. & Duffy, A. H. B. (2000b). Evaluating a model of learning in design using protocol analysis, Sixth International Conference on Artificial Intelligence in Design – AID00. Worcester Polytechnic Institute, Worcester, USA.

Dr Siang Kok Sim
School of Mechanical and Production Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798, Republic of Singapore. Telephone: 65 7905494
Fax: 65 7911859 Email: msksim@ntu.edu.sg

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

THE METHODS LAB – A USER RESEARCH METHODS TYPOLOGY FOR AN INCLUSIVE DESIGN PROCESS

A S Macdonald and C S Lebbon

Keywords: Human factors, user evaluation.

1 Introduction

The creation of successful designs that suit the greatest number of users, to be 'inclusive', is an ever more challenging task. In recent years, this process has benefited from some convergence among the disciplines of design and its related specialisms. User research methods, which range from near-market to highly conceptual, from conventional to experimental, from quick and easy to detailed and exhaustive, have become increasingly vital in understanding users' behaviour and needs. Until recently, no typology existed to assist designers in the selection of methods. This paper discusses the origins and development of the Methods Lab [1], the objective being to build a definitive resource of user research methods in design primarily for design students, students of user research sciences, and designers and researchers in their first years of professional practice. The result is a compendium of user-research methodologies, presented in a 'designer-friendly' format. One of the issues that arose out of the research for the 'Lab' was that even professionals experienced in user research consultancy had only a limited awareness of the range of user research methods utilised in a broad range of companies. It was also found that designers have limited awareness of the diversity of methods available each of which has a value at a different stage of the design process. The format and the content of the Methods Lab are described, together with an evaluation and proposals for its future development.

2 Description of the Methods Lab

2.1 Design and content

The Methods Lab is a visually rich piece of graphic design intended not only to appeal to the designer but to hold all the information that the designer requires in order to select an appropriate user research method. It is schematic and supported with text, making it equally accessible to other disciplines and the business community. Its format allows it to be used as a ready-reckoner of user research methods, and its identity is communicated through a distinctive elongated orange format. The Methods Lab comprises several discrete components: the 'User Research Methods', the 'Map', the 'Icons', and the 'Finder'.

2.2 The Methods

It will be useful firstly to describe the Methods. In researching the Methods Lab, fifty three distinct user research methods were identified and organised into a typology in the 'Finder'.

These are grouped for rapid navigation under a number of headings: future creator; imagine and act out; professional trackers; direct design experience; co-design; co-research; expert observation; stimulus and interview; and current customer information. Some of the methods will be very familiar to engineering designers, others less so. For instance, lifestyle studies, expert interviews, questionnaires/surveys and opinion polls all appear under the group heading professional trackers. Rapid prototyping, and usability testing appear under co-design. For the purposes of development and initial research into the usability and appropriateness of the Methods Lab, out of these fifty-three methods, sixteen were selected for development in detail, with the stated ambition that further descriptions for the remaining methods would be added in the future. Each description was written by a globally-recognised individual, an authority in his or her own field. They were required to write with maximum concision. References or web links were also provided for additional information. For instance, 'Scenarios', written by Alison Black of IDEO, London, UK, is summarised as follows: '*constructing stories can help design teams propose new design concepts from an understanding of people's present experience*' and a two hundred word summary further develops the description of this Method. It also refers to other Methods used in conjunction with 'scenarios', e.g. 'role play', and provides a reference for further reading and a web-link. Figure 1 illustrates other graphic elements which accompany the text for the scenarios, and which are described below: the Methods Map, and the Method Icons.

'Rapid Prototyping' will be a Method familiar to engineering designers, described as *'a way of making realistic models of product concepts quickly from CAD data that can be evaluated by clients and users'*. Written by Ed Matthews of Pearson Matthews, it is linked to Time Compression Technologies, and is used in conjunction with alpha testing, user as developer, and future concept prototypes, all Methods listed in the Finder described below. Other familiar methods grouped under Stimulus and Interview include individual interviews, focus groups, and conjoint techniques, the latter being described as allowing '*researchers to establish how much consumers value individual features of products or services.*' Longitudinal analysis, designed to provide data on physical and mental change, shadowing to provide deep user understanding, and direct observation to provide behavioural data are three of the nine methods described under the Expert Observation section.

Users of the Methods Lab from different professional fields will find that some of the Methods will be very familiar, others less so, and some completely unknown to them. One of the aims of the Lab is to share knowledge from different fields in the hope that it will provide a greater range of methods to allow a more user-centred, inclusive, approach to design. The Methods described in the Lab are well established and proven, and will be of undoubted value during the engineering design process.

2.3 The Finder

The Finder is designed to impart, at-a-glance, a summary of information the designer will require in order to select an appropriate user research method. It comprises a table with the following headings: Typology - methods are grouped in typologies as described in 2.2 above for rapid navigation; Particular Methods - as the Lab is a work in progress, only a selection of methods are described in full and further entries will be added in future; and Method Number - these are positioned in the Methods Map, described below, for comparison, and help locate the full entry in the Methods section. The Finder is designed as a fold-out chart. Together with listing the grouped methods, two further sections assist the designer in the selection process – Output and Input. Output indicates the main benefit to the design team, e.g. 'rapid ethnography' provides *'quick hands-on information'*, and 'immersive experience' provides *'first-hand knowledge', direct experience and depth of knowledge'*), and Input indicates the resources required - expertise, time, staff, and costs.

2.4 The Methods Map

The designer of the section of the Lab known as the Methods Map, Malcolm Johnston, identified that "a major problem with design methodology is accessibility. Often this results from dense, textually based descriptions - but for those prepared to undertake their translation into practice there remains (the difficulty of) how to rapidly locate methods appropriate for any given design project." In order to overcome this difficulty, Johnston proposed a small set of 'logical criteria' to locate methods, and that "in the context of designing usable and desirable products for people, this map proposes two pairs of complementary descriptors, one pair representing the product to be designed, and the other the product's creators and users."

The Methods Map, the diamond-shaped graphic 'map' in Figure 1, locates the relative position of each of the Methods listed in the Finder at a point along two axes. In the horizontal axis, one end (designer-centred) locates a place where a designer would use a method which required no external references. One such Method, located near this left-hand end of the axis, is immersive experience where researchers may 'immerse' themselves in the user experience in order to gain deeper insights into actual life circumstances (Method 18). By contrast, the opposite, right-hand end of the horizontal axis, is the point where ideally each user's needs would be individually met (user-centred). In this area can be found opinion polls (Method 13), - *'representative samples of people are asked questions in order to gain a reliable measure of the views of an entire population'*, and conjoint techniques (Method 41) already described in 2.2 above.

The vertical axis is described by the designer as depicting design projects concerned with purely visual qualities at the top, ranging to those where functional qualities are predominant at the bottom. For instance, 'preference testing' (Method 42) located in the 'Stimulus and Interview' section, and "projective/visual research methods' (Method 29) located under the 'Co-research' section represent those methods more associated with visual qualities, whereas 'physiological testing' (Method 38) located in the 'Expert Observation' section, and 'usability testing' located in the 'Co-design' section represent those methods more concerned with functional qualities.

Johnston's intention in the design of the Map was to provide a quick way to identify candidate methods for a given aspect of a product, and to provide a designer-centred tool. He said, "While the Map is offered as a prototype with only a small number of methods analysed and placed, its value for locating a method is immediately apparent. It is not a precise tool, but a quick way to identify possible candidates for a given aspect of a project - and also a way to keep in mind the scope of a design project."

2.5 The Icons

The inputs of expertise, time and staff or manpower required, and costs for each of the Methods are represented through 'Icons' with a five-level graphics scale that indicates the minimum to maximum level of input required (Figure 2). Again these are meant to be indicative rather than

Scenarios | 4

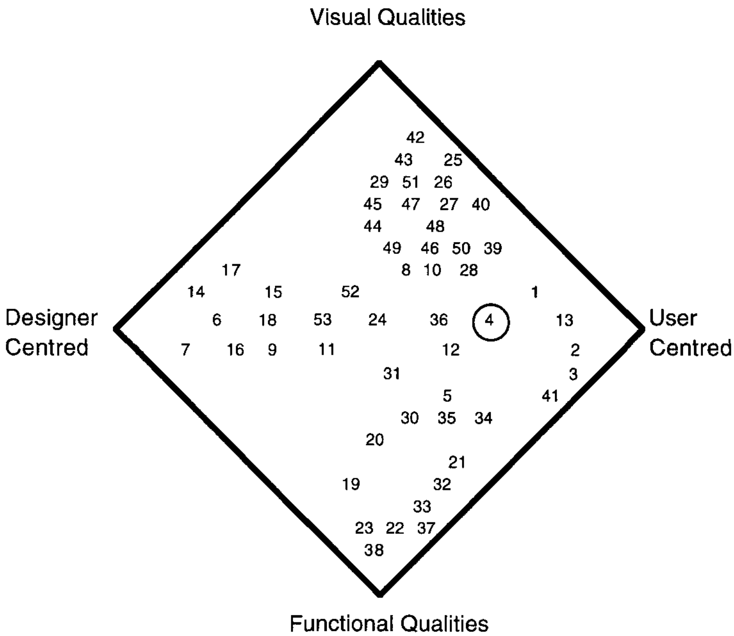


Figure 1. Methods Map 4 - Scenarios. The numbers refer to different Methods, and the number '4' is circled to indicate this 'Scenarios' Method and its position within the Map.

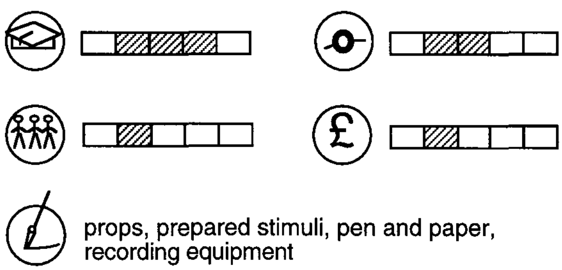


Figure 2. Scenarios - Input Icons

(Both ifigures adapted from the Methods Lab for purposes of reproduction)

definitive. For instance longitudinal analysis requires quite a high level of expertise, a long time to carry out the research, a modest number of staff involved, and is costly to run, Role play, by contrast, requires a competent degree of expertise, but requires little resource in terms of staff and time.

3 Development of the Methods Lab

3.1 Origins

I~design (Inclusive Design) is a consortium initiated by the Royal College of Art's (RCA) Helen Hamlyn Research Centre (HHRC), comprising the HHRC itself, The Engineering Design Centre of the University of Cambridge, the Design for Ability Unit at Central St Martins School of Art and Design, The Design Council, Tangerine, and IDEO. Work by i~design consortium member IDEO, in particular a report produced by Glasgow graduate Ewan Duncan [2], included a user research methods typology. This report was compiled from a detailed study of the user research methods used by a broad range of companies, among whom the better known are, Nike, Apple, Sony, Body Shop, Lexus, Xerox Parc, and Gallup, in the development of new products and services. This report was later taken up as part of the Presence research programme - one of thirteen EU-funded projects under i3 (the European Network for Intelligent Information Interfaces).

3.2 Development

Through a series of Presence working groups, which came to be known as 'Tea Parties', the format and content of the Lab was developed. The Methods Lab evolved to its current form through a 'user discussion method', an iterative process of discussion, refinement, critique, and evaluation, that is, a process of frequent peer evaluation events during its development.

A mix of designers, researchers and ergonomists, working together with a coordinator and an editor-in-chief, met following a 1997 Presence workshop to consider user research inputs to the i~design programme. Discussion was focused through two questions - 1) what was the best basis for building and structuring a resource of user research methods and case studies, and 2) what value did user research really add to the design process? Following on from this initial gathering, the aims of the second more in-depth discussion were to determine what kinds of user research information would be most useful to designers and how they might best be communicated. A series of further meetings was held to develop material for the Lab, to focus on developing a publication, either web-based or print, and a decision was made to develop a compact print version of IDEO's original typology of methods.

3.3 The Design

The design of one of the elements of the Lab, the Map (2.4 above) has already been described to explain its working. The original for the Methods Lab booklet was guided by Jennet Jessel, a member of the research team. It was essential that a product designed for designers would appeal to and be used by them. As part of this process she came up with a set of assumptions, based on previous work and experience, about the way designers like to take in information, and used this to develop her method for communicating to designers. Jessel's assumptions were:

1. Designers don't read (particularly lots of text)
2. Designers don't always listen
3. Designers think with pictures
4. Designer think in 'blobby', often unrelated, thought patterns
5. Designers go through a process of translating received information into imagery they can work with
6. Designers often have trouble translating visual concepts back into non image-led communication
7. Designers still have a lot of the 'kid' in them, i.e. they like to play, explore and be rewarded particularly by the creation of new 'stuff'
8. Designers like 'pretty things'

Jessel didn't deny that designers need to work with hard facts but it is the fact that they always have a visual agenda for a project that it is crucial to understand. She pointed out that designers prefer 'soft' information preferably in 'bite-sized chunks'. Any information also needs to be presented in a visually attractive way so that it can more readily be translated and absorbed by the designer. This had to be balanced out with the need for text-based information that could be incorporated into reports, proposals, etc. The eight assumptions were used as design guidelines by the graphic designer, Sione Raaijmakers, a graduate from the Royal College of Art, with Jessel providing art direction for the publication. The final design comprises a slim (296mm x 105mm, spiral-bound - along the long edge) publication with a section which folds out to a 296mm x 305mm Methods Lab Finder (2.3 above)

4 Evaluation

The Methods Lab was evaluated by both student users and professionals. Sixteen fifth year product design engineering students were surveyed individually for their views on the Methods Lab as published in the 'New Media for Older People' text, a second version of the Lab [3], printed in a different format to the original 'orange' publication. All students, prior to introduction to the 'Lab' had informally used a variety of user research techniques which varied with each individual and with the nature of the product being designed.

The group was asked to comment on Methods Lab as a whole and also its separate constituent components. All students had positive comments to make regarding the 'Lab' as a tool. A number found the structuring and variety of methods outlined helpful for broadening their understanding of methods available, that it was useful at the start of a project and for those who were unfamiliar generally with the broad spectrum of user methods. Most felt each of these components were crucially inter-related and that one would not be useful without the others. A more detailed analysis of the student user responses can be found in Macdonald and Lebbon [5].

The Lab was also distributed to a number of professional users of research methods, including IDEO, Acquaman, Cambridge Consultants, Telenor, Brunel University, Glasgow School of Art, and P5 Consultants. Responses have been very positive, and one interesting fact emerged: although consultants in the field had used methods for a number of years, one comment, typical of many responses was that "there were lots of new things I hadn't come across before".

5 User research methods and the design process

One issue which arose during the evaluation of the Lab by product design engineering students was the issue of case studies illustrating how the Methods Lab could be used in practice, particularly with reference to the design process. Initially, it was thought that Methods would be chosen as required and as felt appropriate by designers. However, Table 1 provides a speculative view of which Methods might be used at various stages of the design process, a view that requires to be tested. Some of the Methods are more appropriate at some stages of the design process than others, and some Methods are valuable at more than one stage.

Table 1 Examples of Methods linked to stages in the design process

Design Stage	User Research Method
Market	lifestyle studies (8), trend tracking (10), expert interviews (11), questionnaires/surveys (12), opinion polls (13), longitudinal analysis (30), video ethnography (31), direct observation (36), interview (39), focus groups (40)
Specification	task analysis (32), time and motion studies (33), shadowing (34), direct observation (36), physiological testing (38)
Concept design	scenarios (4), role play (6), explore (7), interviews (39), focus groups (40)
Detail design	rapid prototyping (20), usability testing (21), video ethnography (31), direct observation (36), interviews (39), focus groups (40), conjoint techniques (41)
Manufacture	rapid prototyping (20), test markets/probes/pilots (50)
Sell	customer return cards (44), sales figures (46), in-built tracking (48), promotional retail (53)

6 Conclusions - context and future plans

The first responses to the Methods Lab have been very positive. It will obviously benefit from further development, user trials and feedback. The ambition is to detail in total the fifty-plus identified Methods, with each description written by an international authority. Undoubtedly other lesser known methods will emerge as more individuals and organisations become aware of the Lab. There are now plans to build on this work as a basis for a more publication of the output of the HHRC strand of the i~design programme - 'Inclusive Design: strategies, tools and user-research methods'. There are also plans for visual impairment access in the next version. The contents would include: a timeline bringing together key events, legislation, publications and design exemplars to demonstrate the emerging trend towards inclusivity; a rationale for the grouping and organising of the strategies (e.g. 'Transgenerational Design'), tools (e.g. Pirkl's charts of ageing characteristics and design responses), and user-research methods (e.g. 'Immersive Experience'); an overview of relevant inclusive design strategies (theories and concepts), their background, key features, and historical origins, (20+ identified to date); a reference listing of available design tools for inclusivity, (30+ identified to date); and a mapping of relevant user research methods, (50+ identified to date).

References

[1] Aldersey-Williams, H., J. Bound and R. Coleman (eds) 1999. *The Methods Lab: User Research for Design*, Design for Ageing Network, Helen Hamlyn Research Centre, Royal College of Art, London.

[2] Duncan, E. 1994. *User Research* (an internal draft discussion document) (IDEO, San Francisco) - *by kind permission of Bill Moggridge.*

[3] Hofmeester, K., and E. de Charon de Saint Germain (eds) 1999. The Methods Lab: User Research Methods, in *Presence: New Media for Older People, (*Netherlands Design Institute) 118-165.

[4] Jessel, J. 1998. www.presenceweb.org (Discussion Forum section) *(Web site no longer linked but document available on request*).

[5] Macdonald, A.S., and C.S. Lebbon, 2001. The Methods Lab: developing a usable compendium of user research methods, in *Contemporary Ergonomics 2001*, Taylor and Francis, London (in press).

Acknowledgement

To Hugh Aldersey-Williams, editor in chief, John Bound, project coordinator, and Roger Coleman, Co-Director of the HHRC, original editors of The Methods Lab for making their material so freely available and allowing us to borrow from their words.

Alastair S. Macdonald
Glasgow School of Art
Department of Product Design Engineering
167 Renfrew Street
Glasgow G3 6RQ

Telephone +44 (0) 141 353 4614
Facsimile +44 (0) 141 353 4655
Email a.macdonald@gsa.ac.uk

Cherie S. Lebbon
Royal College of Art
Helen Hamlyn Research Centre
Kensington Gore
London SW7 2EU

Telephone +44 020 7 590 4242
Facsimile +44 020 7 590 4244
Email c.lebbon@rca.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

VERIFICATION OF A MODEL OF SYNTHESIS – THE METHODS FOR VERIFICATION AND RESULTS

A Tsumaya, Y Nomaguchi, M Yoshioka, H Takeda, T Murakami, and T Tomiyama

Keywords: protocol analysis, process modeling, synthesis, verification, reference model

1 Introduction

Engineering design consists of a variety of thought processes, but most of them can be classified into two types, analysis and synthesis. There is a need to build an advanced CAD system that can support not only analysis but also synthesis. However, traditionally, CAD technologies have paid little attention to synthesis. Our project team, "Modeling of Synthesis", has focused on the synthesis thought process [1] and proposed a thought process model based on knowledge level operations [2][3]. Following these results, this paper reports the verification phase of the project.

We verify the knowledge operation model with the following two methods. One is to test the model against experimental data, and the other is to implement a system based on the model to examine if the system behaves in accordance with the designers' behavior. In this paper, we use the former method and evaluate the applicability of the model by comparing it with other existing design process theories.

In this paper, first, we explain how to select a design case for verification. Second, we propose a method to build a reference model of the design case. This reference model is framed in the knowledge operation model and other existing design process theories, viz., the cognitive design process model of Takeda *et al.* [4] and German design methodology of Pahl and Beitz [5]. By doing so, we can compare the knowledge operation model with these existing theories. From this comparison, we can evaluate and verify the knowledge operation model.

2 Selection of a design case for verification

Before verification, we need to set up a method to select a design case. Observing and recording an actual industrial design case is difficult because of proprietary reasons and data amount. Protocol analysis of design experiments such as Delft design protocol [6], provides us with detailed data, but this approach often became a toy case, because of the limitation of resources and laborious work for analysis.

For the verification of a design process model including synthesis process, design cases are required to satisfy the following conditions:

- The design cases should be real design cases, but they should not be toy problems nor created cases for verification.

- The design cases must have "newness" which is the most important essence of synthesis.
- Concrete data about the design process should be recorded.
- The data of the design cases should be of a reasonable size for analysis. They should not be too huge or complicated.

Considering these conditions, we selected a machine design conducted in a laboratory at the University of Tokyo over four years. The design was the development of a high precision stereo lithography machine for micro photoforming fabrication of micro flexible mechanisms [7]. This research contains "newness" that tries to increase manufacturing accuracy, and the amount and size of the data are reasonable. As the sources of information, we used three bachelor theses completed in 1995, 1996, and 1997, a PhD thesis in 1998, summary reports of the research written occasionally, and weekly reports that usually consist of several lines about what the student did in a week. We investigated these sources to analyze design activities.

From this design case, we built a "reference model" of design process for verification of the model of synthesis.

3 The method to build a reference model

3.1 Building a reference model using design activities

After selecting the design case, we built a reference model. To do so, we introduced the concept of the frame cognition model that classifies design activities into some categories, viz., "naming", "framing", "moving" and "reflecting" [8]. Figure 1 shows the relationships among them. The subject that relates to the present design conditions is posed by "naming". The "framing" builds up a problem in a frame. The design proceeds to a solution categorized to "moving". The solution is evaluated by "reflecting". Design protocols are analyzed and categorized into these four activities. Using this classification, the design case was analyzed into sequentially framed design processes and a reference model was obtained.

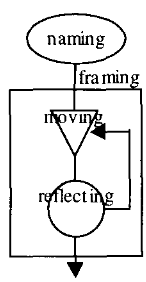


Figure 1. Design Activities of the Frame Cognition Model [8].

The frame cognition model had some problems as well, because this model was originally proposed as a result of the protocol analysis of the conceptual design of a simple component

[6]. Therefore, we had to modify the frame structure to accommodate our design case that was more complicated than the original design experiment. We needed to have, first, hierarchical structure of design processes and, second, consideration about information flow through design activities. Adding these two modifications, design activities were picked out from the sources of the design case, and arranged sequentially. Figure 2 illustrates a part of the reference model displayed in the frame cognition model.

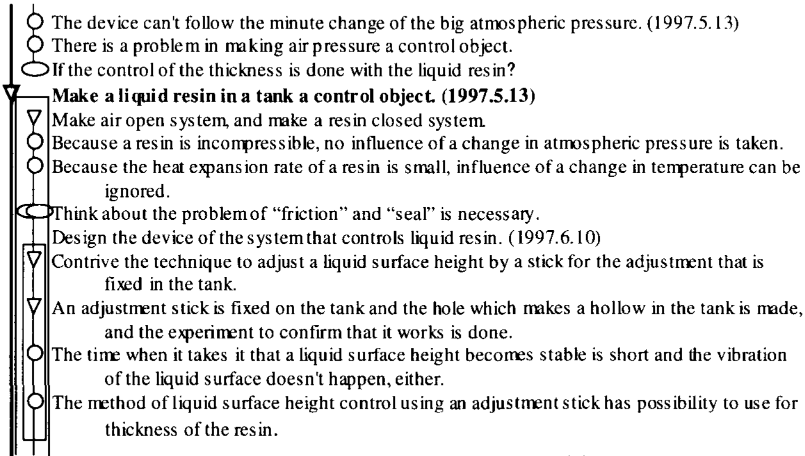


Figure 2. Part of the Reference Model.

Then, we began to verify our proposed design thought process model based on knowledge operations (for further details, interested readers might refer to [2] and [3]). This model, called the knowledge operation model, contains seven knowledge operations: "knowledge/information acquisition", "knowledge/information reorganization", "information confirmation", "conflict resolution", "knowledge/information revision", "solution synthesis", and "object analysis". For the verification of completeness of the knowledge operation model, we checked the correspondence of the seven knowledge operations with design activities in the reference model. Consequently, we could confirm that the seven knowledge operations cover the reference model. Roughly speaking, three knowledge operations, "solution synthesis", object analysis", and "conflict resolution", are cycling in this order. "Information confirmation" is usually observed when the focus of design changes from one component to another, from the whole design object to components, and from a component design to the whole. Another three knowledge operations appear randomly.

3.2 Using vocabulary about design

To evaluate the applicability of the knowledge operation model, we compared it that represents the reference model with other design process models (such as [4] and [5], see Section 4). This resulted in some problems, however, and the biggest problem was that one design activity in the reference model sometimes corresponded to more than one stage of these design process models. This happened due to the difference in the concept granular sizes between a design activity and a stage of the design process models. Because of this, we introduced an intermediate abstract level between them, called *design vocabulary* and composed of standardized terms about design to represent each design activity at more general and detailed level.

For example, we show the following design activities from the reference model.

- *"Because of the moving mechanism of the table at the top, the upper contact side of the tank deviates and a hollow warps caused by the movement of the table."*

- *"Should I fix a table (X-Y table) at the top of the tank?"*

- *"The X-Y table is relocated to the bottom of the tank, and the top of the tank is unified with the tank."*

This series of design activities can be interpreted with the design vocabulary as follows.

- The "estimation" of "*Because of the moving mechanism* ..." and "knowledge acquisition" were performed due to the contradiction in the experiment, and information is collected by the "investigation" to solve the problem.

- The "idea" of "*Should I fix* ..." appeared in the process of the "arrangement of the information" of the collected information.

- According to the idea, "knowledge acquisition" and "knowledge reorganization" are carried out, and then "suggestion" of "*The X-Y table is changed*..." is made.

Using this interpretation, the three design activities in the example above are represented more precisely and uniformly by eight terms. In the same way, each design activity can be explained with the design vocabulary. The vocabulary included 112 standard terms that were obtained through brainstorming of all members of the project. All design activities in the reference model were interpreted, and consequently, 102 design activities were converted into 223 terms correlated to 23 categories of the design vocabulary. We also analyzed the correspondences between the knowledge operation model and the design vocabulary. As a result, we found out that each term in the design vocabulary is correlated to only one of the knowledge level operations. Table 1 shows the result of the analysis.

Table 1. The results of the analysis by using vocabulary about design.

Stage of the knowledge level operation (times of appearance)	Vocabulary about design (times of appearance)
Knowledge/information acquisition (61)	investigation (26), knowledge/information acquisition (19), problem indication (16)
Knowledge/information reorganization (33)	arrangement of the knowledge/information (13), knowledge/information reorganization (10), making concrete (7), drafting (3)
Information confirmation (10)	confirmation (10)
Conflict resolution (8)	conflict resolution (8)
Knowledge/information revision (9)	strengthening of the constraint (6), knowledge/information revision (3)
Solution synthesis (44)	suggestion (24), idea (8), selection (7), improvement (3), decision (1), association (1)
Object analysis (58)	evaluation (27), trial manufacture (14), experiment (9), estimation (4), numerical analysis (3), derivation (1)

4 Comparison of the knowledge operation model with existing design theories

In this section, we verify the knowledge operation model by comparing the model with other existing design theories. The method for comparison between the knowledge operation model and the existing design process theories is shown as follows.

1. Analyze the knowledge operation model that frames the reference model with the design vocabulary, and check the correspondences of the knowledge operations and the terms in the design vocabulary.
2. In the same way, analyze and check correlation between stages of an existing design process theory and terms that represent the reference model.
3. Obtain the relationship among the design process models using the terms in the reference model as a mediator.

4.1 Comparison with the cognitive design process model

First, we compared the knowledge operation model with the cognitive design process model of Takeda *et al.* [4] using the reference model as a mediator. The cognitive design model has the following five sub-processes, viz., "*awareness of the problem*", "*suggestion of candidate solution*", "*development*", "*evaluation*", and "*decision*". Table 2 illustrates the comparison result between the two models that shows good correspondences between them. For instance, "*Suggestion of candidate solution*" corresponds to "**solution synthesis**", "*Development*" to "**object analysis**", and "*Evaluation*" to "**object analysis**". "*Decision*" corresponds to "**information confirmation**" and "**conflict resolution**". In the former, the designer made positive judgment to a solution whereas the designer found some problems about a solution in the latter. "*Awareness of the problem*" corresponds to "**knowledge/information acquisition**", "**knowledge/information reorganization**", and "**knowledge/information revision**", but often implicitly.

We also found that the cognitive design process model cannot classify some terms of the knowledge operation model. This is because the cognitive model treats nothing about information from outside, whereas the knowledge operation model has consideration about that.

4.2 Comparison with German design methodology

In German design methodology (e.g., [5]), the designer begins design with analysis and decomposition of functional requirements, followed by embodiment of function into structure. Because our design case is conceptual, we focused on two parts of the design process in German design methodology, viz., "clarification of the task" and "conceptual design". "Clarification of the task" contains two stages, i.e., "*clarify the task*" and "*elaborate the specification*". "Conceptual design" consists of "*identify essential problems*", "*establish function structure*", "*search for solution principles*", "*combine solution principle and select suitable combination*", "*firm up into concept variants*", and "*evaluate against technical and economic data*".

Next, we compared the knowledge operation model with German design methodology. Table 3 exhibits the result of the comparison, "*Identify essential problems*" corresponds to "**knowledge/information acquisition**", and "*Establish function structure*" to "**knowledge/information reorganization**". "*Search for solution principles*" and "*combine solution principles and select suitable combinations*" correspond to "**solution synthesis**". "*Firm up into concept variants*" corresponds to "**object analysis**". "*Evaluate against technical and economic criteria*" corresponds to "**object analysis**", "**information confirmation**", and "**conflict resolution**". In this stage, first, "evaluation" of a design solution is performed, followed by "confirmation" or "conflict resolution". "*Clarify the task*" corresponds to "**knowledge/information acquisition**", and "*elaborate the specification*" to "**knowledge/information revision**". These two mainly follow "conflict resolution". This result shows similarity of the two models. However, our proposed model has advantage to interpret functional design in that our model can not only acquire and refer to functional information but also treat other types of information, such as physical entities which traditional functional decomposition method cannot handle.

Table 2. Comparison the knowledge level operation model with the cognitive design process model.

		awareness of the problem	suggestion	development	evaluation	decision	cannot be classified
knowledge / information acquisition	investigation						26
	knowledge/information acquisition						19
	problem indication	16					
knowledge / information reorganization	arrangement of the knowledge/information						13
	knowledge/information reorganization	2					8
	making concrete	7					
	drafting						3
information confirmation	confirmation					10	
conflict resolution	conflict resolution					8	
knowledge / information revision	strengthening of the constraint	6					
	knowledge/information revision	3					
solution synthesis	suggestion		24				
	idea		8				
	selection		7				
	improvement		3				
	decision					1	
	association		1				
object analysis	evaluation				27		
	trial manufacture			14			
	experiment			9			
	estimation			4			
	numerical analysis			3			
	derivation			1			

Table 3. Comparison the knowledge level operation model with German design methodology.

		clarify the task	elaborate the specification	identify essential problems	establish function structures	search for solution principles	combine solution principle and select suitable combination	firm up into concept variants	evaluate against technical and economic criteria	cannot be classified
knowledge / information acquisition	investigation			26						
	knowledge / information acquisition	1		5						13
	problem indication	16								
knowledge / information re-organization	arrangement of the knowledge / information				13					
	knowledge / information reorganization				9					1
	making concrete				7					
	drafting									3
information confirmation	confirmation								7	3
conflict resolution	conflict resolution								8	
knowledge / information revision	strengthening of the constraint		6							
	knowledge / information revision		3							
solution synthesis	suggestion					24				
	idea					8				
	selection						7			
	improvement					3				
	decision						1			
	association					1				
object analysis	evaluation								27	
	trial manufacture							14		
	experiment							9		
	estimation								4	
	numerical analysis							3		
	derivation							1		

5 Conclusions

We proposed a new method, called a reference model, to verify the knowledge operation model of synthesis. To do so, we investigated how to select a suitable design case. A reference model was built from an actual design case and design activities were identified. Furthermore, we introduced abstract standard terms, called design vocabulary, to build a reference model. By using the reference model described with the design vocabulary, we confirmed that the knowledge operation model could cover the reference model with good correspondence. This justifies the completeness of the knowledge operation model. We also compared it with other design process theories and found that the knowledge operation model is compatible with them.

This research was supported by the *Modeling of Synthesis Project* (JSPS-RFTE 96P00701) under the *Research for the Future Program* of the Japan Society for the Promotion of Science.

References

[1] Tomiyama, T., Murakami, T., Washio, T., Kubota, A., Takeda, H., Kiriyama, T., Umeda, Y., and Yoshioka, M., "The Modeling of Synthesis – From the Viewpoint of Design Knowledge", Proc. ICED '97, Tampere, 1997, pp.97-100.

[2] Yoshioka, M. and Tomiyama, T., "Toward a Reasoning Framework of Design as Synthesis", Proc. 1999 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, The American Society of Mechanical Engineers (ASME), Las Vegas, 1999, DETC99/DTM-8743 (CD-ROM).

[3] Takeda, H., Yoshioka, M., and Tomiyama, T., "A General Framework for Modeling of Synthesis Integration of Theories of Synthesis", Proc. ICED 2001, (submitted)

[4] Takeda, H., Veerkamp, T., Tomiyama, T., and Yoshikawa, H., "Modeling Design Processes", AI Magazine, 11, 1990, pp.37-48.

[5] Pahl, G. and Beitz, W., "Engineering Design: Systematic Approach", Springer-Verlag, Berlin, 1988.

[6] Cross, N., Christiaans, H., and Dorst, K. (eds.), "Analysing Design Activities", John Wiley & Sons, 1996.

[7] Xie, T., Murakami, T., and Nakajima, N., "Micro Photoforming Fabrication Using a Liquid Hollow Shaped by Pressure Difference and Surface Tension", International Journal of the Japan Society for Precision Engineering, 33, 1999, pp.253-258.

[8] Valkenburg, R.C. and Dorst, K., "The Reflective Practice of Design Teams", Design Studies, 19, 1998, pp.249-271.

TSUMAYA, Akira
Collaborative Research Center for Advanced Science and Technology, Osaka University, Yamadaoka 2-1, Suita, Osaka 565-0871, Japan, +81-6-6879-4191, +81-6-6879-4191, tsumaya@crcast.osaka-u.ac.jp, http://www.crcast.osaka-u.ac.jp/crcast23/index-e.html

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

IMPROVING REQUIREMENT SATISFACTION ABILITY OF THE DESIGNER

S Nidamarthi, A Chakrabarti, and T P Bligh

Keywords: Knowledge for Action, Reflective Practice and Requirement Satisfaction.

1 Introduction

This paper is based on our research in understanding designers' processes of ***requirement satisfaction,*** with the aim of proposing methods to improve their abilities in achieving this. By requirement satisfaction we mean the degree of fulfilment of functional, reliability and safety demands and wishes that a designer should accomplish in his/her design. In this research, protocol analysis is used to examine the design process at a detailed level. Designers were given a design problem, along with its requirements in the form of a descriptive text, and were asked to produce a clear, conceptual solution to fulfil the requirements. The design process was recorded in video and audio tapes for analysis. Three independent evaluators scored the concepts against each requirement. Using these scores and the recorded design process, the reasons for satisfaction of each requirement were identified. In this way, two individual designers and five design teams were studied [1]. This method of reasoning for each requirement's satisfaction differs from that of [2], [3] & [4], who studied overall fulfilment of the design by considering an average of all individual requirement satisfaction scores. These researchers' methods allow one to distinguish between successful and unsuccessful design teams. In addition to such comparison, our method allowed us to reason about successfully or unsuccessfully satisfied individual requirements within each team.

Based on our research findings, we propose several prescriptive methods to support designers in improving their requirement satisfaction abilities. We present the reasons for requirement satisfaction in Section 2, followed by the prescriptive methods in Section 3. We evaluated all the methods with the help of existing literature, and one of them by using it in a design process in an industrial setting. The results of using this method are also discussed in Section 3. Section 4 summarises the paper.

2 Reasons for Requirement Satisfaction

The observed design process was detailed in terms of ***design activities*** and their ***outcomes***. These activities were grouped into two stages namely ***Problem Understanding*** (referred to as PU) and ***Problem Solving*** (PS), and are shown in Table 1. For example, *problem identification* is a part of PU, which was characterised by the activities of *perception, inference* and *modification* of the given problem. Similarly, PS was characterised in terms of the activities listed in Table 1. These activities are explained in detail, and compared with related literature in [1].

Table 1. The designers' activities grouped as "problem understanding" and "problem solving".

Stage	Problem Understanding (PU)			Problem Solving (PS)		
Primary Level Activities	Identify	Analyse	Choose	Generate	Evaluate	Select
Secondary Level Activities	Perceive Infer Modify	Question, Relate, Weigh, Verify, Visualise	Decide	Create, Modify, Detail	Identify characteristics, Question, Relate, Verify	Identify things to do, Compare, Decide

Table 2 shows a discourse within a team designing an endoscope cutter that needs to hold and sever blood vessels in the stomach (for more details, see [1]). Here, the design team *questioned* the requirements, *visualised* the situation and *perceived* a new requirement that 'the endoscope should not over travel'. We term such requirements and solutions as *outcomes* of the design activities. These activities and their outcomes are highlighted in Table 2.

Table 2. An illustration of problem understanding by a design team (the outcomes are highlighted in the text).

Designer	*Text of conversation from video & audio tapes.*	*Activity*
D2:	"cleanly snip it" meaning can we actually ***cut surrounding tissues?***	PU.Question
D2:	This [the endoscope] isn't a tube flying across an empty space	PU. Visualise
D3:	[it is safe] as long as it [the endoscope] goes through cutting around any bit of flesh around the vessel, unless the vessel happens to be next to one of the major nerves.	PU. Visualise
D2:	That means the endoscope should not ***over travel***.	PU.Perceive

Table 2 illustrates the team's understanding of the endoscope cutter's requirements in *cutting* blood vessels *safely* (both requirements were given in the design problem). This team then went on to design an endoscope cutter with *travel control mechanisms* so as to avoid over travel. For this solution, the team had to satisfy specific-requirements such as *accuracy of the travel control mechanism*. We term these outcomes as ***solution-specific requirements*** (SSRs). In all our case studies, the designers addressed increasingly more SSRs as the design progressed further, and addressed the requirements given in the design problem through the solutions and their corresponding SSRs [1].

We have analysed the design process based on the design activities and how the outcomes evolve, and by relating these with the satisfaction score against each requirement (as scored by the independent evaluators). For example, the team featured in Table 2 scored very highly in fulfilling the safety requirement compared to another team of similar composition (both in the number of members and their individual backgrounds). While the successful team analysed the safety requirement with questions and visual pictures, the unsuccessful team did not do so. Similar observations are also made within the same team but between different requirements. From such observations, we identified and detailed the practices of PU and PS that are responsible for effective and ineffective requirement satisfaction. These results are shown in Figures 1 and 2. For example, the problem understanding process for the safety requirement by the successful team (Table 2) can be illustrated by Figure 1-c. The unsuccessful team simply read the safety requirement from the design problem without any further analysis (as in Figure 1-a), and detailed their solution without evaluating it (as in Figure 2-a). Similar observations were also made by [5], but qualitatively.

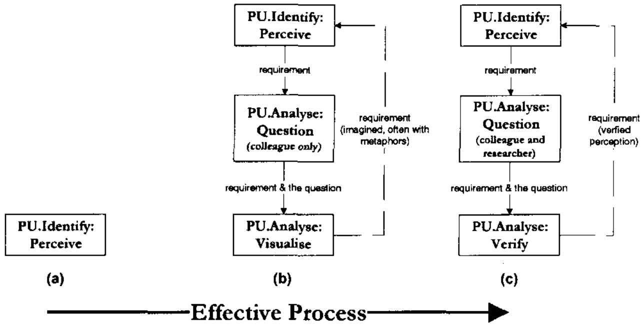


Figure 1. PU that appears to contribute to (a) the least, (b) medium and (c) high requirement satisfaction

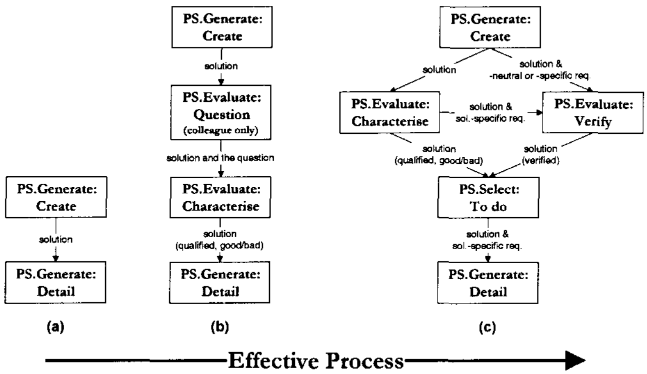


Figure 2. PS that appears to contribute to (a) the least, (b) medium and (c) high requirement satisfaction

In addition to the quality of design activities as explained above, the unsatisfied requirements seem to have been mostly ignored in the design process. This is illustrated in Figures 3 and 4. These figures show the percentage of completion of the design process devoted to fulfilling the particular requirement against the percentage of total design time. Any horizontal segment in these figures indicates inactivity by the team in that requirement for that duration (i.e., time on the X-axis). Similar observations were implied by [4], but only qualitatively, and these were related with the overall design success rather than the individual requirement satisfaction.

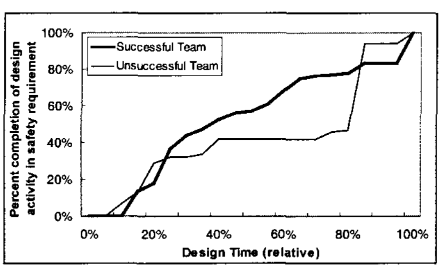


Figure 3. Distribution of activity related to safety requirement by two teams.

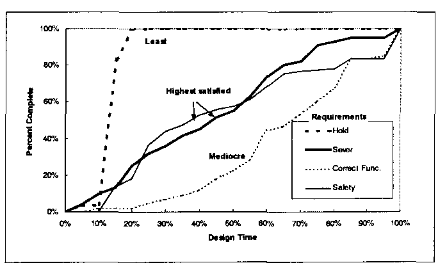


Figure 4. Distribution of activity related to various requirements by the team illustrated in Table 2.

The satisfaction of a requirement was also observed to be related to the percentage of solution-specific requirements which the designers identified and applied with respect to the solutions addressing that requirement (Figure 5). These solution-specific requirements convey designers' ability in elaborating solutions and making them fail-safe.

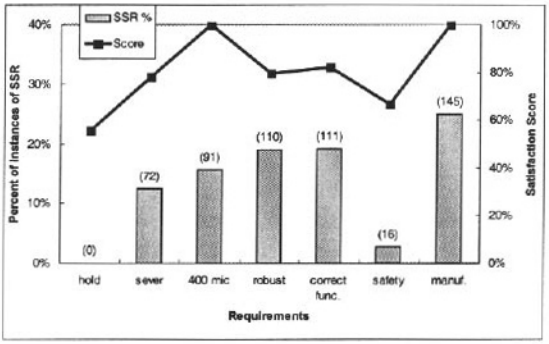


Figure 5. Percent of instances of solution-specific requirements (SSRs) for each requirement, and their correlation with requirement satisfaction (=0.85). The number of instances are showed in parentheses.

3 Prescriptive Methods to Support the Designer

Based on these results, the following prescriptive methods are proposed to improve designers' abilities in requirement satisfaction. The goals of these methods are formulated as follows.

1. Achieve better problem understanding and problem solving (i.e., PU as in Figure 1-c rather than 1-a).
2. Minimise overlooking of important, solution-neutral requirements for a long time.
3. Increase identification and application of solution-specific requirements.

To achieve these goals we propose human and computer-based methods.

3.1 Human based methods

These can be methods that help achieve self-sufficiency, where the designers intentionally or unintentionally follow the good practice (Figures 1-c & 2-c) in their design process. Further, these methods can be implemented in terms of external human intervention, where the designers are prompted to follow the good practice.

Self-sufficiency and Reflection-on-Action

The research observations provide a detailed knowledge in the form of reasoning for (un)successful requirement satisfaction. This knowledge can be used to educate design students or practitioners to make them self-sufficient in following successful processes. The researchers in [6] & [7] propose learning-by-teaching and learning-by-doing methods for achieving such goals. All these education methods embody the ***reflection-in-action*** and ***reflection-on-action*** qualities of the design process (after [8]) with varied levels of emphasis. The reflection-in-action includes reflecting on the knowledge previously gained in the design process. For example, verifying a solution, where the designer re-visits known requirements so that the solution satisfies them, is one form of reflection-in-action. This practice involves considering the design outcomes in an activity, especially looking back on the previous ones. The reflection-on-action involves reflecting on design activities. For example, a designer's realisation of a need to question a requirement, after identifying it, is one form of reflection-on-action. Because reflection-on-action involves referring to outcomes, it can be used to encourage reflection-in-action (e.g., reflecting on verification can be used to look back on relevant solutions and requirements). Both kinds of reflection can help designers to improve their requirement satisfaction. We propose the following reflection-on-action (Table 3) to be applied in the design education methods proposed by the above researchers. A detailed table is presented in [1].

Table 3. Methods to reflect-on the design activities that resulted in high requirement satisfaction.

Action	Reflection-on-Action	Intended Result
PU: Perceive	Have I perceived all requirements?	Identify as many requirements as possible so that no requirement is overlooked in PU.
PU: Question	Have I questioned this requirement?	Understand rationale for the requirement and share similar understanding with colleagues.
PU: Identify → Analyse	Have I analysed all listed requirements?	Minimise overlooking of 'analysis' activity after 'identifying' requirements as in the Figure 1-c.
PS: Create	Can I create more alternatives?	Generate more solutions so that the best alternative can be developed.
PS: Verify	Have I verified this solution?	Ensure that the solution is satisfying its requirements.
PS: Characterise	Have I characterised this solution?	Identify advantages and disadvantages of the solution.
PS: Generate → Evaluate → Select	Did I select this solution without evaluating it?	Minimise overlooking of 'evaluate' and 'select' activities after 'creating' solutions as in the Figure 2-c.

In order to be self-sufficient in minimising the overlooking of requirements, and in considering solution-specific requirements, the designer needs to both reflect-in-action (e.g., refer to requirements) and reflect-on-action (e.g. Did verification take place?). However, the usefulness of this method depends on the degree of self-consciousness of the designer. One way to increase this self-consciousness is through documenting requirements, solutions and solution-specific requirements, and referring to them in the design process. Another method is to take external individual's help rather than to depend entirely on self-consciousness. This method is discussed in the following section.

External Human Intervention Method and Knowledge-for-Action

This method can especially suit design teams, where one individual can act as a participating, observing and influential member to guide the design process for high requirement satisfaction. The key issue here is in knowing when to act. For example, if designers seem to be detailing solutions without evaluating them (as in Figure 2-a), the external individual can act by questioning the solution or asking the designers to verify it (to change the design process as in Figure 2-c). The research results can make the influencing individual knowledgeable about when to act for a better requirement satisfaction. That is, it is about knowing the relevant outcomes, recognising the need to act, and acting accordingly. We term this knowledge as ***knowledge-for-action*** [after 9]. Table 4 illustrates knowledge-for-action (further details are given in [1]).

Table 4. Knowledge-for-action: Knowledge required for suggesting certain action.

Knowledge for Action	Recommended Action
Sources of requirements contain more information than the requirements.	PU: Perceive
Absence of background information of requirements.	PU: Question
Only a few alternatives are generated.	PS: Create
Inconsistent solutions with respect to their requirements.	PS: Verify
Absence of reasoning -- for and against a solution.	PS: Characterise

The first author had an opportunity to act as an intervening individual during his case study in an industrial setting. In this study, the author documented the design process as a member of the team, and intervened in the process whenever necessary. The following is a list of some of the types of those interventions, how they were executed, and the number of instances of each (in parentheses). This list is arranged in ascending order of the amount of conscious effort involved in reflection-on-action on the part of the author.

- Providing details of a previously created solution when the team was unable to recollect them. The author reminded the team of the solution (once).

- Asking the team to evaluate solutions, when these were being detailed without evaluation; the author prompted the team to question, characterise and verify the solution, while reminding the solution (thrice).

- Asking the team to refer to high-level requirements, when the solutions being developed contradicted them; the author reminded them of the requirements and the solutions, and highlighted the contradiction (twice).

Each of these interventions resulted in a positive result. For example, the design had space constraints for its deployment. On one occasion, the team expanded the design in length and breadth without referring to these space constraints. The author reminded the team that they should verify the design with respect to the constraints. The team then evaluated the design, and corrected it. A majority of such interventions involved the author reminding the team about forgotten requirements and solutions. The documented design process was helpful in recalling these forgotten outcomes, and it could also be the cause of these interventions in terms of the author gaining the needed knowledge-for-action. Moreover, the effort in realising and recalling inconsistencies related to the outcomes was less than that related to the design activities. These observations illustrate the importance of documenting the design process. We base our computer-based methods for designers on the documented process, and discuss them in the following section.

3.2 Computer-based Methods

Existing computer tools such as QFD and requirements management tools [10], or design process capturing systems [11] use structured methods to record information, but do not provide active help to designers in terms such as reminding them about forgotten requirements. Although current computational reasoning does not match with that of the human to provide such help, it may at least be initiated in a simple form to assist designers.

For example, whether or not a requirement is questioned can be derived from whether or not that requirement's rationale is known. That is, the status of outcomes can be used to compute knowledge-for-action and thereby guide the designer for appropriate action. With this view, we present the computable knowledge-for-action with the following expected uses, see Table 5 (further details are given in [1]).

Table 5. Possible ways to use computation to provide active help.

Computable Knowledge for Action	Uses in terms of Suggested Action
1. Scan sources of requirements for keywords (e.g., must, should, will), 2. Use checklists (e.g. [12]) for identifying requirements.	PU: Perceive: 1. Highlight sentences with the keywords (these are likely to be requirements), 2. Highlight unchecked boxes or lists.
Check requirements for their rationale, and prompt the designer if absent.	PU: Question: Ask for rationale of the requirements.
Check for solutions for each major requirement, prompt the designer if absent.	PS: Create: Ask for solutions (more alternatives if possible).
Check solutions for their requirements (including SSRs), and for 'verification result', prompt the designer if absent.	PS: Verify: Ask to check the solutions with requirements, and document the 'verification result'.
Check solutions for their solution-specific requirements, which characterise the solution, and prompt the designer if absent.	PS: Characterise: Ask to assess the solutions for their good and bad qualities, and record them with appropriate links to the solutions.

The above knowledge-for-action can only be computed if the necessary information is recorded and the rules for prompting the designer are codified. For example, in order to prompt designers to verify their solution, the computer needs to know that 'the solution was not verified' (e.g., through an unfilled variable 'verification result'), and about 'the requirements that the solution refers' (through recorded links between solutions and requirements). This type of reasoning can be introduced into support systems like PROSUS [11] to encourage the good practice for better requirement satisfaction. In order to minimise overlooking of requirements, the computer may prompt the designer of a requirement if s/he did not access any documents related to that requirement for a considerable amount of time. In order to make sure that more solution-specific requirements are explored, the computer may emphasise the problem solving activities that produce SSRs (e.g., characterising), and those that apply SSRs (e.g., verification). We hope such a computer system can be developed, and validated for its usefulness.

4 Summary

In our research, we found the good practice in problem understanding (PU) and problem solving (PS) that are responsible for effective requirement satisfaction (Figures 1-c and 2-c). Moreover, the unsatisfied requirements seem to have been those that have been neglected for a considerable amount of time. Based on these results, this paper proposed prescriptive methods to support the designer in achieving better requirement satisfaction. These prescriptive methods were grouped into human and computer-based methods. Designers can be trained to reflect in and on their activities to ensure that the good practice is achieved. While reflection-in-action is essentially reflecting on the outcomes such as requirements and solutions, reflection-on-action requires being conscious about activities and the good practice.

An external individual who can intervene in the design process for its improvement can further enhance the conscious effort involved in this method. This paper detailed the necessary knowledge for one to realise when to intervene for a better process. This is termed knowledge-for-action. This intervention method was used in an industrial setting, and evaluation of its use highlighted the various degrees of conscious effort involved. From this result, we have proposed a computer-based method to enhance the self-conscious effort. Unlike traditional methods, this method is intended to provide active guidance to the designer by exploiting the documented design process.

References

[1] Nidamarthi, S. "Understanding and Supporting Requirement Satisfaction in The Design Process", PhD Thesis, Engineering Design Centre, University of Cambridge, UK, 1999.

[2] Ehrlenspiel, K. and Dylla, N. "Experimental Investigation of Designers' Thinking Methods and Design Procedures", Journal of Engineering Design, Vol. 4, No. 3, 1993, pp. 201 – 212.

[3] Fricke, G. "Successful Individual Approaches in Engineering Design", Research in Engineering Design, Vol. 8, No. 3, 1996, pp. 151 – 165.

[4] Günther, J. and Ehrlenspiel, K. "How Do Designers From Practice Design?", Designers: The Key to Successful Product Development, Editors: E. Frankenberger, P. Badke-Schaub and H. Birkhofer, Springer-Verlag, London, UK, 1998, pp. 85 – 97.

[5] Morgenstern, S. and Knaab, H. "A Descriptive Analysis of Requirements Identification Processes", Research report, University of Cambridge, UK, 1996.

[6] Newstetter, W. C. "Of Green Monkeys and Failed Affordances: A Case Study of a Mechanical Engineering Design Course", Research in Engineering Design, Vol. 10, No. 2, 1998, pp. 118 – 128.

[7] Badke-Schaub, P., Wallmeier, S., Dörner, D. "Training for Designers: A Way to Reflect Design Processes and Cope with Critical Situations in order to Increase Efficiency", Proc. ICED '99, Munich, 1999, Vol. 1, pp. 205 – 210.

[8] Schön, D. "The Reflective Practitioner: How Professionals Think in Action", Basic Books, New York, 1983.

[9] Argyris, C. "Knowledge for Action", Jossey-Bass, San Francisco, 1993.

[10] Nidamarthi, S. "Evaluation Report on Requirements-based Support Tools for The Designer", Technical Report, CUED/C-EDC/TR49, Engineering Design Centre, University of Cambridge, UK, 1997.

[11] Blessing, L. T. M. "A Process-Based Approach to Computer-Supported Engineering Design", PhD Thesis, University of Twente, the Netherlands, 1994.

[12] Pahl, G. and Beitz, W. "Engineering Design: A Systematic Approach", Springer-Verlag, London, UK, 1996.

Dr. Srinivas Nidamarthi
ABB Corporate Research
Wallstadter Str. 59, 68526 Ladenburg, Germany.
Tel: +49 6203 716171, Fax: +49 6203 716253, E-mail: srinivas.nidamarthi@de.abb.com

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DESIGN, SCIENCE, AND THE PROBLEMS OF UNDERDETERMINED PROBLEMS

K Dorst

Keywords: Design methodology, philosophy of science , problem solving

1 Introduction

Design processes can be seen as a special kind of very complex creative problem solving processes. Their complexity lies partly in the fact that, in designing, large amounts of information have to be processed and creatively combined in elegant and simple solutions [1]. And something is 'not quite right' with design problems: they have been characterized as 'ill-defined' [2], or even 'wicked'[3] problems. They have no definitive formulation, they are open to many different interpetations, etc (see Rittel and Webber for the whole dismal list...). Philosophers would say that these problems are underdetermined [4].
But what does it really mean, when a problem is underdetermined? What repercussions does the underdetermined nature of a problem have on the problem solving process? In this paper, we will explore how the different paradigms in design methodology have dealt with the issue of underdetermination in design problem solving, and consider how we can start to model the design problem solving proces to more closely reflect the underdetermined nature of the problems being solved.

We will first have to define underdetermination, and examine the ways in which Design Methodology, in the description of design problem solving processes, has (implicitly or explicitly) dealt with the underdetermined nature of design problems (§2). Then we will discuss some of the issues that have arisen in this inventarisation, and explore how the study of underdetermination in the 'context of discovery' of scientific theories could strengthen our effort (§3). In the conclusion we will introduce a substantial and unique international research programme that is currently being set up to deal with the questions we have outlined in this paper.

2 Underdetermination in Design Methodology

Design activities entail the reasoning from a set of requirements and intentions to a new bit of reality, consisting of a (physical) structure and an intended use. This process of reasoning is abductive: there is no closed pattern of reasoning to connect the requirements and intentions with a form and mode of use ([2],[3],[5],[6]). Design problems are underdetermined in the sense that the set of requirements and intentions can never be stated in such a way that the creation of its solution can be a deductive process. In this paper we will focus on two different ways in which a

design problem is underdetermined: (1) a functional description can never be complete and (2) 'function' and 'form' belong to different conceptual worlds [7].
These two aspects of underdetermination alone may lead to the feeling that the solution of design problems must be well nigh impossible. But despite the fact that design problems are underdetermined, designers in practice DO somehow overcome this underdetermination of design problems in their design processes.
We will first explore how this design problem solving process has been modeled in the two paradigms of design methodology, and focus on the ways in which these two aspects of underdetermination are dealt with in those models.

2.1 The rational problem solving theory of design

The relation between 'function' and 'form', and the resulting reasoning patterns that form the basis of designing have always been one of the core subjects of Design Methodology.
In the early sixties the founding father of Design Methodology, Christopher Alexander, described design as a process of *translating* function into form. Design problems were taken to be theoretically underdetermined, but not to the extent that 'the best' solution could not be determined in a rational process. This led to attempts to completely rationalise and even automate the solution of design problems [8]. But the theoretical models from those times were criticised for being overly rational, weak in the description of design processes, and completely impractical.

The criticism was most strongly heard in Architecture, leading to a development in which Architectural Design Methodology concentrated instead on the ill-structuredness [3] and creative aspects of architectural design activities [9]. In Engineering this same criticism led to attempts to incorporate more structural knowledge of designers and design problems into the existing rational phase models of the design process. The theories of Simon [10] provided a framework for this. Simon forged a link between the existing design methodology and the problem-solving theories from computer science and psychology. Simon's theories provided a model to describe designers and design problems within the paradigm of technical rationality, enormously extending the scope of design studies. They also finally provided a sound, rigorous basis for much of the existing knowledge in design methodology. In Simon's paradigm design is seen as a ***rational problem solving*** process, and this view has become the dominant influence shaping prescriptive and descriptive design methodology ever since.
According to Simon, (design) problem solving can be described as

> *'..... the search for a solution through a vast maze of possibilities (within the problem space)... Successful problem solving involves searching the maze selectively and reducing it to manageable solutions.'* [10]

These search processes have been studied through protocol analysis of subjects solving chess- and cryptarithmetic problems. They can be displayed and analysed in 'problem behaviour graphs' [11]. This approach has been developed in domains where problem solving involves mainly deductive and inductive reasoning. Simon's key contribution to design methodology was to state that the productive thought of design could be captured in the same positivistic framework. If this theory is valid for design, then design problem solving also takes place within a problem space that is structured by the structure of the task environment, which in its turn determines the 'programs' (strategies or methods) that can be used for creating a design. The core 'design' problem-solving process would be the same as in other kinds of problem solving.

In a later paper Simon [2] addressed some of the difficulties that arose in applying the rational problem solving approach to practical design by defining design problems as 'ill-structured problems'. Ill-structured problems are to be tackled in an 'immediate problem space'. This is a part of the total problem space, which is deemed too large, ill-structured and ill-defined to be described. The immediate problem space is addressed and put together by an (unspecified) 'noticing and evoking mechanism'.
These adaptations of general problem solving theory never solved the problems design methodologists had in using problem solving as a framework for describing design in practice . The basic problem might be that Simon never developed a theory for the solving of underdetermined problems as such: in his modelling of design the 'ill-structured' design problems have to be turned into well-structured ones (by a largely unspecified mechanism), and then solved.

2.2 Design as Reflective Practice

The difficulties faced by design methodologists in using this rational problem solving paradigm to provide a framework for the detailed description of design process lead to a search for alternative models of designing, supported by the abundance of empirical studies into design in the 80's and 90's.
Foremost among these was Donald Schön's 'primer' for a radically different paradigm of design, in which designing was modelled design as a process of *reflective practice* [12]. In this constructionist theory design is effectively described as a process of performing experimental actions ('moves') within a coherent framework (a 'frame'), and reflecting on the results of these actions. This reflection can then lead to contentment, to the execution of further moves within the same frame or to the reframing of the design problem. Design is thus not modeled as a linear search process; the learning loops ('iterations' in rational problem solving terms) are within this paradigm seen as the vital process through which design progresses.
But in spite of the fact that Schön's theory sparks immediate recognition in designers, it has had very little impact in Design Methodology. The basic concepts Schön uses are just too vague, and it has proven difficult to develop methods for designers based on this theory [13]. The theory of reflective practice does not provide us with enough clarity and structure to describe of the way in which designers tackle the underdetermined nature of design. Both kinds of underdetermination (the incompleteness of the functional descriptions and the conceptual rift between 'function' and 'form') are absorbed in the notoriously vague notion of 'frame': these frames' are supposed to be determined enough to allow sensible experimentation, and they are supposed to be 'views on the problem that imply a possible solution'.
The description Schön gives of the way designers deal with frames, the dynamics of choosing a temporary frame of reference and testing it, could however be useful in the description of design as the creation of solutions to underdetermined problems.

The empirical studies executed in the 80s and 90s however demonstrated that designers use a number of different reasoning patterns and tools (other than just 'framing') that seem to help them to overcome the underdetermined nature of design problems. In empirical design methodology we can find descriptions of the use of analogies, metaphors, the adoption of existing design solutions, the use of temporary representations of a design problem and solution (like the splitting up of a design problem in functional parts), the supplementing of a set of functions with own criteria – thereby heightening the degree of determinedness of the problem, etc etc...([14],[15],[16]). To simply label all of these actions as 'framing'is just not enough. In the next section we will discuss some further developments in the modelling of design, that might lead to eventually capturing the ways in which designers treat their underdetermined design problems.

2.3 Design as Co-evolution

The Rational Problem Solving paradigm can still be considered the most fruitful way we have to model design processes, although it has its limitations. The modelling opf design as a rational problem soilving process has been further developed by the 'design-AI' movement, and it provides a rigorous and logical basis for the thinking behind many computational models of design [17]. In their paper *'Co-evolution of Problem and Solution Spaces in Creative Design'*, Cross and Dorst [6] have attempted to integrate these developments in computational modelling of design with the Reflective Practice paradigm, on the basis of an in-depth analysis of empirical data.

They argue that empirical research has led to the realisation that creative design is not a matter of first fixing the problem and then searching for a satisfactory solution concept (like in Simon's theories). Instead, it seems more to be a matter of developing and refining together both the formulation of the problem and ideas for its solution, with constant iteration of analysis, synthesis and evaluation processes between the two 'spaces'. In creative design, the designer is seeking to generate a matching problem-solution pair. Cross [18] suggested that the creative event is not so much a 'creative leap' from problem to solution as the building of a 'bridge' between the problem space and the solution space by the identification of a key concept. The interdependent development of problem and solution spaces, and the recognition of a satisfactory bridging concept for a problem-solution pair is considered one of the features that characterise creative design as *exploration* rather than search [17].

The computational model of design that Maher et al [19] have developed is based on such a 'co-evolution' of the problem space and the solution space. The problem (behaviour) space and the solution (structure) space co-evolve together, with interchange of information between the two spaces (see Figure 1). The Maher co-evolution model functions as a regulative cycle in the (computer-simulated) design process.

When we compare this to the empirical sudies of design, this seems to model the designers' behaviour very well, that is, their behaviour after the conception of the original idea. It models the creative design process, but not the creative 'leap' or 'bridge building'. It fails to capture both the initial chunking of information and the selection of this chunk. As such it does shed some light on the way designers could tackle the problem of incompleteness of functional descriptions (by the evolution of the problem space), but it fails to describe how the conceptual rift between 'function'and 'form'is bridged.

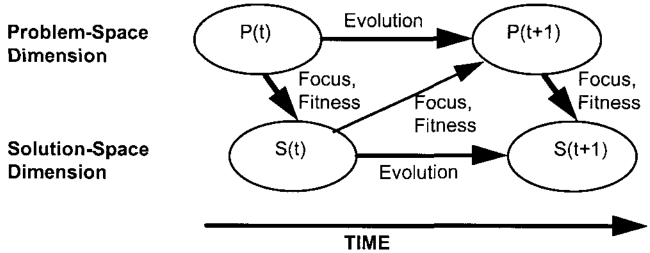


Figure 1. The co-evolution model of Maher et al.

3 Discussion

It will be clear from this quick overview of underdetermination in design methodology that we are still at the very beginning of developing a model of design as the solving of underdetermined problems.
The above model of design as the co-evolution of problem and solution spaces does offer some hope for mapping the ways in which designers deal with the incompleteness of the functional description in the design problem. By now design methodologists have gathered a solid basis of empirical data that can be studied in this light. A lot can be learned about the ways designers deal with the incompleteness of the functional descriptions in design problems by 'harvesting' this from the body of design methodology literature. We will do so in the research project that is outlined in the next section.

The second aspect of underdetermination, the conceptual rift between 'function'and 'form', will be much harder to clarify. The conceptual discussion about 'function'and 'form' has been raging within design methodology for quite some time, without reaching closure. Design is a notoriously difficult subject to study conceptually, because it is so very rooted in 'the real world': design problems are contex dependent, and part of the knowledge designer use is of a tacit or implict nature. Design decisions often seem to be based on design experience that is hard to capture... Design problems are 'wicked' [3], and abstracting from them in a controlled manner is difficult.
To tackle this it might be helpful to step back from design for a while and study the ways in which the philosophy of science (as specialists in the clarification of concepts) has dealt with the analoguous problem of the creation of scientific hypotheses and theories. The problems a scientist faces in developing a new theory are underdetermined, in the sense that a theory has to perform a certain 'function' within the field of study, and the scientist has to come up with a 'form'that will fullifill that need. The possible advantages in studying the underdetermination of research problems lies in their often well-contained, abstract en explicit nature (this in clear opposition to design...). Duhem and Quine describe a kind of 'Holistic Underdetermination', in which hypotheses –like design concepts- are considered as 'webs of belief' [21]. This closely resembles the matching problem-solution pairs decribed in §2.3, but puts them in a much wider perspective.
The possible empirical basis for these philosophical reflections lies in the adjacent field of the methodology of science. A real disadvantage is that there is little empirical data available about

the way scientists deal with their problems. The dominant modelling of such processes still resembles the rational problem solving approach that is also dominant in design. As a result of this, research methodology tends to concentrate on the falsification and further development of theories, and not so much on their conception. The attention is focussed on the 'context of justification', the 'context of discovery' is largely neglected [20]. If the 'context of discovery' is described at all, its problems are often decribed as those of choice between alternatives, even though the scientists whose processes are described in such a way do not agree:

> *... 'Scientists repeatedly point out that the problem generally facing them is finding even one hypothesis that will fit with the available evidence...' [21]*

Some case descriptions can however be found, for instance in studies by Koestler [22], examining parallels between scientific invention and artisitic creation, and by Gardner [23], who is interested in pinpointing the mode of creativity involved.

Of course we have to be careful in transferring their conclusions to the field of design, but the concepts of 'function', 'form' and 'underdetermination' are abstract enough to make a fruitful inferences about design problems, too. And as design methodologists, we are in an ideal position to make this link. We know both worlds: we ARE scientists, in the subject of designing.

4 Conclusion

We are now setting up an extensive research project in which the problem of underdetermination in design is studied in detail. This will entail the following four steps:

- An analysis of design strategies

The analysis of empirical data of designers at work will focus on the question how the underdetermination of design problems is perceived and dealt with in design practice.
A literature survey has already yielded descriptions of various strategies designers use. All of these strategies will be described in detail, and their structure will be analysed.

- The development of a conceptual framework

The modelling of designing as the creation of solutions to underdetermined problems will focus on the development of a conceptual framework, that should enhance and refine our understanding of design. To develop this conceptual framework we will use the empirical data on the strategies designer's use as a starting point, and base ourselves on the existing conceptual frameworks in design methodology and the philosophy of science.

- The development of a classification

This analysis of empirically found design strategies and the development of the conceptual framework will be done in close interaction with eachother, and with the conceptual analysis of kinds of underdetermination, leading to the development of a classification of underdetermined design problems.

- Conclusion: the creation of solutions to underdetermined design problems

Finally we need to develop an overview of the strategies for dealing with underdetermined problems. The fundamental insight that the analysis and conceptual framework have generated should lead to a framework for the better use of the existing strategies, and possibly serve as a basis for the development new tools and strategies for the creation of solutions to undertermined design problems.

This study of underdetermination is part of a research programme in which philosophers and design methodologists work together to develop a coherent conceptualisation of technical artifacts [7]. This is an ambitious four-year program set up by the Delft University of Technology and Eindhoven University of Technology in close cooperation with Virginia Tech, Buffalo and MIT. In this programme an extensive conceptual analysis will be made of the link between physical structure and the functions of artefacts. One of the ways to shed light on this connection is to study the process in which engineers connect function and structure in the design process. This project is a unique cooperation in the sense that philosophers and engineers will be working together on the fundamentals of the creation of technical artifacts, and that this meta-study is based on philosophical reflection, the theories of design and on empirical data.

Acknowledgements

The author would like to thank Nigel Cross for the chance to work with him on the theory of Co-evolution.

References

[1] Roozenburg, N.F.M., Eekels, J., Product Design: Fundamentals and Methods, Wiley, Chichester, 1995.

[2] Simon, H.A., The structure of ill-structured problems, Artificial Intelligence, 4, 181-201, 1973

[3] Rittel, H.W.J. and Webber, M.M., Dilemmas in a general theory of planning. In: Cross, N.G., (ed.) Developments in Design Methodology Wiley, Chichester, 1984

[4] Meijers, A.W.M., The relational Ontology of Technical Artifacts, In: Kroes, P.K. and Meijers A.W.M., The empirical turn in the philosophy of technology, Elsevier Science, Oxford, 2000

[5] Cross, N.G., A history of design methodology, In: de Vries, M., Cross, N.G., Grant, D., (Eds), Proceedings of the Nato advanced research workshop Eindhoven, 1992.

[6] Cross, N.G. and Dorst, K.: Co-evolution of Problem and Solution Spaces in Creative Design, in: Gero, J. and Maher, M.L., (eds.); Computational Models of Creative Design, Key Centre of Design Computing and Cognition, University of Sydney, Australia p. 243-262, 1999

[7] Kroes, P.A. and Meijers, A.W.M., The Dual Nature of Technical Artifacts, NWO project proposal, Delft University of Technology, 1999.

[8] Alexander, C., (1964) Notes on the synthesis of form, Harvard University press, Cambridge MA, 1964.

[9] Lawson, B., How designers think: the design process demystified (2nd ed) Butterworth, London, 1990.

[10] Simon, H.A., Sciences of the Artificial, The MIT Press, Cambrigde MA, 1992

[11] Newell, A. and Simon HA, Human Problem Solving , Prentice-Hall, EnglewoodCliffs, 1972

[12] Schön, D.A., The Reflective Practitioner Basic Books, New York, 1983

[13] Roozenburg, N.F.M. and Dorst, C.H., Describing design as a reflective practice: observations on Schön's theory of practice, In: Frankenberger, E., Badke-Schaub, P., Birkhofer, H., (Eds) Designers: The key to succesful product development, Springer, London, 1998.

[14] Dylla, N., Denk- und Handlungsablaüfe beim Konstruieren Hanser, München, 1991.

[15] Frankenberger, E. and Badke-Schaub, P., Modelling design processes in industry – empirical investigations of design work in practice, In: Akin, O. and Saglamer, G. (Eds), Proceedings of DMD'96, Istanbul, 1996.

[16] Dorst, C.H., Describing Design - a Comparison of Paradigms, thesis TUDelft, 1997

[17] Gero, J.S., Towards a model of exploration in computer-aided design, In: Gero, J.S., Tyugu, E., (Eds) Formal Design Methods for CAD, North Holland, Amsterdam, 1994

[18] Cross, N.G., Modelling the Creative Leap, paper submitted to the Third International Round-Table Conference on Computational Models of Creative Design, Heron Island, Australia, 3-7 december 1995

[19] Maher, M.L., Poon, J., Boulanger, S., Formalising design exploration as co-evolution: a combined gene approach, in: Gero, J.S., Sudweeks, F., (eds) Advances in Formal Design Methods for CAD, Chapman and Hall, London, 1996.

[20] Kuhn, T.S., The structure of scientific revolutions University of Chicago Press, Chicago IL, 1966.

[21] Laudan L., Underdetermination, In: Routledge Encyclopedia of Philosophy, Routledge, London, 1998

[22] Koestler, A., The Act of Creation Picador, London, 1964.

[23] Gardner, H., Creating Minds Basic Books, New York, 1993.

Author:
Dr Kees Dorst, Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, The Netherlands, Tel: +31402475535, email: C.H.Dorst@TM.TUe.nl or Kees_Dorst@hotmail.com

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

REFLECTIVE PRACTICE AS A WAY OF OBSERVING AND ANALYSING PRODUCT DESIGN TEAMS

R C Valkenburg

Keywords: Empirical study, reflective practice, design teams.

1 Introduction

In this paper and the presentation we will present the results and conclusions of a PhD-research project on team designing [1]. In order to improve team designing, we have to understand it. In order to understand it we must be able to describe it. Therefore the research project had two goals. The first goal was to develop and evaluate a description method for describing team design activities. The second goal was to understand the nature of team designing by exploring design teams at work.

The first research goal focussed on a method for studying team designing. We adopted the well-known theory of reflective practice, introduced by Donald Schön [2, 3], to develop a method of description for team design activities. A model for the reflective practice activities was developed, together with a notation system to process empirical data. With this method four team design projects were described in two separate empirical studies. The method of description was evaluated, using both a multi-coder approach and an arbitration discussion on the data.

The second research goal focussed on the insights to be gained. We observed team designing from the reflective practice viewpoint in order to gain an understanding of what teamwork is really about. In the empirical studies two key areas of interest were investigated. The first area was the use of the four different reflective practice activities within the teams (*naming*, *framing*, *moving*, and *reflecting*). The second area was the interaction between the reflective practice activities within the teams, to explore patterns in the team's reflective practice strategies.

First we will provide some of the backgrounds of this research project. We will relate this study to other studies and theories in design. We will also introduce the empirical studies of this research. This will all be described in section 2. In the third section we will then discuss the results of reflective practice as a research method. In the fourth section we will discuss the insights into the nature of team designing that we gained from the empirical studies. We will conclude with a discussion on the possibilities and limitations of the research method and recommendations for future applications in section 5.

2 The research project

2.1 Previous research on team designing

A review of literature on design research shows that few researchers deny that working in and as part of a team is a crucial part of design, only very few have taken a serious look at what that involves. Outside these few exceptions, researchers largely persist in doing studies of individual designers and building tools for them, without understanding the context of the use of these tools.

Studies that do look at co-operation in design reveal more and more insights in the complexity of the design activity (e.g. [4, 5]). They indicate that 'context variables' influence the performance of the design process. These empirical studies indicate the problems that design teams face in practice. They also show that in studying team designing, research should no longer focus merely on the design process. They rather integrate characteristics of the designer (or design team) and of the design task.

Some explorative studies on design as a social process have taken place (e.g. [6, 7]). These studies provide detailed descriptions of cases of team designing and examples from team protocols. These explorations also reveal the difficulties in doing this type of explorative research. The studies end in case descriptions of 'anecdotes' and lack generalisability of findings and conclusions.

It is therefore our view that the study of team designing could benefit from a structured approach. If this research project can provide a good way of looking at team designing to support analysis of team behaviour, it can help the progress of the research in this field. We also want to found this approach in a general theory of design to integrate team design aspects into the current know-how on design. Because the theory should not only focus on the design process, but integrate aspects of the designer and the design content, we have chosen to use Schön's theory of reflective practice [3, 4].

2.2 The empirical studies

In this research project two empirical studies were executed. In the first study we observed two, multidisciplinary product design teams participating in a design competition. The teams, 'The Delft Pitchbulls' and 'Tecc', were located in one room for two days, creating a conceptual design. Their task was to design a remote controlled product that could transfer small balls from a competition table into a basket (an extensive description of this empirical study is published elsewhere [8]).

For the second empirical study we chose data, well known to the design research community; the team protocols of the Delft Protocol Workshop [9]. These data provide the observations of two design teams, 'Fran, George and Harold' and 'Ivan, John and Kerry', both composed of three experienced designers, working on a conceptual design task for two hours. Their task was to design a fastening device to fasten a backpack onto a mountain bike.

The observations of all four teams were coded and described according to the model of reflective practice. These descriptions served as input for the analysis of team behaviour and communication.

3 Reflective practice as a research model for describing design

Schön's work has inspired a great deal of research in design [10, 11, 12]. Also much criticism has been levelled, in trying to fully understand his ideas [13]. In our analysis of Schön's work, we also indicated several shortcomings. We noticed the inadequate empirical evidence of reflection-in-action, the lack of precise terminology, the difficult task of making the theory of reflective practice operationable, and the paradox within the theory between generalisation and specification. We will now summarise and discuss the key findings on the first research goal of this study. We will do so by addressing the criticism on Schön's work.

3.1 Empirical evidence for Schön's ideas

Schön's ideas are based upon empirical studies of professionals. His examples don't show designers at work, performing reflection-*in*-action, but rather tutors *reflecting* on student's design work. These examples therefore show reflection-*on*-action. Our data on the other hand, concern design teams at work. We have shown that reflection-*in*-action occurs in some teams, while other teams work in a different way.

In the studies we identified reflective practices in two team design projects. The teams 'Tecc' and 'Ivan, John and Kerry' design in a reflective practice manner. They *name* issues in the design task that require attention, and explicitly organise and choose these *names*. They explicitly *frame* their view on the problem. This *framing* provides coherence and sets a direction for action. Within this context they make *moves* to solve the design problem. As they spin their web of *moves*, their attitude towards the design task changes. Every now and then they *reflect* on these changes, deciding whether their *moves* are aimed in the desired direction and whether their current *frame* is satisfactory as guidance for further activities. By stating the directions they are going to take, they harmonise not only what they are doing in procedures but also what the content of that activity is. For instance when they decide to generate ideas, they explicitly set the direction for these ideas, such as 'ideas for bridging the gap between the competition table and the basket' (Tecc) or 'ideas for positions of the backpack on the bike' (Ivan, John and Kerry).

In the studies we also identified other ways of working within two design projects. Team 'The Delft Pitchbulls' *name* important issues in the design task that require attention. Within the project they continue to attend to all of them. They divide their attention over all issues and interrupt discussions that aim to focus on a subject. They are searching for solutions for these issues, but without generating many alternatives. The solutions they decide upon are not compared to others, or evaluated extensively. Team 'Fran, George and Harold' also deal with the design problem as a set of separate design issues. They start by looking for a solution for the first issue and then connect all other issues one by one as they progress into the project. This seems to work well in this project, although the concept in the end is not optimised as an integrated solution. It is only evaluated and optimised on the separate design issues. It is difficult to point the finger at what exactly is happening within these teams. A feature of the communication within 'The Delft Pitchbulls' team is that every time one of the team members tried to focus their attention, another one brought up a different issue that is 'very important' and the team's discussion changed direction. Working in this way they do not effectively deal with any issue. 'Fran, George and Harold', on the other hand, do work together purposefully. Bit by bit they construct their design out of all the separate issues that they encounter in the design problem. It is notable that the team does not take a critical attitude towards their activities, nor towards the design result.

The results from the empirical study show that not all designing is done in a reflective practice way. This is remarkable, because, according to Schön, all designing is reflective practice. It is also remarkable that both teams that do use reflective practice make an extensive evaluation of their end results at the conclusion of their design projects. 'Tecc' makes a list of the weak and strong points of their design, which is regularly quoted in their discussion with the expert judges. 'Ivan, John and Kerry' extensively compare their design with a list of requirements that they themselves made. 'Fran, George and Harold' as well as 'The Delft Pitchbulls' don't seem to have this critical attitude towards their design activities.

3.2 The reflective practice terminology

Over the years Schön's work repeats a lot in explaining and illustrating his basic ideas, but the explanations do not progress far in providing more refined ideas or definitions. Sometimes Schön even fails to hold on to a consistent approach to his own concepts. Using Schön's defective descriptions as a starting point, we developed a notation system for reflective practice. In the explorative empirical study we encountered problems that we tried to solve. These problems concern the precise definitions of the notions as well as the interpretations the researchers have to decide upon.

For example, it is difficult to unambiguously *define* what a *frame* is. However, the discussion among the coders indicated that *frames* are not difficult to *identify*. It can easily be agreed upon what the *frames* are that the teams used. Objectively defining *frames* is difficult, because *frames* depend on the specific situation and the design content in this situation. The *frame* is not only determined by this design content, but also by the way that the team deals with this content. Therefore we rather identify *framing* than the *frames*. For example, there is a moment in the team projects where the teams 'Ivan, John and Kerry' and 'Fran, George and Harold' are doing similar things in the design project. Both teams *name* different issues they have to consider in the design project. 'Ivan, John and Kerry' *name* 'positions', 'joining techniques', and 'materials'. 'Fran, George and Harold' *name* 'the bracket to the bike' and 'the frame to the bracket'. After this *naming*-activity 'Ivan, John and Kerry' continue by *framing* 'joining technology, attaching pack to rack' and focus on the exploration of that *frame*. 'Fran, George and Harold' continue by checking the idea they generated earlier and connecting an attachment solution to it. In this example both teams *name* issues in the design task (even comparable issues). In one case this *naming* leads to *framing* of the problem, in the other case it leads to a *move*.

We found a way of *identifying* reflective practice. By focussing on design activities, we can picture the interaction between the design activity (process), the design task (content) and the designer(s). At the same time, this interaction hampers the operationalising of the constructs within reflective practice. Defining general notions that are content- and activity- dependent is per definition impossible. The way of identifying these notions did provide clarity. Clarity on the reflective practice within design teams, but also on the reflective practice theory itself.

3.3 Making reflective practice work

While there has been much theoretical speculation about Schön's notions, surprisingly few researchers have set out to operationalise them. We made reflective practice work within the boundaries of its beliefs. The reflective practice way is, up till now, still the only way that is able to describe the developing design content within projects, while linking this content to the design process and designers. For studying team designing it gives a clear picture of the developing design progression.

We already indicated that this interaction also introduces problems. In this research project we developed a method to deal with these problems. We evaluated the method of description by processing the data with three different coders. Then we set up an arbitration discussion between the coders to identify and deal with the agreements and disagreements between the coders. This evaluation of the method of descriptions indicated a good reliability of the data processing among three coders. The initial disagreements in data processing among coders proved to be systematic, for example the way that a coder interpreted the coding of *frames*. By discussing these systematic differences and the arguments behind them in relation with the data, always led to shared understanding between the three coders. Eventually, all disagreements could be dealt with.

Qualitative research involves subjective interpretation by the researchers. These interpretations will always differ, inherent to the researcher's interpretation of the world. Researchers need to find ways to deal with this interpretation in the development of research methods. They should not try to ignore or avoid subjective interpretation, but reliably include it in research. The method of multi-coding, combined with an arbitration discussion, proved to be a useful evaluation method to deal with the multiple, subjective interpretation of the coders.

3.4 The paradox between generalisation and specification

To Schön, every designer and every design task is unique, but in the end he still describes the reflective practice process as a general view on designing. The uniqueness that Schön addresses is embedded in the design task from the viewpoint of the designer. In this research project we attempted to exclude the uniqueness from the viewpoint of the researcher. Different observers can identify the same issues in the design task, which makes reflective practice, as a research method, less individual and subjective.

Concerning the uniqueness of the situation and design task, we cannot make statements on the basis of four design projects. Observing two sets of two projects does raise questions that, we think, can be dealt with using the reflective practice approach to research.

4 Reflective practice as a way of looking at design

In this section we will summarise and discuss the key findings of the empirical studies concerning the reflective practices of the teams. We will look at the use of the four different reflective practice activities within the teams. The data shows patterns in the interrelationship of the four activities. Therefore we will also indicate the teams' reflective practice strategies.

4.1 The reflective practice activities

The reflective practice theory provided a starting point for the research project. From Schön's work we indicated that designers work by *naming* the relevant factors in the design situation, *framing* this situation in a certain way, making (experimental) *moves* toward a solution and *reflecting* on those *moves*. We looked at reflective practice in different situations, and we will discuss how the teams deal with the four reflective practice activities.

'Tecc' starts the design project by using *names* to divide the design task. Then the team chooses one of the *names* and makes that their *frame* for the moment. During the rest of the project 'Tecc' keeps referring to earlier *names* as a stepping stone to create new *frames*.

'Tecc' uses six different *frames* sequentially during the design project. These *frames* originate from the different functions they have *named* early in the process, but also from insights that the team gains while designing. The actual designing takes place within *moves*; when the team is really handling and changing the design content. Once the context for *moving* is set, by stating a *frame*, the *moves* work towards exploring the design task and challenging the chosen *frame*. 'Tecc' *reflects* throughout the design project. 'Tecc's' *reflection* is initiated by evaluating what the team is doing in relation to the *frame* they are working in. A *reflection* can also be initiated by the design content. These *reflection* moments always occur in relation to the current *frame* and the team's progress in the design project. 'Ivan, John and Kerry' also explicitly use *frames* to approach the design task. Half an hour into the design assignment the team is generating (*naming*) *frames* for the design task; what they call 'classifications'. Then they explore these different viewpoints on the design problem. They do this by exploring different *frames* and generating many alternatives, before integrating them into a total solution. 'Ivan, John and Kerry' extensively explore four problem *frames* before thinking on an integrated solution. During the design project, in some moments of *reflection* 'Ivan, John and Kerry' *reflect* on what they are doing at that specific moment. There are also *reflection* moments that are aimed for a broader perspective, where the team not only considers the current activity, but also the current *frame*. These *reflections* not only lead to new *moves*, but also to the reconsideration of the context (*reframe*).

In the analysis of the teams' reflective practice we indicated that the teams *name* different issues in the design task. The team members pass over these *names* to each other and then decide upon a focus within the design problem. They do so by *framing* a view on the problem or the solution. The actual designing takes place within *moves*. *Moves* outside *frames* appeared to lack a shared goal. This lack makes it difficult to aim the team's discussion. *Frames* are important elements for they improve the team's *moves*, and therefore the team's designing. The development of *frames*, both stating a *frame* and modifying or rejecting it again, is important in building an understanding of the design task and its solution. This is guided by the mechanism of *framing*, *reflecting* and *reframing*. Therefore, *reflection* is crucial in designing. Only by *reflecting* on its design activities a team of designers can rationally make a decision to start a new activity or to *reframe* their approach towards the problem.

4.2 Reflective practice strategies

All four activities can be found in the observed team design projects. All observed team design activities can be indicated as one of the four reflective practice activities. Applying the detailed mechanism of *naming*, *framing*, *moving* and *reflecting* onto a design project also reveals patterns and large-scale strategies. Schön completely missed this for he described reflective practice as a locally controlled activity. The teams' descriptions are not just a flow of separate activities; patterns occur in the reflective practice of the teams.

It is interesting to notice that *frames* mostly end with a *reflection* moment, where the team closes the current series of activities and moves on to the next. If necessary by *reframing* the problem. Such a mechanism indicates an underlying learning process, where the team creates an understanding of what they are doing and how the design content is developing. The exploration of the *frame* is concluded by *reflection* and so is the learning cycle within.

Not only *frames* help in this 'learning process'. Also conscious *reflection* on what the team is doing and where it gets them in solving the design problem, builds this knowledge and understanding of the design task. Regular *reflecting* in the project seems to be a precondition for reflective practice. We already indicated the pattern of *reflection* moments ending *frames*.

In the empirical study we also identified *reflection* moments in a sequence of *moves*. Every now and then, the teams critically look at what they are doing, and then decide how to proceed.

Another pattern in the reflective practices of the teams is the development of *frames* throughout the project. Teams Tecc and Ivan, John and Kerry use different views on the problem to explore. These views are generated early in the design project (*naming*). Following the development of these *frames* in the project traces the progression of the design content, as well as the understanding the team builds on the design task.

The *frames* we identified in the empirical study are of a very practical and concrete nature. Tecc views the design problem as 'shooting the balls', 'getting the balls into the basket', and 'simple and integrated solution'. The first two *frames* concern a part of the design problem, whereas the third *frame* indicates a view on the problem. Ivan, John and Kerry explore three views of the design problem: 'positions', 'joining techniques' (divided in joining techniques 'pack to rack' and 'rack to bike'), and 'materials'. Although these *frames* may seem to be parts of the design problem, the team is really generating ideas for the whole solution, but from the viewpoint of the set *frame*. The *frames* derive from an extensive analysis on the design problem and are changed by insights the team gains in the design project.

5 Conclusions

Concluding, we can say that the research project provides us with a detailed empirical investigation on examples of reflection-*in*-action. Evaluating the research method we learned to deal with Schön's heritage and attempted to develop a way to deal with subjective interpretation in qualitative research. We also extended reflective practice to design *projects*. Schön describes reflective practice as a local activity. Applying the method of reflective practice to an entire project also shows different patterns, which Schön does not deal with at all. It provides a good picture of what the teams are doing and how the design content progresses.

We must also admit that some accomplishments turned out differently than we expected them to. We could not completely validate the research method, in the sense that every coder would get the same results from data processing. We did, however, develop a way for dealing with differences in data processing, which we think, is of more use in this type of research. We can not provide clear definitions of the notions within the theory of reflective practice, but we did find a way to identify them in team designing. We can not prove the existence of reflective practice, rather raise the question whether reflective practice is always "good" designing.

Looking at designing in-depth, the way we did in this research project, provides too few data to generalise conclusions for teams. A good exploration on the teams' activities does provide a way of clarifying questions for further research.

This research project is only a first step in the investigation of team designing in the real world. Reflective practice provides us with insights into issues in designing that have not been attended to earlier. It provides us with a way of communicating, as well as a language to communicate, about issues in design that we were not able to 'capture' before. Intuitively accepting that working in a reflective practice manner delivers better results and learning processes, we cane formulate recommendations on the improvement of team designing in a reflective practice manner.

References

[1] Valkenburg, R.C., "The Reflective Practice in product design teams", PhD Thesis, Delft University of Technology, 2000.

[2] Schön, D.A., "The reflective practitioner", Basic Books, New York, 1983.

[3] Schön, D.A., "Educating the reflective practitioner", Basic Books, New York, 1987.

[4] Hales, C., "Analysis of the engineering design process in an industrial context", PhD thesis, University of Cambridge, Grants Hill Publications, Hampshire, 1987.

[5] Frankenberger, E. and Badke-Schaub, P., "Modelling design processes in industry - empirical investigations of design work in practice", The Journal of Automation in Construction, volume 7, number 2-3, 1998.

[6] Cross, N.G. and Clayburn Cross, A., "Observations of teamwork and social processes in design". In: Cross, N.G., Christiaans, H.H.C.M. and Dorst, C.H. (Eds): Analysing design activity. Wiley, Chichester, 1996.

[7] Brereton, M.F., Canon, D.M., Mabogunje, A. and Leifer, L., "Collaboration in engineering design teams: mediating design progress through social interaction". In: Cross, N.G., Christiaans, H.H.C.M. and Dorst, C.H. (Eds): Analysing design activity. Wiley, Chichester, 1996.

[8] Valkenburg, R and Dorst, K., "The reflective practice of teams", Design Studies, volume 19, number 3, 1998.

[9] Cross, N.G., Christiaans, H.H.C.M. and Dorst, C.H. (Eds): Analysing design activity. Wiley, Chichester, 1996.

[10] Dorst, K., "Describing design - a comparison of paradigms". PhD thesis, Delft University of Technology, 1997.

[11] McDonnell, J., "Resolving design as reflective exploration with formal description of the design situation". In: Akin, O, and G Saglamer, (Eds): Proceedings of the first conference on Descriptive Models of Design *(DMD'96)*, Istanbul, 1996.

[12] Pereira, M., "My reflective practice as research", Teaching in Higher Education, volume 4, number 3, 1999.

[13] Roozenburg, N.F.M. and Dorst, C.H. "Describing design as a reflective practice: observations on Schön's theory of practice". In: Frankenberger E, P Badke-Schaub and H Birkhofer (Eds): Designers: The key to successful product development. Springer, London, 1998.

Dr. ir. Rianne Valkenburg
Delft University of Technology, School of Industrial Design Engineering, Department of Product Innovation & Management, Jaffalaan 9, 2628 BX Delft, The Netherlands.
t: +31 (0)15 2783068, f: +31 (0)15 2787662, R.Valkenburg@io.tudelft.nl

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

SOME USEFUL METAPHORS FOR THE DESIGNING ENGINEER

P M Winkelman and O R Fauvel

Keywords: design philosophy, reflective practice, metaphor

1 Introduction

Much of the education and discussion of engineering design today is centred around *what* kind of designing engineers do and *how* they do it. The question of *what* is usually answered with reference to technology and the practical situations for which it might be appropriate, and the question of *how* is linked to the intelligent application of science and mathematics. Questions regarding *why* engineers design what they do are seldom entertained as serious matters of academic or intellectual pursuit.

Perhaps part of the reluctance to speak at great length on the question of why has to do with the history of science within engineering. Within the scientific paradigm, one asserts that one is dealing with facts. The paradigm tacitly promotes the belief that these facts essentially speak for themselves. As we shift towards the question of *why*, we must deal with engineering values and the scientific paradigm can no longer remain front and centre. Other paradigms must necessarily enter to facilitate meaningful inquiry. Having had the virtues of science extolled from the beginning of their education, engineers tend to consider a step away from science as a retrograde step. Compared with more scientific pursuits, studies aimed at understanding the *why* of engineering are perceived to take on a lower profile and carry reduced prestige. (Perhaps the recently introduced requirement that accredited engineering programs in Canada [1] embody a more meaningful and extensive design experience is at least indirectly intended to redress this perception.)

We believe that the dismissal of the question *why* within the engineering context is cause for concern. Engineers are no longer expected nor encouraged to work in isolation. Their roles in society have become more complex, thus placing new demands on them to conduct themselves prudently in a world with a multiplicity of value systems. How can engineers be made aware of their values and underlying assumptions in a way that is meaningful to them? We propose to address this question through the application of metaphor.

This paper begins by examining metaphor and its use. The metaphors for the engineer are then used to illustrate how practices and values may be expressed and viewed in ways which can be useful to the designing engineer.

2 Metaphor

Once considered largely a method for literary embellishment, metaphor has been gaining greater recognition as being central to human understanding of the world [2]. In other words, we cannot simply say that we use metaphors to describe our observed reality, but metaphor actually creates our reality.

Although engineers may employ metaphor, they usually speak of it in terms of being a model or an analogy. For example, the flow of water in pipe is typically presented as being analogous to the flow of electricity. Such analogy is deemed to have pedagogical value insofar as it aids comprehension. Its weakness can be seen in that, in using the word analog rather than metaphor, we overlook the linguistic implications of the pedagogical tool.

The linguistic part of metaphor can perhaps be more readily recognized in the example of computer design. A defining metaphor might be A COMPUTER IS AN OFFICE. In this particular example, the usefulness of metaphor can be reflected in the many words that have become standard computer terminology. Words such as *file*, *desktop* and *trash* have been borrowed directly from office vocabulary. Furthermore, this example shows the important design implications of metaphor. First, the metaphor provides engineers and software personnel with a pre-existing framework of conceiving a computer and, by extension, a way of designing the system. Second, when the computer was first being introduced into offices, the application of the metaphor also proved helpful for office personnel to more readily understand the functionality of the computer. Thus, part of the computer was already known to office personnel even though they may not have worked with computers before. As a result, the computer became instantly more user-friendly through the intelligent application of metaphor.

The computer example also demonstrates an additional feature of metaphor: metaphor is both concise and descriptive. In this paper, we intend to exploit this feature further and extend the use of metaphor beyond that of a learning aid or a design tool. We will use metaphor as a simple means to allow engineers to better understand the assumptions, some tacit and some overt, that have become part of their profession as designers. We propose to highlight aspects of the practice of engineers with reference to three metaphors: the engineer is a gardener, the engineer is a map-drawer and the engineer is a bridge-builder. We chose this order to facilitate the development of our ideas. The first and last metaphors highlight the engineer's assumption of improvement, and the middle one draws attention to the historical and philosophical roots of engineering.

2.1 The engineer is a gardener

The profession of engineering has been compared to gardening [3]. Gardens are basically constructed of things taken directly from nature. They typically contain plants of various kinds which grow on various terrains, sometimes on rocks, sometimes in soil. The plants take their nutrients from the soil and depend on the rain and the sunshine. Like plants in the wild, plants in the garden grow, produce seed and eventually wither and die.

Although they are composed of natural things, gardens differ from nature in that the nature therein is *ordered*. Rather than sown randomly, plants are separated and placed in designated plots. The location of the various plant groups may arise from an effort to control light or moisture or perhaps to practise companion planting. Some plants, such as vines, may require support structures to promote growth. Pathways allow access to the whole garden from only a few points of entry. Irrigation may be used if rainfall is deemed insufficient, and heat lamps may be used in the fall to stave off an early frost. Furthermore, the food chain may be modified by the application of pesticides or herbicides. From the standpoint of the human intent of a garden, nature has been improved upon.

What nature is to the gardener, science is to engineering. Engineering looks to science to provide an accurate description of nature. These descriptions, often termed "laws", become the "seeds" or "plants" for the engineering product. If we define design as bringing order to disorder, then the imposed order of the garden refers to the design work of engineers. Engineers rearrange nature to produce a form that is more in tune with their specific purposes. By imposing their own sense of order, engineers strive to improve on the basic ingredients (laws and materials) from nature.

But why should an engineer or a gardener impose order? What constitutes an improvement in this context? We can answer these questions with reference, for example, to the measure of *production*. An improvement is generally associated with more output with respect to some input. By separating plants and placing them in rows, it is easier for the gardener to remove unwanted competition and harvest more easily. The harvest is further facilitated by locating pathways through the garden, improving accessibility. When conceived, this practice heralded the dawning of agriculture. Agriculture made it possible for humans to stay in one place without loss of food supply. Resulting settlements allowed civilizations to come into existence. Like the gardener, the engineer orders nature to spur production. Mass quantities make the production process easier. Like the paths within a garden, the production goals of engineering are heavily dependent on a good transportation system. And like the garden, not all outcomes can be perfectly predicted.

2.2 The engineer is a map-drawer

A map represents another way that engineers order their world. In this case, the order is largely spatial. The map metaphor is quite rich in that it symbolizes a variety of engineering assumptions.

The map and mathematics

A map allows a coordinate to be assigned to every location in the real world represented on the map [2]. This mapping is a reflection of the contribution to engineering of Rene Descartes in the form of his Cartesian coordinate system. Descartes also considered mathematics to be a model of reality, a belief which has been reinforced many times within engineering.

The map and possibilities

A map can show what is, what might be and the spatial relationship between the two. Contours on a map, for example, may point to suitable locations for a communications tower, a dam, or a highway. Thus, a map can be an embodiment of that which is being designed.

The map and politics

Political maps are required by the government for questions of jurisdiction. These maps indicate borders between countries, provinces, states, counties, municipalities, cities, etc. The areas of concern of various government officials can be indicated on maps. Police typically cover particularly areas and political maps, combined with roadmaps, determining the efficacy with which a given police force can patrol its territory. The map thus symbolizes the jurisdiction of engineers. Normally, engineers concern themselves with the technical area of design. However, they must also be aware of where their work borders the work of others, which may include political issues.

Various maps can also be used to solve legal matters. There may be a dispute as to whether or not a boat was fishing in national or international waters. Squabbling neighbours may settle boundary disputes with reference to a map. We can think of standards as providing the engineer with a map of what constitutes proper design. Certain standards, such as for safety, may have legal implications as well.

One of the uses of maps is that of facilitating the unification effort in 18th Century France [3]. These maps showed where existing roads were and where others might be built. In this sense, we can think of roadmaps with reference to globalization. Roads provided the early transportation and communication links. Later, these same roads provided pathways for the installation of electricity and telephone lines. By building them along the roads, they were readily accessible for maintenance and repair. In still more modern times, we can consider a contour map as facilitating globalization through communication: contour maps show the high point of ground suitable for the construction of radio transmission towers.

The map and the military

The military make extensive use of maps in planning strategy. Lloyd, the Welshman who spied in England on behalf of the French, drew maps of the coast to provide possible landing points into England. These maps were also intended to indicate where supply lines could be maintained once the army has advanced inland. The map in this case signals the military function as well as represents the military past of engineering and the some of the military values it has inherited, such as the ordeal nature of engineering education [4].

The map and distortion

Maps come in various forms, and the form required depends on one's purpose. Maps are drawn according to different scales and therefore show various levels of detail. Some details are omitted. For example, a road map does not typically show contours. Other details are misrepresented. Roads are shown in various colours and are often much wider than the scale would dictate. Cities are often shown as dots rather than the odd shapes of their actual borders. There are also details included which do not literally exist, such as a dotted line used to represent some type of political border. The distorted view contained some in map projections can be likened to the engineering perspective of the world which may appear distorted to others. At the same time, the distorted view may satisfy some particular purpose.

The map and production

Consider a relief map. Engineers can use such a map to locate production sites. In the first instance, production is based on agriculture. The relief map shows lakes and rivers that can be used for irrigation, water power, water transportation links, all of which promote agricultural production. Contours on a relief map show steep terrain that may be unsuitable for certain types of agriculture. Contours allow for the planning of roads, determining the best route in terms of grade and distance covered. These roads provide the pathways for the transportation of the produced goods.

The map and engineering history

One of the main tasks of engineers in 18th Century France was the production of maps [3]. It was short-lived, as the "Age of the Map" drew to a close around 1800. Thus, the map can provide a window on both the history and prestige of engineers.

The map and design

The map also speaks of an etymological link to engineering. Maps were drawn by hand and the French word for a drawing is *dessin*, which can also be translated as *design*. Those who did the drawing were known as *dessinateurs*, or *designers*.

The map and science

Engineers also used maps to learn about nature [3]. Engineers see themselves as improving nature, or at least revealing certain parts of it. Furthermore, as science attempts to explain nature, we can even think of maps of nature for scientific purposes. One example of a map would be a map of the human body or the DNA that represents it. Another example, more closely related to engineering, is a hydrologic map. Contours, rivers, lakes and rainfall data can be used to predict flooding or to control water resources. Other maps can show wildlife distribution to aid in the decision as to the proper borders of a national park. Maps can thus embody and represent scientific information.

2.3 The engineer is a bridge-builder

Bridges are often presented as examples of engineering daring, technology and even artistic sense. With the bridge metaphor, however, I refer to the links that engineers create. By creating a bridge, engineers express their belief that nature is lacking (garden metaphor) and that there are suitable places to create that link (map metaphor). Engineers filled in the gaps of nature. Engineers have succeeded where nature has failed.

The bridge also represents links in a broader sense. There are transportation links, such as highways and vehicles, railroads and trains, and airplanes. There are also communication links, such as mail service (using a transportation system), the telephone and internet. There are even links to the unknown, such as embodied in the space program.

The bridge also speaks of the roots of engineering. One of the early engineering schools in France was *Le Corps des Ponts et Chaussées* [5] (*pont* of course, being French for bridge).

In a less flattering concept, bridges can be viewed as an imposition upon nature. This imposition is even referred to as being violent [3]. Some of the other links that engineers have helped create may also be viewed as violent or perhaps intrusive. The communication links created by the internet allow news to travel very quickly to many parts of the world. Although these links may have the benefit of keeping us informed, privacy has become harder to maintain.

3 Using the Metaphor

We have used the concept of metaphor to capture some of the *why* of engineering design. Metaphors are well suited to this task; being concise, they are easy to remember; being descriptive, they can highlight a variety of historical issues and ideological or philosophical assumptions within engineering. Furthermore, the descriptive power of metaphor is linguistic, reminding engineers that mathematics is not the only descriptor at their disposal.

To make the metaphors accessible, we have avoided those that rely on poorly understood objects. Instead, the metaphors we have presented (the garden, the map and the bridge) are simple things from everyday life and are very familiar to engineers. The three metaphors highlight different aspects of engineering. The garden, although not an engineering concern *per se*, highlights the aspiration to improve production by rearranging nature by employing the concepts of science. The map, perhaps the richest metaphor, was historically an engineering concern. It embodies a wide range of engineering assumptions and beliefs, the most important of which might be the belief in mathematics and a rendering of what might be. The bridge has been and continues to be an engineering concern and reminds us of the belief of engineers that designing links is a good thing. That bridges can fail also serves as a warning about acting beyond one's knowledge.

Finally, though the metaphors highlight different aspects of engineers, there is continuity among them. The same concept may appear in different metaphors, although in different forms. For example, both the garden metaphor and the bridge metaphor highlight a belief that nature can be improved. In addition, the bridge represents the engineering belief that links are basically good and the map shows where those bridges might be built.

These metaphors can also be used in the teaching of engineering. It has been observed that the student often has little understanding of what an engineer does and why he or she does it; what guiding principals are available and how limits to knowledge figure in engineering practice have also been seen to be amenable to learning via metaphor.

4 Concluding Remarks

These metaphors provide accessible ways in which engineers may critically reflect on their practice to better understand and communicate what they are doing and why they are doing it. As the need and their ability to self-critique grow, engineers can become more aware of the impact their designed products may have on society and their responsibility towards society.

References

[1] Canadian Engineering Accreditation Board, Accreditation Criteria and Procedures, 1998.

[2] Lakoff, George and Johnson, Mark, "Metaphors We Live By", The University of Chicago Press, Chicago, 1981.

[3] Picon, Antoine, "Architectes et Ingénieurs au Siècle des Lumières", Éditions Parenthèses, Marseille, 1988.

[4] O'Neal, J.B. Jr., "Engineering Education as an Ordeal and its Relationship to Women in Engineering", ASEE Annual Conference Proceeding, pp. 1008-1101 [sic], 1994.

[5] Grayson, Lawrence P., "The Making of an Engineer", John Wiley and Sons, New York, 1993.

Paul Winkelman

1.0 Department of Mechanical and Manufacturing Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, T2N 1N4, Canada, Tel: 1(403)220-4145, Fax: 1(403)282-8406, E-mail: winkelma@enme.ucalgary.ca

HEEDFUL ACTION, REFLECTION, AND TRANSFER IN THE DESIGN PROCESS

K Lauche

Keywords: Reflective Practice, model of design process, design understanding, total quality management

1 Introduction

Different approaches have come up with descriptive or prescriptive models of how to do good design, both recognising that design has an important role in prevention of quality problems right from the start. However, there has been considerable dispute between the normative school of thought advertising guidelines such as the VDI 2221 [1] on the one hand and design practice observations that emphasize reflection in action on the other hand [2]. The objective of this model is to overcome this diversion by bringing theoretical considerations from design methodology to an empirical validation with design practice [3]. This paper gives an account of reflection in terms of designers' experience and a diagnostic tool for the organisational prerequisites for good design practice. The focus is not purely on a methodology for the individual designer but on the interaction among design teams and the management of the design process.

This paper describes the theoretical concepts and empirical results for the three core elements of the model heedful action, reflection and transfer. The theoretical concepts integrate management approaches such as quality management [4,5] and psychological action theory about goal-directed behaviour [6] with design research. They describe both individual, collective and managerial processes for monitoring design and process quality.

For the validation of the model, qualitative interviews were conducted with 18 mechanical engineers from the Swiss machine industry [7]. The sample was selected from different types of companies (R&D departments in trusts, medium-sized companies and engineering service provider), different professional experience and age groups. The designers were asked to describe their task, how they go about their work and what kind of organisational support they receive. The transcript was categorised according to the model for each of the theoretical concepts and for each person. The results were discussed and validated in a workshop with the interview partners and experts from design science and quality management.

2 Heedful action

People act heedfully when they act more or less carefully, critically, consistently, purpose-fully, attentively, studiously, vigilantly, conscientiously, pertinaciously [8]. According to psychological action theory, this can be seen in a sequence of goal setting, orientation, planning of how one would go about the task, carrying out the plan and evaluation of the results according to the goal [6]. It describes the general, often unconscious process of human goal-directed action and resembles parts of design methodology. At the core of this methodology was Jones' three step model of analysis, synthesis and evaluation. These elements were combined into an iterative process model (see figure 1):

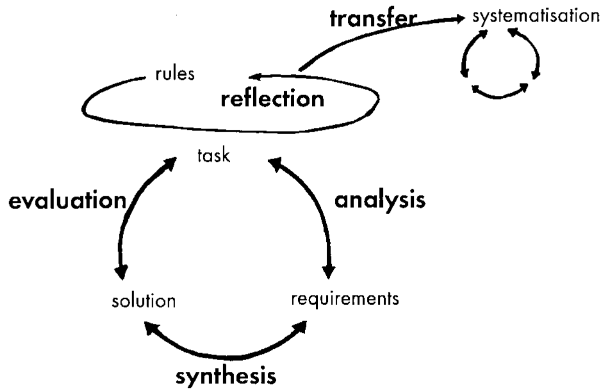


Figure 1. Model for heedful action [7]

The analysis goes back and forth between the task and the requirements. Thereby the task is reformulated into a personal understanding of it and transformed into a goal. The synthesis is the heuristic search for a solution in the face of the requirements. The solution is carried out and tested, ideas are permanently and finally judged against the task. In terms of systems engineering, this model does not resemble the live stages of a product but the problem solving cycle: This heuristic process can occur at any stage of developing a new product.

The participants of the study mentioned that missing the topic is the source of most problems and the thorough analysis is a critical issue. When planning a project, it should be taken into account that things might not work out as planned and that a second option is available if one does not work. The interrelation with others was seen as important, e.g. to ask others when uncertain or stuck with a problem and to communicate difficulties to the customers immediately. Depending on their personal experience, some engineers put more emphasis on search for low-cost alternatives, others on systematic testing – both aspects of quality as reliability and customer satisfaction. One should not only wait for feedback but also try to obtain it and should keep records of data and decision for later inquiries. The condensed practical knowledge from the interviews contains more aspects of social competence and interaction with others than the initial theoretical model. For the designers, dealing with the customer and with the understanding among the team members were important issues for heedful action.

2.1 Context specific social interaction

Following the descriptive approaches in design science, the model takes the organisational context into account: What may account for heedful action is dependent on the demands from the market, the technological options, the internal organisation and strategic goals of the company and the available infrastructure and tools.

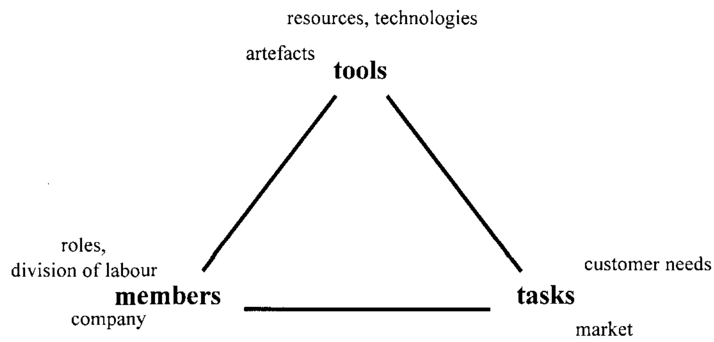


Figure 2. Design as interaction with the context of team members, tools and the task [7]

In the interviews, the designer dedicated more importance to the context and to social interaction than was previously assumed. The task and the organisation had major impact on what is considered heedful action and on how the designers proceed in their work. For the medium-sized companies of the sample, the task was to find an adaption of existing solutions to meet the needs for the specific customer. What is required id not the best result but a pragmatic, fast solution, which had to be monitored throughout production and assembly. Opposed to that, the engineers in R&D departments of big companies were more specialised and handed their results over to engineering for industrialisation. They had a budget and project plan for research and conceptualisation. The third type, the service providers were flexible, project-based organisations who took over subtasks of major projects as an emergency help for other companies. The collaboration within the team and with other parties changed with each project and was described as an important factor to the overall quality.

According to the differences in the context, the methodological approaches differed between the three types of companies. Only the engineering service providers followed the procedure described in VDI 2221, but also emphasized the constant communication with their customer, which has no role to play in the guideline. In the R&D departments, innovation management was more dominant than design methodology: The main issue was to spend resources on the promising projects whereas the design process itself was left to the individual. For the medium sized companies dealing with variants of known problems, the classical methodological training, which is focussed on solving new problems, has very little to offer for pragmatic answers in a short time.

2.2 Prerequisites for heedful action

According to the model, heedful action is not a personality trait but can be supported by organisational factors. The business excellence and TQM approaches [8] and psychological theories offer a concept on how task orientation can be established.

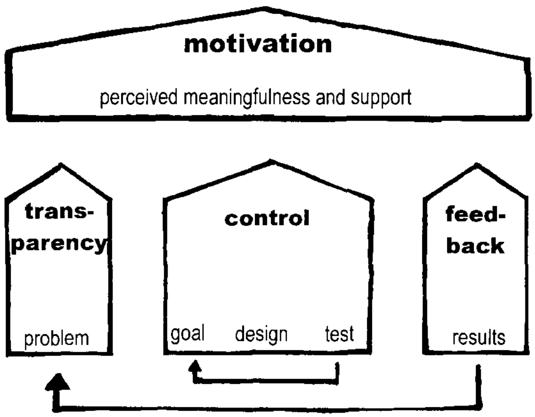


Figure 3. Prerequisites for heedful action [7]

The following prerequisites for heedful action have been formulated:

- transparency and availability of information on requirements in order to know what the need for the product is
- options for design, control and intervention during the development process so that one can actually influence the quality of the output,
- feedback on action results for motivational reasons and in order to adopt adequate strategies,
- perceived meaningfulness of one's work and effort.

For these prerequisites, an assessment tool for task analysis was developed [8]. According to the listed prerequisites, forms of organisational support were elicited from the participants of the interview and assessed. The tool can be used as an internal benchmark to improve the design process and to specify required actions.

How do you know what is required?	
How do you get input from the market?	
Which kind of information are you given on management objectives?	
How do you receive information on new technologies?	

Table 1. Example from the assessment tool for prerequisites for heedful action [8]

The organisational prerequisites generally work fine as far as the product was concerned: Designers received the necessary information on requirements or could investigate for themselves. In some cases they lacked contact and information from the customer. Information on technological progress and existing solutions from the literature was judged poor in most cases. Also, options for influence on the quality of the product were perceived as immense, but organisational aspects were beyond the reach of the designers. All designers naturally received feedback from their calculation and test, but lacked information about more distant outcomes such as customer satisfaction and market impact. The motivation for heedful action was seen in the challenge of the task itself, as the model assumed. The dissatisfier varied between the different companies for whatever was seen was missing support or lack of understanding from either management or other parts in the company.

3 Reflection

Reflection is a central element of the proposed model as research on complex problem solving has shown that it is the most effective strategy to avoid typical human errors in reasoning. It can occur as an individual state of mind during designing or can be applied systematically as a critical review of the chosen methodological approach for learning and improving strategies. This has been successfully used in training for critical situations [9] and with groups as articulation and public reflection for educational purposes. Experts pass on their knowledge and experience by making their mental reasoning accessible to novices. Schön [10] even describes the entire process of designing as reflection in action, as a dialogue between the designer and the object as well as his social surroundings.

The normative idea of this model was that engineers should deliberately apply reflection to improve their way of working and that milestones in a project could be used to go through a collective post mortem analysis in the team. The team would recall the stream of actions and review them by questions like the following:

- How did we go about the problem?
- Which goals were met?
- What went wrong and why? Did we have an inappropriate concept of the system?
- What went well and why? Was success predicted or did it just happen by chance?
- How was the co-operation in the team and with other partners? What was positive and why? What created conflicts and how could they be solved better?
- How did we handle resources? What was missing? What kind of support do we need to do better?
- What did we learn about the market and the customer? What can be transferred to future projects?
- What was our methodological approach in searching, ideas generation and decision making? How should it be changed?

The results of this team reflection should be visualised during the discussion, evaluated and transformed into actions to be taken. One of the service providers suggested to add such a project review to their quality management system to improve their performance. However, it had only been practised a few times. Overall, systematic reflection was mentioned only rarely in the interviews. It occurred as a pondering by chance, not as a deliberate postmortem analysis of one's strategies. It was only described as a private event, not as a review of the methodological approach. Such collective reflections were considered unproductive time that no one would pay for. In the workshop, experts and designers agreed that reflection had a great potential but required professional guidance in order to avoid futile discussions.

4 Transfer and organisational learning

The transfer of experiences and knowledge and reuse of solutions has gained importance recently. Instead of reinventing the wheel organisations try to exploit the existing body of knowledge. However, reuse has not always been so easy when client requirements differed or when designers felt inclined to come up with a new solution or had different preferences from the original designer [12]. It is also a well-known fact that organisations know less than their members do, so even the information about existing solutions may not be publicly available.

In this approach, transfer is taken to mean the sharing and exchange of experiences to the extent that others would know the area of expertise and could direct specific questions to the right person. Successful practices or procedures to avoid problems can become common knowledge e.g. as part of a management system or a data base. Thereby, the individual experience is transformed into systemisation and standards of the company. Organisational learning approaches would recommend a combination of a top-down strategy and local exchange on a personal basis within project teams.

In the interviews, transfer was considered an important but neglected issue. Fourteen out of 18 participants regretted that they had too little opportunity to mingle with other colleagues on parallel projects and could not help reinventing the wheel without even noticing. The normative imperative to communicate relevant aspects proved difficult in practice as it was not clear who would need the information. The problem of transfer became particularly salient when companies were growing. The positive examples worked on an immediate, local basis of teams who meet occasionally. Stable teams, collocation and infrastructure of informal meetings in the cafeteria were seen as helpful. Internal presentations of projects worked fine where hierarchy was not a problem. Also, product data management systems proved to be helpful for finding existing solutions. However, the designers were not interested in additional paperwork or in formalised, abstract collections of knowledge imposed by the company.

5 Conclusion

The proposed model gives an account of the social nature of design that explains how the social nature comes into play. It makes recommendations on how the interaction with the social context can be used to improve the design process by heedful interrelation. Communication among design partners as well as the internalised dialogue with the anticipated other – colleagues from other disciplines, service and production or the customer – can play an important part in prevention of faulty design.

The model also emphasizes reflection and transfer to link the individual experience to the local community of engineers in the organisation and to arrive at better practices and business processes. Reflection can help to improve strategies and to articulate assumptions and intuitive knowledge. However, if reflection is to be taken seriously, it requires a change in engineering education and in the identity of the discipline. Professional organisations still advertise engineering as "creating progress for humankind" by means of technological development. If the *homo faber* is to be freed from this mysticism and transformed into a reflective practitioner, they need to be trained to consider their task at a wider dimension and to reflect and communicate with others.

From this empirical study, the recommendation for industrial application would be to establish a forum for internal knowledge transfer where strategies are explicated and collective reflection is encouraged. In design education reflection could be introduced as a pedagogic means of mutual instruction. The curiosity which initiates design research – to understand human creativity and mastering of complex problems – could also be passed on to engineering students to envision their activity from a meta view. This ability to look at oneself in an abstract way is a basic training for interdisciplinary work and sociological discussions about technological progress. Instead of supplementing arbitrary arts courses, engineering education could profit from a deliberate instruction in reflection of strategies. This, however, requires intensive dialogue between the traditions and disciplines within design science.

References

[1] VDI 2221, Systematic Approach to the Design of Technical Systems and Products, Berlin: Beuth, 1987.

[2] Dorst, K., Two Kinds of Design, and Two Sides of Design, Pro. of ICED '99, p. 1547 - 1552, 1999.

[3] Schregenberger, J. W. The Further Development of Design Methodologies. In E. Frankenberger, P. Badke-Schaub, and H. Birkhofer (eds.), Designers. The key to Successful Product Development. London: Springer, p. 57-67, 1998.

[4] ISO 9000:2000. Quality Management systems – Guidelines for performance improvements, 2000.

[5] Hackman, J. R. and Wageman, R. Total Quality Management: Empirical, Conceptual, and Practical Issues. Administrative Science Quarterly, 40, 309 – 342, 1995.

[6] Hacker, W., Action theory and occupational psychology. Review of German empirical research since 1987. The German Journal of Psychology, 18, p. 91-120. 1994.

[7] Lauche, K., Quality Action in Product Development. Theoretical Model, tool for analysis and results for improvement. (Qualitätshandeln in der Produktentwicklung. Theoretisches Modell, Analyseverfahren und Ergebnisse zu Förderungsmöglichkeiten). Zurich: vdf, 2001.

[8] Weik, K. & Roberts, K. Collective mind in organizations: heedful interrelating on flight desks. Administrative Science Quarterly, 38, p.357-381, 1993.

[9] Lauche, K. Individual and Organisational Options for Controlling and Improving Quality. Proc. of Int. Conf. on TQM and Human Factors, p. 65 – 71, 1999.

[10] Badke-Schaub, P., Wallmeier, S. and Dörner, D., Training for Designers: A way to reflect design processes and cope with critical situations in order to increase efficiency. Proc. of ICED '99,p. 205 – 210, 1999.

[11] Schön, D. A., The Reflective Practioner. How Professionals Think in Action, New York: Basic Books, 1983.

[12] Busby, J.S. The Problem with Design Reuse: An Investigation int Outcomes and Antecedents. J. of Eng. Design, 10, p. 277-296, 1999.

Dr. Kristina Lauche
Department of Psychology
University of Aberdeen
Aberdeen AB 24 2UB
Scotland UK
Tel. +44 (0) 1224 272280
Fax +44 (0) 1224 273211
k.lauche@abdn.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

OBJECT-ORIENTED MODELLING OF THE DESIGN PROCESS

N Pavkovic, D Marjonovic, and D Dekovic

Keywords: Object-oriented methodology, design process modelling, computer-based design support.

1 Introduction

The need for development of integrated software framework for design process support (often named designers workbench) arises from growing capabilities of modern CAE systems and requirements for new engineering design methods. The goal of this research project is to develop a kernel structure for such a framework that is founded on object oriented design process model. Presented approach tends to establish a flexible structure, providing the tools for managing the design process and to integrate the usage of computer-based design support tools. The system is targeted to support the classes of adaptive and variant design. None of the design theories which are in use today, like e.g. descriptive, prescriptive, general, and axiomatic design theories, is alone capable, to describe and to formalise the whole design process and to make possible a complete computer support of this process [1].

Keeping this in mind, the presented research aims to develop a framework that could enable the use of different partial models in an integrated manner. We assume that the object-oriented methodologies can provide more appropriate and flexible ways of such a software system modelling than the other techniques. The design process is here viewed as a sequence of transitions from an initial state of data, constraints and goals to a final state, the description of the artefact being designed. These transitions are allocated to individual participants or teams, and individual or integrated computational tools or models. A design process may no longer be viewed as a static institutionalised structure, but rather as a dynamic network that is constructed in real-time as design goals, needs, priorities, and resources evolve [2]. In the framework of an integrated CAD environment, the designer should work with an open toolbox, enabling him to create his own classes and partial models of the design process according to his current needs. The goal is to develop a framework that enables to model the design process (and to integrate used software tools) independently of design phase and class of design task.

Engineering data structures are much more complex than in common business database applications. The associations between structures are numerous, and the same data structure can have many different roles. Hence, a design process topology is here considered as two networks of relations:

- Complex multi-level relationships between engineering data structures
- A network of sequence relations between design process steps, including the models of iterative processes

The thesis of the presented research is that a design process topology (as above defined) could be efficiently modelled with object-oriented methodology. This thesis is examined and validated on an example of the prototype model implementation, realised in object database. The structure of the proposed object model is documented in the UML language.

2 Methodology

In the proposed approach, every occurrence, kind of action, set of information, relation and other real-world “things” from the domain of the design process, are attempted to be modelled as objects. In the presented research phase, the considerations are limited to recognition of basic entities, their attributes and relationships, while the necessary operations are only denoted. Proposed design process model is conceived as an “open toolbox” with packages of classes and objects. In such an environment the designer should mainly use the existing classes as templates and building blocks. The applied research methodology could be best described with scheme according to Duffy & Andreasen [3], [4].

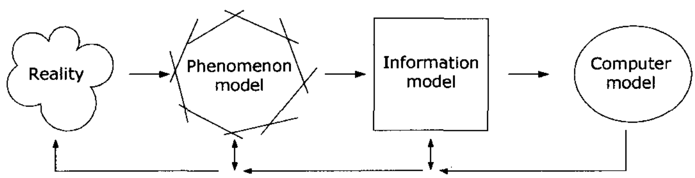


Figure 1. The research methodology

The path from left to right explains research carried out where phenomenon models are gradually formalised by means of information and computer models [4]. The path from right to left is verification, thus models are confronted with reality, i.e. empirical observations.

Object-oriented analysis begins here with an examination of the real-world 'things' that are part of the problem to be solved. Those things (which we will call objects or entities) are characterized individually in terms of their attributes (transient state information) and behaviour (functional process information). In object-oriented terms, we discover and describe the classes involved in the problem domain, i.e. the reality and the phenomenon model. In parallel to these individual characterizations, we also model the links or collaborations between the problem domain's objects (and therefore the solution's classes). Object-oriented design then turns from modelling the problem domain towards modelling the implementation domain, i.e. the information model and the computer model.

3 Entities and the structure of the proposed object model

The main result of this research is the proposal and the definition of the structural, relational and behavioural entities of the design process model. The proposed model is founded on four loosely coupled sets of entities (classes), as shown in figure 2. Such a structure promotes a bottom-up approach – the elementary entities and their relationships are combined and used in building the more complex entities to describe the design process. "Service classes" model the dynamic issues of the proposed system exploitation. Other sets of classes are focused on static, mainly data, structures of the proposed model. "Service classes" include sets of operations that realize the functionality of the software system - e.g. interfaces and procedures for the design plan generation and execution. Service classes should have no instances, therefore they are not stored in object database. Particular entities in each of four main subsystems are specified in table 1.

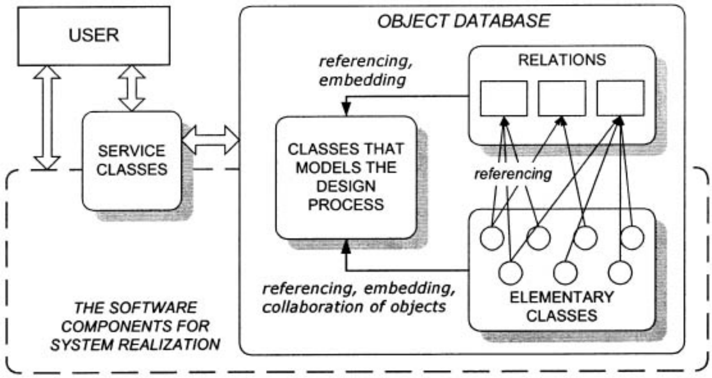


Figure 2. Fundamental (global) structure of proposed model

Table 1. Entities of the proposed object-oriented design process model

ELEMENTARY ENTITIES	RELATIONS BETWEEN OBJECTS OR THEIR ATTRIBUTES	PROCESS REPRESENTATION AND EXECUTION	THE COMPONENTS OF SYSTEM ARCHITECTURE
• design parameter • parameter's database • product information object • product structure object • design process action • software tool interface • design task • designer	• dependency relations between parameters • design structure matrix (design tasks dependencies) • membership relations between objects of different classes • expressions – relations between object attributes • design constraints – relations between design parameters and functional requirements • design decision rules – relations between conditions and actions	• design plan • design plan node • matrix representation of the design plan network • design process execution flow • design process capturing • design plan generation & execution • design plan sketch • knowledge bases about particular design processes	• "service" classes – the system functionality • object database dictionary • object database with design plan skeleton (framework) • created plans archive • executed plans archive • interfaces to knowledge bases and EDM/PDM systems • knowledge base about design support software tools

3.1 Elementary entities (classes)

Design parameter - Considering a design process as a process of information generation and transformation, it is assumed that the basic (simplest) entity of the design process model is a variable, often named a design parameter or design attribute. Design parameter, modelled as an object encapsulates: value; value status; references to knowledge; and the procedures for capturing and recording the proposals and arguments in collaborative processes (similar as in

[5]). The parameter *value status* is proposed as one of: determined, but can be changed; determined and fixed; assumed single value; value assumed in discrete or continuous interval. The notion of "value status" can be particularly useful in iterative processes, where it enables the development of improved algorithms for solving such problems [6].

Product information object - Encapsulates all occurrences, concepts and events in the lifecycle of a particular set of information about product being designed. This object can represent (model) a computer-stored information or some other kind of information storage. The information set can be from functional or physical domain. Product information object can be very complex in structure because it can contain other objects or references to other objects in design process model. The examples of product information objects could be: 3D CAD model, 2D CAD drawing, bill of materials, product assembly structure, any other I/O data set, etc. Product information objects undergo the most of the operations in the design process. At the end of the design process, the set of the product information objects forms the information collection about the product being designed.

Action - Models the calls of the operations included in classes of the proposed model. Actions mainly operate on product information objects. Actions should be modelled as objects in order to be used as elements in building the description of design process flow dynamics. In this approach the actions are primarily considered in the context of information generation and transformation process, not in the context of the design methods.

Design task - A design task is here defined as a set of sequenced actions performed on a set of product information objects. The design task goal is to reach a required state of all the objects that are in scope of a particular task. Usually, the design task represents the process of solving a subproblem, or a part of a decomposed structure of a product design process. Design task is a complex object which includes: the references to a set of product information objects that are in scope of a particular task; the desired goal states of those objects; I/O parameters, resources, available services and terms, and the responsible designer(s).

3.2 Related research projects

Similar treatment of OO formalism in representation of the design process can be found in the work of Gorti et al [7]. Their model is implemented as a layered scheme that incorporates both an evolving artefact and its associated design process. They define the design process with five primitive objects: goal, plan, specification, decision and context. Liang and O'Grady [8] have developed a concept of a design object that represents physical entities such as parts or components as well as non-physical entities, such as design history or vendor information. They build the design process model formalism on a definition of a design model, design objects and design methods.

3.3 Relations between entities

In the presented approach, the relations between objects are also viewed as objects [9]. Therefore the proposed design process model should contain special classes to represent and manipulate the relations between elementary structural and other more complex entities. When analysing different information models we can find some variations in classification of relations. In the proposed model we rely on four types of relations, according to the authors of Unified Modelling Language: dependency, generalization, association and realization.

4 Modelling the design process network topology

The complex network of relations between design process model entities is analysed as a central issue of the presented research. It turned out that a very large and complex object network could be efficiently modelled and managed in an object database environment [10]. Thus, a design process model is built using a "bottom up" approach in which the elementary structural entities and their relations are used as a building blocks for more complex entities which model and control the design process flow.

Let us consider a network of relations between elementary entities of the proposed model. Generally, every entity from table 2 can be related to every other entity from table 2. It is obvious that only a subset of such relation network can be of significance as a part of design process model. However, the object technology enables the whole network to be easily modelled, recorded and managed. Every "x" mark in table 2 indicates that it should be useful to establish the relations between instances of such classes. Only some of the most significant combinations from the table 2 are emphasized and discussed in the rest of this chapter.

Table 2. A subset of relation network between classes of the proposed object model

		DP	PDO	PSO	STI	A	DT	D	DPR
design parameter	DP	x							
product document object	PDO	x	x						
product structure object	PSO	x	x	x					
software tool interface	STI	x	x	x					
action	A				x				
design task	DT	x	x	x			x		
designers	D		x	x			x		
design proc. representation objects	DPR	x	x	x	x	x	x		x

4.1 Dependency and "membership" relations between sets of objects

In the design of object-oriented systems, the most interesting part are not the objects themselves, but the relationships among objects, and these relationships constitute a large proportion of the semantics of most systems. Generalization (specialization) relations in the proposed model are implemented as class hierarchies. Relations shown in table 2 are implemented as associations between instances of the same class or between instances of different classes. Association relations are recorded in "reduced" matrix form (table 3), where each instance of class 'A' has a set of pointers to related instances of class 'B'.

Table 3. Relations between sets of class instances, recorded in a matrix form

	b_1	b_2	b_3	b_4	b_5	b_6	....	b_n
a_1				x				
a_2			x			x		
a_3		x			x			x
....								
a_n					x	x		

→

	Pointers to related instances of class B	Relation semantics
a_1	b_4	$a_1 = (b_4)^2$
a_2	$b_3\ b_6$	$a_2 = b_3 + b_6$
a_3	$b_2\ b_5\ b_n$	.....
....		
a_n	$b_5\ b_6$	

A matrix, "reduced" to three columns, can be considered as a set of "one to many" relations. In the proposed model we distinguish the "*dependency*" and "*membership*" semantics of

relation. In the development of the prototype model, two sets of dependency relations are implemented: the design structure matrix (dependencies between design tasks), and “the design equation matrix” – algebraic dependencies between design parameters. Those relations are modelled as separate classes. The operations for reorganization of design structure matrix will be developed and implemented in the next research phase. We expect that the proposed approach enables to fully integrate a “partial” model of the design process, such as the design structure matrix, in the framework of the more general computer-based design process model.
The semantics of “membership” relations is that a referenced object “belongs” to “parent” object, i.e. the object of class ‘A’ has a set of pointers to belonging objects of class ‘B’ (table 3). Membership relations are modelled as associations between instances of different classes.

The proposed object model is completely developed and documented in UML language. Figure 3 shows the UML diagram of “membership” associations between designers, design tasks, design parameters, product information objects and product structure objects.

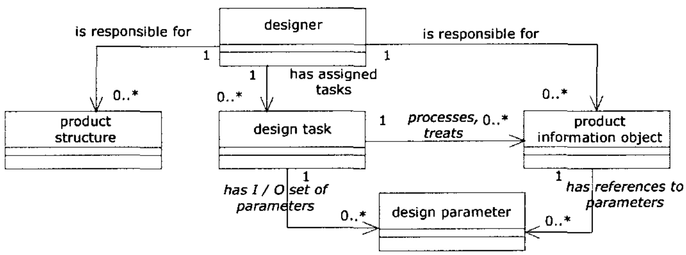


Figure 3. An example of UML class diagram - the associations that model the “membership” relations

UML model of the proposed system is mapped to object database dictionary which contains definitions of all design process model classes, i.e. it constitutes a framework of the object-oriented design process model. The database dictionary can serve as the open toolbox – the designer can easily create new instances of existing classes. Furthermore, “templates” of design methods and procedures can be modelled as composite classes and integrated in the design process model framework.
The object-oriented database maps objects that has been created in C++ or Java onto objects in the database. This kind of direct mapping enables the database schema to be generated automatically. Even in the case of highly complex object models, there is no difficulty writing them to the database. The database "knows" what an object is and also recognizes the relationships (references, pointers) between objects. These are simply stored together with the objects themselves, and are therefore reproduced easily at any time the data is required.

5 Representation of the design process

A design process is represented with the design plan - a collection of nodes and their connections in a directed graph, recorded in adjacency and incidence matrices [11], [12]. The connections between nodes represent the information flow and/or the sequences of node execution. The design plan node models one step of the design process [10], including: checking of preconditions, list of actions, checking the postconditions and deciding about the next step. Preconditions and postconditions include sets of constraints and rules, also modelled as objects. Constraints and rules include references to design parameters and attributes of all classes of objects that constitute the design process model. Design plan nodes are hence very complex objects, with sets of direct or indirect references to the majority of other class instances of the proposed model [10]. Besides, the node class contains all the necessary operations for managing the design process flow. An "agenda" (processing schedule) is attached to every set of references in node. Such a structure makes possible to develop the procedures for dynamic modifications of the design plan at the execution time. This is a very important issue for further research, because such possibilities could bring the proposed model much more closer to the reality, i.e. the human abilities of planning and adapting a finished plan to the new (unplanned) situations that can occur while executing. The process of using a proposed system in modelling the particular design process and (in the same time) integrating the existing software tools is shown on figure 4.

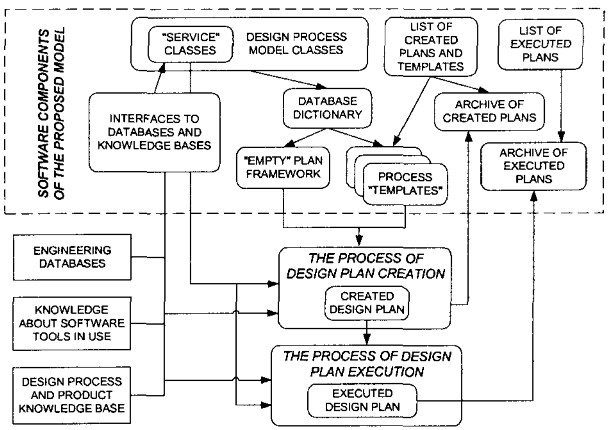


Figure 4. The global schematic view of proposed system usage

5.1 Case studies

In this phase of research, case studies have been considered only theoretically, based on the author's experience in integrating the computer-based design support tools in design office of the medium size electrical machines factory. The design process in mentioned office includes typical variant and adaptive design tasks assigned to small design teams. It turned out that such design tasks could be efficiently modelled in the framework of the proposed system. More complex case studies are expected to be done in the next research phases.

6 Conclusion

The object-oriented database as an implementation environment offers the benefit of designing the object model with no regard for the shortcomings of traditional database technologies. Thus, a model that best suits the particular needs can be implemented. One of the main advantages of the proposed approach is the ability to easily model and manipulate very complex data structures and networks of relationships. The presented research phase includes the analysis primarily of the structural and some of the behavioural elements of the proposed design process model. The proposed object structure is still a scaled prototype model. A lot of work remains to be done: refining the proposed structure and definitions; defining "use cases"; defining and diagramming collaborations between objects; designing "main" system operations that have the threads of control; developing graphical user interfaces. The techniques of object oriented modelling and design have been intensively explored in last ten years, and it seems that they finally reached a high level of maturity. The benefits of object oriented methodology raises new challenges. Using unified modelling methodology (such as UML), a distributed design community can easily collaborate in developing the design process model. Such approach could enable initiating the process of creating a theoretical framework for integrated computer-based design process support that will be widely accepted in a design community. The main difficulty of this research, particularly in developing the entities of the proposed model, was (and still is) the lack of the general consensus in design terminology, taxonomy, and typology, as emphasized in [13].

References

[1] Vajna S., Wegner B., Optimization of the Product Development Process with Evolution Algorithms and Simultaneous Enginerring, Proceedings of ICED 97, Vol. 3, pp. 3/67-3/70, WDK, Heurista, 1997.

[2] Wallace D. R., Abrahamson S. M., Borland N. P., "Design process Elicitation Through the Evaluation of Integrated Model Structures", Proceedings of DETC 99, ASME Design Engineering Technical Conf., 1999.

[3] Duffy A.H.B., Andreasen M.M., Enhancing the Evolution of Design Science, Proceedings of ICED 95, Vol. 1, pp. 29-35, WDK, Heurista, 1995.

[4] Mortensen N. H., Design modelling in a Designer's Workbench, Ph.D. thesis, Lyngby, Technical University of Denmark, 1999.

[5] Nagy R. L., Ullman D. G., Dietterich T. G., A Data Representation for Collaborative Mechanical Design, Research in Engineering Design, Vol. 3, No. 4, pp. 233-242, 1992.

[6] Eppinger S.D., Whitney D.E., Smith R., Gebala D., "A Model-Based Method for Organizing Tasks in Product Development", Research in Eng. Design, Vol.6, No.1, pp.1-13, 1994.

[7] Gorti S.R., Gupta A., Kim G.J., Sriram R.D., Wong A., An object-oriented representation for product and design processes, CAD, Vol. 30, No. 7, pp. 489-501., 1988.

[8] Liang W.Y., O'Grady P., Design With Objects: an Approach to Object-Oriented Design, CAD, Vol. 30, No. 12, pp. 943-956, 1998.

[9] Pavkovic N., Marjanovic D., Entities in the Object-Oriented Design Process Model, Proceedings of 6th Int. design conference DESIGN 2000, pp. 29-34., FSB, WDK, 2000.

[10] Pavkovic N., Object-oriented approach to design process modelling, Ph.D. thesis, University of Zagreb, Faculty of mech. engineering & naval arch., 2000. (in Croatian)

[11] Pavkovic N., Marjanovic D., Structuring a Designers Workbench with Object-Oriented Design Plans, Proceedings of ICED 99, Vol. 3, pp. 1401-1406, WDK, 1999.

[12] Pavkovic N., Marjanovic D., Considering an object-oriented approach to the design process planning, International Journal of Technology Management, Vol. 21, No. 3/4, 2001.

[13] Andreasen M. M., Wognum N., Considerations on a Design Typology, Proceedings of 3rd International Workshop IPD 2000, Otto-von-Guericke University Magdeburg, 2000.

Corresponding author: Neven Pavkovic, PhD
University of Zagreb, Faculty of Mechanical Engineering & Naval Architecture, Design Department, Ivana Lucica 5, 10000 Zagreb, CROATIA
TEL: + 385 1 6168 545 **FAX:** +385 1 6156 940 **E-MAIL:** neven.pavkovic@fsb.hr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

PATTERNS OF PRODUCT DEVELOPMENT INTERACTIONS

S D Eppinger and V K Salminen

Keywords: Process modeling, product architecture, design teams, design structure matrix, complexity management.

1 Introduction

Development of complex products and large systems is a highly interactive social process involving hundreds of people designing thousands of interrelated components and making millions of coupled decisions. Nevertheless, in the research summarized by this paper, we have created methods to study the development process, identify its underlying structures, and critique its operation.

In this article, we introduce three views of product development complexity: a process view, a product view, and an organization view. We are able to learn about the complex social phenomenon of product development by studying the patterns of interaction across the decomposed elements within each view. We also compare the alignment of the interaction patterns between the product, process, and organization domains. We then propose metrics of product development complexity by studying and comparing these interaction patterns. Finally, we develop hypotheses regarding the patterns of product development interactions, which will be helpful to guide future research.

2 Methodology

In this research, we study product development situations by assessing the patterns of interactions within three domains and then compare the patterns across the domains.

2.1 Three Product Development Domains

To study product development, we believe that there are three relevant domains: product, process, and organization. In complex development situations, each of these three domains is decomposed in order to manage the complexity. We therefore begin our analysis by documenting the decomposition of each of the three domains:

- Product: A complex product or large system is decomposed into sub-systems, and these in turn may be further decomposed into sub-assemblies and/or components.
- Process: A full development process is decomposed into phases or sub-processes, and these in turn may be further decomposed into tasks, activities, and work units.
- Organization: A large development organization is decomposed into teams, and these in turn may be further decomposed into working groups and individual assignments.

2.2 Patterns of Interactions

Once we have documented the decomposition, we then document the patterns of interaction between the decomposed elements. It is interesting to do so within each domain (the three matrices in Figure 1):

- Product: The architecture of the product is defined not only by the decomposition of the complete product into elemental components, but also by the interactions between these components. The interactions may include well-specified interfaces and undesired or incidental interactions. System architecture design principles [1] [2] [3] suggest ways to plan architectures with minimal interactions across sub-systems, maximizing the density of interactions within. Documentation of complete patterns of system architecture interactions has been accomplished using matrix-based methods [4] [5]. Analysis of such patterns may be used to suggest clusters forming effective product modules.
- Process: The product development process is generally a complex procedure involving information exchange across the many tasks in order to execute the work. Various network-based methods have been used to map and study development processes [6] [7] [8] [9] [10]. Analysis of product development processes allows us to study product development efficiency and to suggest process improvements.
- Organization: The organization structure determines who works with whom and who reports to whom. However, in development organizations we are particularly interested to study the communication patterns of the people conducting the technical development work. This follows from well established methods used to study communication networks in R&D organizations [11] [12] and can be used to assess whether necessary interactions are taking place within the organization.

2.3 Comparison Across Pattern Types

We believe that the three types of patterns should be strongly related (the three arrows in Figure 1). After all, the development organization is executing the development process, which is implementing the product architecture. When we can compare the map of interactions in one domain to another, we hope to be able to answer questions such as:

- Does the organization properly execute the development process?
- Is the development process effectively implementing the product architecture?
- Are the architecture interactions driving the organizational communications?

But the comparison across such different types of data can be problematic. We have found that where there exists a one-to-one mapping from one domain to another, a direct comparison becomes straightforward. For example, if there is a single development task assigned to each individual team member, then a direct comparison between process and organization is possible. Similarly, where there is a single team assigned to each subsystem, we may directly compare the interactions within the organization to the interactions within the product architecture [13].

In practice, a perfect one-to-one mapping rarely exists in real and dynamic engineering design environments. Many industrial product development situations involve scarce or shared resources, multi-tasking, outsourcing, and dynamic or uncertain development demands, all of which make the analysis difficult. Utilizing a many-to-one or a many-to-many mapping from one domain to another yields a model of potential interactions, not simply expected ones. Since this reduces the predictability of the model, we prefer to conduct the analysis in situations with simpler structures (one-to-one mapping).

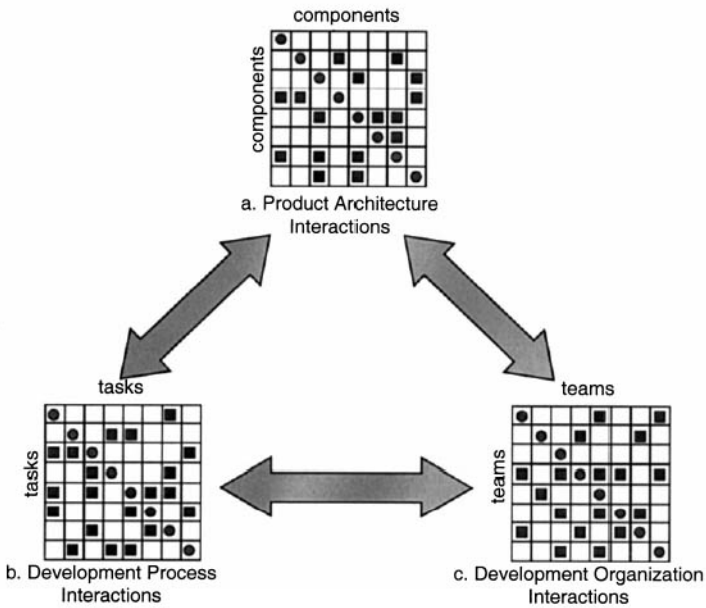


Figure 1. Three domains of product development interactions: product, process, and organization.

3 Industrial Examples

While case studies of this type take a great deal of effort, we have been able to make progress by utilizing graduate student internship projects at several companies. We are not able to describe these projects in detail within this summary article, but publications are available documenting the results of each application.

Figure 2 illustrates industrial examples showing the patterns of interactions in the three individual views, product architecture (Figure 2a), development process (Figure 2b), and development organization (Figure 2c). Figure 2 also shows examples of comparing across two of these three domains at a time (Figures 2d, 2e, and 2f). We have not yet attempted to compare all three views together for a single industrial example.

3.1 Product Architecture Example

Figure 2a shows a model representing the decomposition of a climate control system into 16 components and documenting the product architecture as interactions between the components. 34 interactions were identified along the technical dimensions of spatial, energy, materials, and information [4]. Clusters can be formed along each of the dimensions of interaction individually or by aggregating all of the dimensions into an overall distance measure for each pair of components. The clusters identify groups of highly interrelated components, thus suggesting modules for development, production, and/or potential outsourcing.

3.2 Development Process Example

Figure 2b shows a matrix illustrating the procedure followed by an automobile manufacturer to determine the feasible layout of the engine compartment based on a digital mock-up using CAD solid models [14, p.349]. Interactions in this type of model represent flows of information and data between the tasks. The planned and unplanned iterations within the development process become apparent through analysis of these process data [15]. Such a model is useful for process reengineering by suggesting and analyzing alternative processes in terms of development time, cost, and risk.

3.3 Development Organization Example

Figure 2c shows the decomposition of the organization used to develop a new automobile engine. The organization involved 22 cross-functional teams, each with responsibility for design and manufacturing engineering of a major component or subsystem. The matrix depicts the interactions across the 22 teams in terms of the frequency of their required technical communications. A clustering analysis of the team-interaction data suggested an efficient arrangement of five system-engineering team assignments, with four system teams focused on interactions across groups of the components, and one integration team addressing overall system performance [15] [16].

3.4 Comparing Product Architecture to Organization

Figure 2d shows a comparison of the interfaces specifying the product architecture with the communications inside the development organization for a jet engine. In this case, there was a single product development team responsible for the development of each of the 54 components. This study not only confirmed the ability of design interfaces to predict technical communication, but also revealed several reasons why development professionals do not communicate even when their components interact, and further reasons why teams do interact while their components do not share a direct interface [13]. This research also identified differences in the behavior of teams designing modular components from that of teams designing distributed components [17].

3.5 Comparing Development Process to Organization

Figure 2e shows a comparison of the product development process to its development organization. In this study of designing electronics hardware components, there was not a one-to-one mapping of development tasks to individuals in the organization. While this did hinder the comparison, it was still possible to show that the process model predicts technical communications in the organization much better than earlier models based on geographical layout of the personnel [18]. We also found that even where the development process shows uni-directional information transfers, the actual communications between individuals are predominantly bi-directional exchanges.

3.6 Comparing Product Architecture to Process

The comparison of the product development process to the product architecture of an elevator system is a case study still in progress. This example allows us to study the differences between the nominal and actual development processes and how these changes arise from the particular implementation of the architecture chosen for the product.

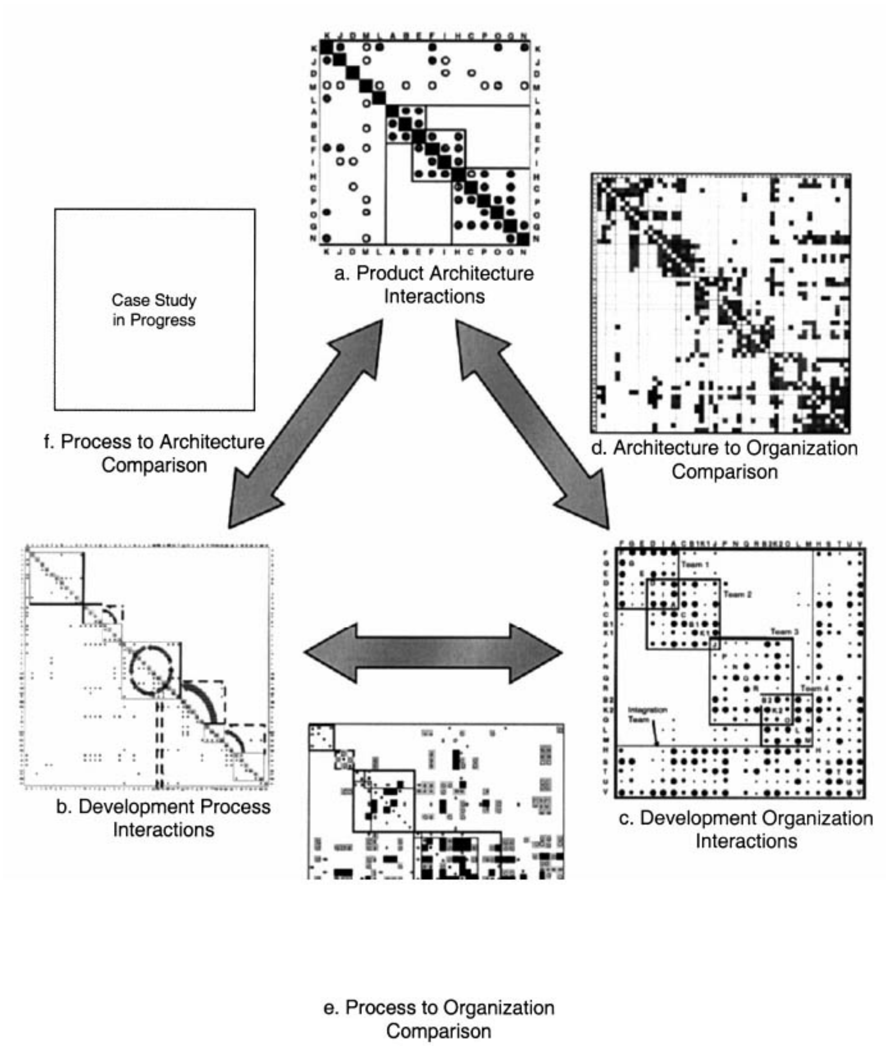


Figure 2. Examples showing matrix-based mapping of interactions in the product architecture (a), product development process (b), and development organization (c). We have also found it possible to compare such models across these domains (d, e, and f).

4 Discussion

4.1 Impact on Industrial Practice

The matrix-based methods summarized in this paper have proven useful for diagnosing and improving product development processes, product architectures, and development organizations. This approach to documentation of interactions and interfaces has been successfully applied to a number of well-established engineering design situations. However, where there is little industrial experience with the development challenge at hand, there is less information available from which to create the models, and we are less able to predict where the difficulties will arise in the development process.

Nevertheless, we believe that the approach applies quite broadly to many engineering-based industries and has special advantages to design of complex systems. In particular, analysis in the three individual domains provides direct benefits:

- Product: Analysis of the product architecture suggests more effective module and sub-system boundaries, highlights critical interfaces, and identifies appropriate outsourcing opportunities.
- Process: Analysis of the product development process leads to streamlining and accelerating the process, reducing and focusing design iterations, identification of failure modes within the process, and replacement of chaotic information flows with more formal procedures where necessary.
- Organization: Analysis of the product development organization can yield more effective system team arrangements and formation of system engineering functions for better integration of the overall product or system.

The three possible comparison views require somewhat more work to build two independent models and interpret their patterns jointly. Still, we have found that these analyses serve to help diagnose cultural and dynamic causes of process-related and organizational failures to efficiently develop the selected product architecture. We also expect that such comparisons will help us to capture system-level knowledge and better understand where it resides.

4.2 Complexity Metrics

Meaningful measurement of complexity can serve to improve our understanding of and ability to work with complex systems. With the help of complexity metrics, it will be possible to track complexity changes over several product generations. We may also be able to benchmark one company's product or process complexity with respect to its competitors. We have not utilized complexity metrics in our research thus far, and this remains an interesting area for future exploration. However, we believe that useful complexity metrics will consider several factors:

- The number of decomposed elements (components, tasks, or teams in our three views)
- The number of interactions to be managed across the elements
- The uncertainty of the elements and their interfaces
- The patterns of the interactions across the elements (density, scatter, clustering, etc.)
- The alignment of the interaction patterns from one domain to another

4.3 Hypotheses for Future Research

In considering what we have learned through many case studies, we have formulated several hypotheses about the dynamics of product development interaction patterns. Future empirical research will involve additional case studies, data collection, and analysis specifically designed to test these hypotheses:

1. Maturity: One hypothesis we have is that the density of the known interactions within any particular view varies with maturity of the product architecture, experience of the organization, and skill in managing the process. Specifically, we hypothesize that at first there are quite few interactions known. Then with experience, more interactions become evident. Finally, a mature architecture has more focused and clustered interactions, with others eliminated or minimized in impact.
2. Learning: We expect to find that, of the three views, experience builds most quickly in the product architecture view for complex, engineered products. This is because engineers learn quickly about the product and its technology, even while the development process and organization remain informally structured.
3. Evolution: We believe that the pattern of interactions within each domain changes over time, not in a random or unplanned manner, but with respect to a reference model [19]. Such a model may be the should-be product development process, the perfect product architecture, or the ideal organization. We believe that the presence of a reference process or architecture will affect the changes in the interaction patterns over time.
4. Co-Evolution: We further hypothesize that the interaction patterns in the three domains change in coupled ways. The organization evolves to address deficiencies in its ability to implement the development process and product architecture. Furthermore, the product architecture and development process may change to compensate for shortcomings in the development organization.
5. Alignment: Finally, we expect to find that industrial firms in which the interaction patterns across the three domains are well aligned will outperform firms for which the patterns are not aligned.

4 Conclusion

This paper presents three important perspectives for studying product development: product architecture, product development process, and the development organization. Within each domain, we focus on the pattern of internal interactions. We analyze these patterns to learn about the particular product development situation and how to improve it. We are also able to compare patterns across the three domains to assess the effectiveness of the process and organization to develop the particular product. After using this approach to study several industrial situations, we have developed some hypotheses which may guide future research in this area.

References

[1] Alexander, C. *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, MA, 1964.

[2] Rechtin, E. and Maier, M.W. *The Art of Systems Architecting*, CRC Press, Boca Raton, 1997.

[3] Suh, N.P. *The Principles of Design*, Oxford, New York, 1990.

[4] Pimmler, T.U. and Eppinger, S.D. "Integration Analysis of Product Decompositions", *ASME Conference on Design Theory and Methodology*, Minneapolis, MN, pp. 343-351, September 1994.

[5] Malmström, J. and Malmqvist, J. "Trade Off Analysis in Product Structures: A Case Study at Celsius Aerotech", *Proceedings of NordDesign'98*, Stockholm, pp. 187-196, 1998.

[6] Steward, D.V. "The Design Structure System: A Method for Managing the Design of Complex Systems", *IEEE Transactions on Engineering Management*, vol. EM-28, no 3, pp. 71-74, August 1981.

[7] Marca, D.A. and McGowan, C.L. *SADT: Structured Analysis and Design Technique*, McGraw Hill, New York, 1988.

[8] Warfield, J.N. *A Science of Generic Design: Managing Complexity through Systems Design*, Intersystems Publ., Salinas, CA, 1990.

[9] Eppinger, S.D., Whitney, D.E., Smith, R.P., and Gebala, D.A. "A Model-Based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, vol. 6, no. 1, pp. 1-13, 1994.

[10] Park, H. and Cutkosky, M.R. "Framework for Modeling Dependencies in Collaborative Engineering Processes", *Research in Engineering Design*, vol. 11, pp. 84-102, 1999.

[11] Allen, T.J. *Managing the Flow of Technology: Technology Transfer and the Dissemination of Technological Information Within the R&D Organization*, MIT Press, Cambridge, MA, 1977.

[12] Van den Bulte, C. and Moenaert, R.K. "The Effects of R&D Team Co-location on Communication Patterns among R&D, Marketing, and Manufacturing", *Management Science*, vol. 44, no. 11, pp. S1-S18, November 1998.

[13] Sosa, M.E., Eppinger, S.D., and Rowles C.M. "Understanding the Effects of Product Architecture on Technical Communication in Product Development Organizations", MIT Sloan School of Management Working Paper, no. 4130, August 2000.

[14] Ulrich, K.T. and Eppinger, S.D. *Product Design and Development*, McGraw-Hill, New York, Second Edition, 2000.

[15] Eppinger, S.D. "Innovation at the Speed of Information", *Harvard Business Review*, vol. 79, no. 1, pp. 149-158, January 2001.

[16] Eppinger, S.D. "A Planning Method for Integration of Large-Scale Engineering Systems", *International Conference on Engineering Design*, Tampere, Finland, pp. 199-204, August 1997.

[17] Sosa, M.E., Eppinger, S.D., and Rowles C.M. "Designing Modular and Integrative Systems", *ASME Conference on Design Theory and Methodology*, Baltimore, MD, September 2000.

[18] Morelli, M.D., Eppinger, S.D., and Gulati, R.K. "Predicting Technical Communications in Product Development Organizations", *IEEE Transactions on Engineering Management*, vol. 42, no. 3, pp. 215-222, August 1995.

[19] Salminen V. and Pillai B. "Strategic Management of Adaptive, Distributed Product Development of a Mechatronic Product", *International Conference on Machine Automation*, Osaka, Japan, September 2000.

Contact Address

Prof. Steven D. Eppinger and Mr. Vesa Salminen
Massachusetts Institute of Technology
Center for Innovation in Product Development
Cambridge, MA 02142-1347 USA
Email: eppinger@mit.edu vesas@mit.edu

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

IMPLEMENTATION OF PETRI NET BASED DESIGN NETWORK ON THE AUTOMATION OF MECHATRONIC DESIGN

S Güroglu, Z Erden, and A Erden

Keywords: Design automation, descriptive models of the design process, models of design

1 Introduction

Today's highly competitive market conditions require the development of innovative products in shorter time. This requirement necessitates complete automation for the product development stages. There is already a growing research and application trend for automating particularly the manufacturing phase, however, research to automate the design process, that involves creativity and innovation in product development, is rather scarce. The automation of design is to make either the whole design process or some parts of it, independent from human designers through the use of systematic approaches implemented by computer technology.

In order to create a framework to automate the conceptual design stage of the mechatronic design process, a decentralised design inference network model called the Petri Net Design Network (PNDN) has been developed and its basic features and theoretical framework are already published [1]. Various interrelated components based on different physical principles are used in mechatronic systems commonly and they have to be integrated. This inherent decentralised characteristic of the mechatronic systems is modelled functionally through a network architecture and the integration is achieved by the information flow over the network.

Petri Nets, that are bipartite directed multigraphs, have been recognised as one of the most powerful tools for modelling discrete event systems as a representation of individual and interrelated processes in distributed, concurrent dynamic systems [2]. The motivation behind using Petri Nets for design automation is stimulated by their modelling power, a mathematical arsenal supporting the analysis of the modelled systems [3]. This proves to be crucial in the evaluation of design alternatives; for example, inconsistencies are pruned according to occurrences of deadlocks that adversely affect the "liveness" of the network. Reachability of designated subfunctions may be of high importance for a certain design, then, the design alternatives can be and are evaluated accordingly based on the reachability of the required state. We thus achieve the modelling of integrated mechatronic design using Petri Nets that model not only the interrelations between various subfunctions of a mechatronic system but also their integration through the information flow.

2 An Overview of PNDN

The theoretical architecture of PNDN is developed on a functional basis, rather than any physical realization, and therefore it is independent of any specific design problem. In addition, since the interactions between subfunctions of a design artefact are modelled via information flow, characteristics of the PNDN make it a suitable tool for the design of multi-disciplinary systems. In this research, we have selected applications for the Design Network Simulator (DNS) software from the mechatronic design domain.

2.1 Functional representation

A first step in an engineering design procedure may be the functional representation of a candidate system, which satisfies the given requirements after a design need is identified with related design requirements [4]. A systematic way for functional representation of such a system is to establish a functional design tree (FDT), which is a functional decomposition hierarchy that involves subfunctions of systems at various levels of resolution, and where the top most node is to satisfy the required overall function [5].

The concept of functional cells provides a way of symbolic representation for the material, energy and information flow in a system through the execution of subfunctions. It is important to recognize that, functional cells at the first level of decomposition are representational variables (symbols) at the highest level of abstraction. As one proceeds to the lower levels of FDT, functional cells gain precision in their definition due to lower functional resolution such that at the leaves of the tree, Atomic Functional Cells (AFCs) are defined numerically or in a formula-driven formal way representing precise subfunctions with almost precise input/output mappings. This top-down approach results in a transition from an abstract functional representation of the system to be designed to a numerical representation at the lowest level. In the most abstract representation, the only item, which is modeled in transition through the network is the information flow, while for the lower level resolutions, energy, specific material and information items are explicitly described depending on the input-output relations for AFCs.

2.2 Places, transitions and tokens in the PNDN

Subfunctions of an integrated engineering system interact with each other through the flow of information at higher levels. The formal structuring of the PNDN, aims at constructing the representational formalism of subfunctions and the relationships between subfunctions of a system through information flow at an abstraction that reflects the conceptual design stage of the engineering design procedure.

The first step in structuring the design network for conceptual design is the determination of subfunctions at the corresponding level of the FDT and the required flow of information to integrate these subfunctions. Using the subfunctions at the first level of FDT a PNDN is constructed. Once the structuring of PNDN is complete, each subfunction existing in it can be successively replaced by its corresponding PNDN model until the AFC level, which refers to the embodiment design in the general design process, is reached.

In order to create the Petri Net-based design inference network, places (passive components) and transitions (active components) should be determined and identified. In the PNDN, knowledge on subfunctions are representational formalisms considered in some "modeled environments". Information existing in these modeled environments can be "physical information" related with various real environmental conditions (e.g. temperature of the environment) or conditions in the system itself (position of a mobile robot with respect to an object). It can also be in the form of an information about information and this is called as "meta information" (e.g. information about information about the environmental temperature). These modeled environments are places of the PNDN and are represented by the symbol (O). Transitions of the PNDN are defined as the Functional Cells (FCs) at the first decomposition level, because FCs represent processes, transforming information existing in places. Transitions are denoted by the symbol (□). Availability of information in the places of the PNDN is represented by tokens (•). After the determination of places and transitions, it is required to represent the various input-output relationships between them by means of I/O mappings illustrated by directed arrows in the PNDN.

In order to structure the PNDN for the conceptual design phase of a system S, we proceed in three steps:

1. Generation of the variable representation for F(S),

2. Formulation of the hybrid automata model for F(S),

3. Creation of the PNDN for F(S) using the hybrid model.

Let S be the system to be designed for performing an overall function F. Our aim is to develop a model for F(S) to determine what type of behavior S can have at the conceptual design phase, using the formal definition of requirements based on F. In order to create such a model, the following variable representations are required and need to be defined by the designer [6], [7]; Functional State Set (FS), Continuous Variable Set (CVS), Discrete Variable Set (DVS), Functional State Matrix (FSM), Instantiations of the Discrete Variables (DVS_{ins}), Invariant Conditions for Functional States, Instantiations of the Continuous Variables (CVS_{ins}), Discrete State Transition Matrix (DSTM).

The hybrid automata model of F (*Hybrid Model of F (H_F)*) can be created based on these variable representations. The hybrid model of F (H_F) is a 6-tuple [8].

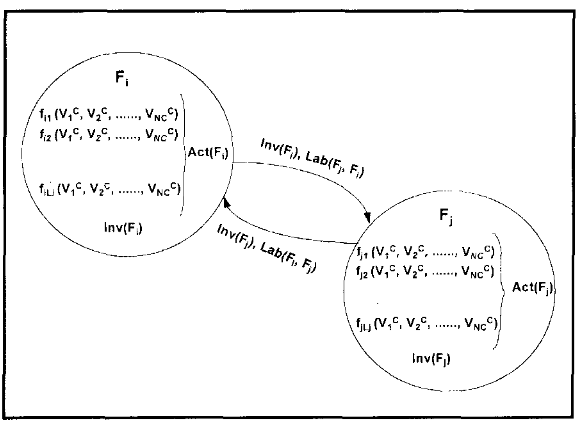


Figure 1 The Hybrid Model of F (H_F).

A finite set **(Loc)** of vertices called *Locations,* a finite set **(Var)** of real-valued *Variables,* a finite set **(Lab)** of *Synchronization Labels,* a labeling function **(Act)** that assigns to each location $F_i \in$ Loc a set of *Activities*, which represent the continuous behavior for F_i , A labeling function **(Inv)** that assigns to each location $F_i \in$ Loc an *Invariant Condition.* A finite set **(Edg)** of edges called *Transitions*.

PNDN is a 4-tuple, and it is defined mathematically as; **PNDN = (P, T, I, O),** where **P** is a finite set of ***places,*** **T** is a finite set of ***transitions,*** **I** is defined as a binary relation ***from P to T*** called the ***input relation,*** finally, **O** is defined as a binary relation ***from T to P*** called the ***output relation.*** Variables and their instantiations are represented by ***tokens*** (•) inserted in the relevant places and information flow in the system is modeled by token flow in the PNDN.

Detailed formulation of the PNDN as a Petri Net using the prior formalism of the problem is already published [1], [6].

As in the case of detail design automation or manufacturing automation, besides the mathematical formalism, automation necessitates the computer implementation of the theory. For this purpose, a software package, named "Design Network Simulator (DNS) Software", is developed.

3 The Design Network Simulator (DNS) software

The main aims behind the development of the software package are the generation of a visual basis that facilitates interchange of ideas on constructed functional design alternatives among several designers, by the flow of tokens the simulation of the information flow on created network and for the elimination of defective designs, having the analysis ability. In addition, this package should give the designer ability of reducing design time and eliminating design flaws.

Starting point in developing a Petri Net based design inference network model applied to mechatronic systems is the functional decomposition of these systems. This means that the model itself is based on the object decomposition. Therefore an object oriented programming tool is the most suitable one for development platform. Hence, the Design Network Simulator (DNS) is developed with Borland C++ Builder, which is an object-oriented, visual programming environment. The elements of the design network, namely; transitions, place sets and switches are components that perform a specific programming task. These components are defined as separate objects in DNS.

The software package algorithms are developed in three parts; the algorithm for design network creation, the algorithm for simulation of deterministic and non-deterministic token flow, and the algorithm for analysis of the network.

In PNDN, dynamic behaviour of design is obtained by modelling the information flow between the network elements. This information flow is represented by the token game. PNDN provides two types of token flow. These are deterministic and non-deterministic token flows. While DNS uses deterministic token flow for the structural examination of the constructed design network, the non-deterministic token flow handles the uncertainties in it. The algorithm for the non-deterministic token flow is given in Figure 2.

The software package developed, namely DNS, with its easy to navigate interface shown in Figure 4, helps the designer for the rapid and easy creation, simulation and evaluation of the design network. In addition, especially for adaptive design, DNS allows the designer to make modifications to previous designs and in so doing reduce the design time.

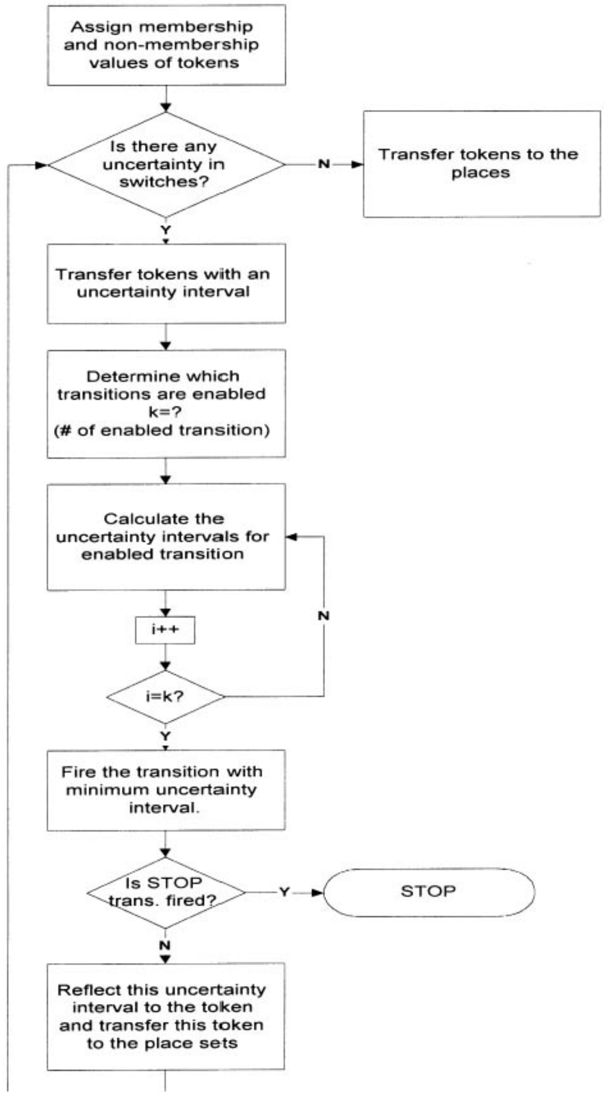


Figure 2. Algorithm of the non-deterministic token flow.

4 Case study

The software package has been applied to the design of 30 mechatronic systems as case studies. Some of these case studies are reverse engineering of already available products. The purpose of reverse engineering is to test the validity of the PNDN model. The other case studies are considered where DNS is used to develop several design alternatives based on a unique design need. Based on these case studies, the software package is evaluated. The overall performance of the software is acceptable and further improvements are under development.

Example: A Mobile Robot

The example presented in this section is the design of a Candle Extinguishing Machine (CEM), which was given as a ME 407 Senior design project in the spring semester of 1998-1999 academic year. The aim was to design a machine, which will find and extinguish 10 candles placed on a table randomly. Once the machine completes a 360° turn on its axis, it is assumed that the candle is detected. After finding, machine moves to the candle until it senses the candle is close enough to be extinguished and extinguishes it. Then the loop will be processed again for other candles [9].

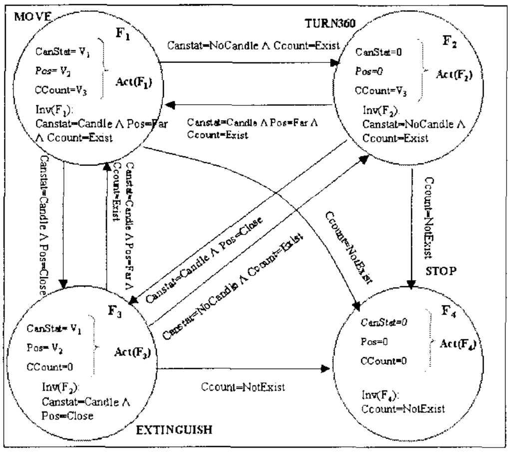


Figure 3. Hybrid Model of F of the Candle Extinguishing Machine, CEM

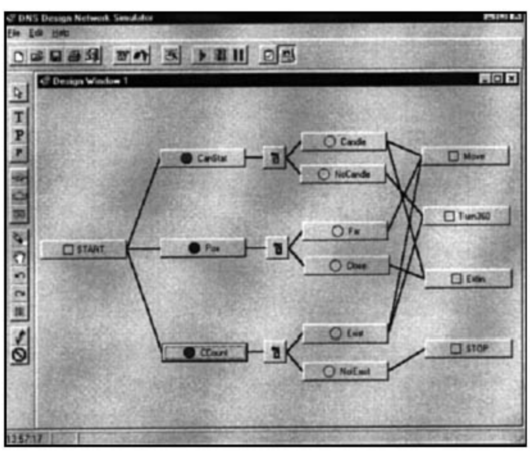


Figure 4. Screen Shot of Design Network Simulator for CEM (CanStat: Candle Status, Pos: Position, Ccount: Candle Counter)

In this example PNDN is structured only for the first level of FDT. At this level of functional decomposition, the functions are MOVE, EXTINGUISH, TURN360°and STOP. The hybrid model shown in Figure 3 is constructed to reveal the relations between the functions and the flow of information on the PNDN variables. The next step is to compose this information into the network structure. At this point, DNS software package provides an easy to navigate interface to designer. The PNDN constructed PNDN structure for CEM is shown in Figure 4.

In addition to network construction, DNS has a deterministic and non- deterministic simulation capability. By means of this feature, the flow of information on the network and execution of functional states can be observed. The non deterministic token flow feature gives the designer the chance of handling uncertainties in signal and processes propagating and changing over the network by representing any token state in PNDN with an intuitionistic fuzzy proposition [10].

DNS is designed to warn the designer in almost all possible user oriented errors in the creation of PNDN or the simulation. Moreover the reachability and liveness definitions for Petri Nets are used in the analysis of created network by checking all available marking conditions in token flow.

After these construction, simulation and analysis steps, DNS software package shows that the PNDN created for CEM is live and deadlock free. Therefore the conceptual design phase for CEM is completed.

5 Conclusion

A decentralised design inference network architecture based on functional decomposition of mechatronic systems and information flow in these systems has been developed by using Petri Net theory. The theoretical model is implemented in a computer code, called Design Network Simulator (DNS). It is a specially developed software to implement PNDN for modeling mechatronic systems. DNS is an object oriented software package, which is a distinguishing feature that makes DNS a flexible visual tool for mechatronic design modeling.

This tool allows the designer to evaluate a number of design alternatives in a short time and helps the designer eliminate design flaws. By checking the reachability and controlling whether there exists any unreachable functional state or not by investigating all marking combinations in token flow; DNS performs an effective analysis of the design model. Moreover, DNS enables visual representations that are generated for functional design alternatives. This facilitates interchange of ideas among several designers on a visual base as well as on a measure-based evaluation.

30 mechatronic systems have been modeled using the DNS software package and simulations of these models have been performed to test the applicability of the algorithms to real life design problems. It is observed that the software does not require expertise for its user interface and allows a rapid development of the system model. The software also proved sufficient on locating errors caused by the improper network construction and/or the design model itself.

Ongoing research focuses on reaching the embodiment design stage through the implementation of a PNDN model to the lower levels of FDT. With the addition of this feature to DNS, we are planning to obtain a powerful tool for the automation of conceptual design stage in the engineering design procedure.

References

[1] Erden, Z., Erkmen, A. M. and Erden, A., 1998, "A Petri Net-Based Design Network with Applications to Mechatronic Systems", SDPS Transactions, Journal of Integrated Design and Process Science, Vol.2, No.3, pp.32-48.

[2] Andreadakis, S. K., 1988, Analysis and Synthesis of Decision Making Organisations, Ph. D. Thesis, MIT, USA.

[3] Erden, Z., Erkmen, A. M., Erden, A., 1997, "Modeling Job-Resource Interactions in Mechatronic Design Process by Petri Nets", International Conference on Engineering, ICED'97, Tampere, Vol. 2, pp 753-756.

[4] Pahl, G., Beitz, W., 1988, "Engineering Design A Systematic Approach", Springer-Verlag, UK

[5] Erden, Z., Erkmen, A. M. and Erden, A., 1996, "Generation of Functional Cells for a Mechatronic Design Network", Proc. of the 5th UK Mechatronics Forum Int. Conf. (Mechatronics '96) with the 3rd Int. Conf. on Mechatronics and Machine Vision in Practice (M^2VIP '96), Vol.1, pp.233-238, Guimaraes, Portugal.

[6] Erden, Z., 1999, A Petri Net-Based Inference Network for Design Automation at Functional Level Applied to Mechatronic Systems, Ph.D. Thesis, METU, TURKEY.

[7] Erden, A, 1997, "A Experience with Capstone Design Projects at Senior Level Engineering Design Education", Proc. Int. Conference on Engineering Design ICED'97 Tampere pp. 411-414.

[8] Alur, R. et al., 1994, ''The Algorithmic Analysis of Hybrid Systems'', Proc. 11th International Conf. on Analysis and Optimization of Discrete Event Systems, Lecture Notes in Control and Information Sciences 199, Springer-Verlag, pp.331-351.

[9] Güroğlu, S., 1999, Implementation of An Algorithm For a Petri Net Based Design Inference Network, MSc. Thesis, METU, TURKEY.

[10] Erden, Z., Erkmen, A. M. and Erden, A., 1999, "Handling Uncertainty in Design Automation Using Intuitionistic Fuzzy Propositions", Proc. the 12th Int. Conference on Engineering Design, ICED 99, Munich, Germany.

Serkan Güroğlu
Middle East Technical University, Department of Mechanical Engineering
İnönü Bulvarı 06531
Ankara, TURKEY

Telephone: +90 312 210 52 09
Fax: +90 312 210 13 29
E-mail: serkan@design.me.metu.edu.tr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

MODEL OF THE DESIGN PROCESS – A SYSTEM APPROACH FOR CREATING PROTECTABLE SOLUTIONS

J Käosaar

Keywords: Descriptive models of the design process, process modelling, human creativity, problem solving techniques.

1 Introduction

F.Taylor, who wrote in 1914, vividly describes a design process at the beginning of the 20th century: "Technics as a science has emerged so recently that I remember the times when all manufacturers were very suspicious of an educated scientist-engineer. The calculation of details and design were carried out not based on strength theory but relying on the individual opinions of people who had a good eye. People who could remember the dimensions and general shapes of particular machine parts and guess the required dimensions and strength of parts did it. The majority of the drawings of a new machine were sketched in chalk on the floor of a workshop or by a stick on the earth floor of a smithy and quite often an oral explanation was more important than a drawing. An engineer did not know exactly what he wanted before a model showed the shape a machine was going to have. When the design of a new machine was completed, the engineer usually said "Now that the machine is ready, we shall start it and see why it does not work". And so did not the uneducated, but the best engineers of those days" [1].

In the middle of the 20th century it was understood that primary attention to the designer as an original source of technical creation should be paid, and his thinking and creative ability should be activated. In response to this demand, an organisation called Workshop Design Konstruktion (WDK) was founded in 1978. This International Society for the Science of Engineering Design is an informal and international network of people interested in advancing knowledge about engineering design. Members of WDK are scientists, engineers and educators who take an active part in the investigations, meetings and publications of the society. The main goal of WDK is to rationalise engineering design work, and to bring together the world community of engineering designers, both theoretical and practical [2].

2 A system approach for creating protectable solutions

The development of new technical products can be described by using a scheme "idea generation - ideas - engineering design - drawing - production - product", which has evolved from the history of engineering design worked out by me on the basis of the development of graphic expression of a technical ideas [3].

The last part of the scheme "drawing - production - product", i.e. the issue of improving the quality and speed of the production process based on the present drawings has been sufficiently studied. The first part "idea generation - ideas - engineering design - drawing" has been less examined.

The creation of new technical products enfolds a number of scientific and technical disciplines, and many factors, which influence both the designer and the designed object, must be considered. The designer's knowledge must embrace various areas of science and technics, he must be technically well prepared and master the protection of industrial property and the methods of problem solving and engineering design.

Following from the preceding, I shall look at the design engineer as the main component of the system in creating protectable technical solutions. The system is composed of parts, their characteristics and their mutual relations. The parts are the input, the process, the output, as well as feedback and the restrictions.

Figure 1. Shows the main scheme of a "system approach for creating protectable solutions"

Objective world (Block 1)	1. Objective information 1.1. Scientific and technical disciplines - Experience of engineering design - Methods of problem solving and engineering design - Protection of industrial property 1.2. Customer 1.3. Technical aids

The design engineer (subjective world) (Block 2)	2.1. Discursive and intuitive thinking 2.2. Subjective information 2.3. Designer's prerequisites and talent 2.4. Designer's experience 2.5. Engineering design (idea generation - ideas - engineering design - drawing – production - product)

Figure 1. A system approach for creating protectable solutions

The subjective information, which directly involves the thinking process and engineering design of the designer, must be examined in conjunction with the objective information. The input for the designer is the order with its concurrent information, which is necessary for the development of the process. If the designer does not find suitable solutions, he shall process new objective information to create protectable products.

The design engineer's work and the results of his work depend to a large extent on the variety of subjects and means of teaching we use for training designers, as well as on the organisation of discursive and intuitive thinking process of the designer.

It does not suffice to have theory only; the latter has to be proved in practice.

The design process transforms the input into the output. In the given case the process involves thinking, i.e. the generation of ideas on the basis of discursive and intuitive thinking, and specific work of the design engineer in preparing sketches, drawings and project documentation etc.

The final output is the requested project documentation for creation of a certain product. The solutions on the sketches, drawings, schemes, etc. are intermediary solutions, where the "crystallisation process" of the final output takes place.

The ability to transform the given input into a specific output is a characteristic of the given system. We are interested in the following characteristic features of the main part of the system - the design engineer: which inputs are needed to reach the expected outputs both in the education of design engineers and in their future work. Thus, what are the new subjects that the design engineer must study and what is the relevant information for the design engineer and when do they need it etc.

The development of the process is determined by various interactions, i.e. the output of one process is the input of another process. There are links between various blocks during an engineering process and after it has been finished, i.e. the completed project documentation and the products based on it enrich both the subjective as well as the objective information, while being the source for new technical ideas and contradictions.

The feedback secures the coherence between the expected and the real output.

The restrictions, on the one hand, are related to the order, which determines the conditions the designed product has to conform with, but on the other hand, limited by the existing level and possibilities of engineering and technology.

Every system consists of subsystems on different levels. The higher level of the system is the design engineer himself and his activity, i.e. the designer's thinking on different levels - conscious, sub-conscious and unconscious [4].

The subsystems on the following different levels are for example the design engineer and dialectics, design engineer and logic, design engineer and psychology, design engineer and methods of problem solving, design engineer and industrial property etc. - consequently all the disciplines on which the design engineer's thinking is based.

The system must be developed as a whole realm of human knowledge - the design science together with its various subsystems.

The system is based on the following principles:

- discovering the talent of a design engineer, enhancing his abilities and involving the scientific bases of thinking in educating the design engineer;
- teaching problem solving methods for idea generation by using discursive and intuitive thinking;
- forecasting new ideas and designs by retrospective analysis of technical solutions;
- teaching industrial property to get protectable solutions.

3 Block 1

Block 1 represents the objective world, where the designer works.

1. Objective information

Objective information is the information, which exists outside the design engineer; a part of this becomes subjective information after being acquired by the design engineer.

1.1. Scientific and technical disciplines

There are disciplines not taught in all technical universities, but without which the work of a design engineer at present and in the future is ineffective. Such disciplines include for example dialectics, logic, psychology, decision making, methods of problem solving, industrial property etc.

3.1 Methods of problem solving and design and the method of trial and error as a basis of all human activity.

For understanding the nature of thinking as an activity upon problem solving, I shall examine the neurophysiological bases of human activities, whereas following the principle of trial and error I shall proceed from the functional system, which was created by P.Anohhin (Figure 2) in 1935 and continuously improved later on [5].

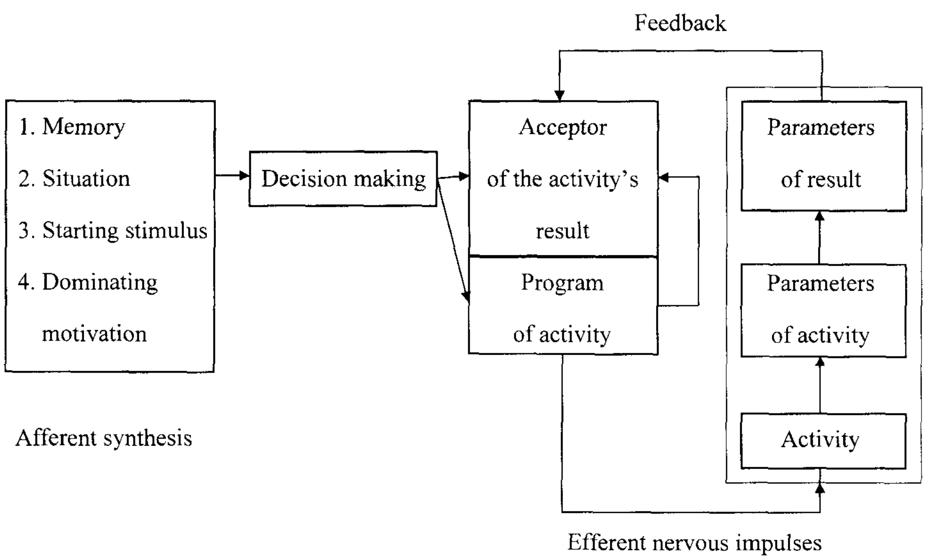


Figure 2. The functional system

An activity begins in the phase of an afferent synthesis (afferent – bringing or carrying towards, a nerve conducting the nervous impulse to the brain). In this phase the following afferents are simultaneously synthesised: dominating motivation and the situation at the moment, starting stimulus and memory. Thus, the afferent synthesis conducts an organism to the problem solving; i.e. what kind of result the organism has to achieve at the given moment and provides a target to which the whole logic of a system is dedicated. At the end of the afferent synthesis a decision is made, where the most useful activity at the moment is chosen. The acceptor of the activity's result (acceptor – the one that joins the activity) foresees the activity's result, which has to be achieved by a selection in the course of deciding and therefore anticipates the course of events between the organism and the outer world. The acceptor of the activity's result is a complicated apparatus. It does not only forecast the result, but it also has to compare it with the result achieved by feedback. It is this very apparatus, which provides the organism with an opportunity to correct the mistakes of an activity and finish the incomplete activity. In this way searches and corrections to get a useful result take place through the feedback. The whole chain "result - feedback - comparison and appraisal in the acceptor of the activity's result - correction – new result " takes place within a part of a second. If the result is dissatisfying, a new afferent synthesis will take place. Thus, after several trials and errors, a result satisfying the system will be achieved. The thinking of a design engineer should work in the analogue way.

3.2 Industrial property – legal protection of patents and utility models

Original technical solutions may and must occur at each stage of the long and complicated process of design, therefore it is very important to know well the possibilities for the protection of original solutions as patents or utility models. In the first stages of a design, before materialising the construction, the protectability of the developed construction must be carefully considered. The designer must foresee the protection of the construction, attacks from competitors, as well as possibilities for avoiding the competitor's protected construction.

A research by Ford Motor Company reveals that only 5% of the final price of a product are spent on design, whereas 70% of decisions made at this stage influence directly the final price of the product. A comparison of the design process of new car models of the car producers of Japan and USA shows that Japanese companies make more changes at the initial stages of design than the Americans, and therefore the average time of implementation of a new car model is 3,5 years in Japan, and 5 years in America [6].

Thus, the decisions made during the design process influence most the formation of product price and the time of implementation.

As it is the fastest, easiest and cheapest way to make changes to a construction on paper or on a computer screen and in case of test models or series manufacture it is much more expensive to change the construction, the search of technical solutions that can be protected and the protection of these is important from the earliest stages of design.

At the protection of technical solutions, the following problems deserve special attention [7]:

- is it pioneer invention, i.e. an invention which has no analogues;
- is it the case about developing a new construction or elaborating an existing one;
- is it a utility model or a patented technical solution;

- is it possible to switch over to a patent from a utility model and vice versa;
- is the construction already existing and protected and is it necessary to avoid it (for example, a well-sold product with good image that can be used);
- is the product meant for internal or external market, and in this connection, does the construction require protection in one's own county, in different countries or whether the assistance of the European Patent Convention or the Patent Cooperation Treaty has to be used.

4 Block 2

Block 2 represents the designer with his subjective world

4.1 Discursive and intuitive thinking

Formerly, e.g. in the former Soviet Union, most of creative solutions were attributed to discursive thinking, but today the major role of intuitive thinking in the creation of protectable solutions is acknowledged.

A great role in solving new problems is played by the subconscious, which involuntary stores a part of information and participates in intuitive problem solving. The designer receives 95% of new information through visual perception and, as figures are the language of technics, graphic information has a significant part.

The unity of intuitive and discursive thinking is the mechanism of creative intellectual activity. Their co-operation takes place in smooth transitions between hierarchical structural levels. The first stage is most intuitive and the last is most logical [8].

I transferred the above-mentioned mechanism to the designer's field. If the human experience contains ready-made logical programs, then problem solution is carried out on the level of logic, i.e. standard problem solving. In this case the designer will turn for typical solutions to his memory, reference literature, patents, etc. adjusting the existing constructions to the problem at hand. If there is a lack of correspondence between subjective logic and objective connections, then problem solving becomes creative. At such point solutions can be found by intuition. It is natural, as the problem is in fact unsolved, but all available knowledge has been exhausted. The solution may be "prompted" only by objective logic and in the more straightforward case by the things themselves. The designer tries to modify the solutions found in the information search using his intuition.

The organisation of human activity descends to lower, i.e. intuitive levels. In solving a creative task, designer first makes use of his consciously organised experience. If such experience is accumulated in the course of a particular action - the unconscious experience is applied. Such experience sometimes holds the key to a creative task. The unconscious experience manifests itself at a happy moment as an unexpected "prompt" leading to an intuitive solution.

A logical solution of a creative task happens intuitively, i.e. when the task has actually been solved. There is the need to logically convey the solution found intuitively to other people, to explain and approve the correctness of the solution, and to use it for the solution of more complex problems of similar type.

4.2. Subjective information

- basic knowledge acquired from theoretical disciplines,
- information derived from memory,
- information obtained by direct specific activities of a design engineer (experience).

Rapid access to new information is very vital to every designer. Too early and too late access is both inefficient. The influence of new information is dependent on the thesaurus of the designer (vocabulary and knowledge stored in the memory of the subject). When the thesaurus is small, the designer is unable to understand information; if the thesaurus is large, the information obtained may be minimal. At this point the dependence on design experience is obvious. Thus it is important that new information should be provided at the right time and is accessible and understandable in view of the thesaurus of the designer [9].

4.3. Designer's prerequisites and talent

The prerequisites of a future designer (prerequisite characteristics, aptitudes, and talents) have to be ascertained already in the childhood (in the young age). Abilities have to be developed in the course of educating and teaching.

4.4. Designer's experience

Any kind of specific experience reduces the inertia of thinking, enables a designer to think a lot wider, but general experience contributes to the creation of new connections and widens the field of possible solutions. The inertia of thinking retards creation.

The wider the experience of the designer, the bigger is the role of the intuitive problem solving.

Up to recently, both general and specific technical subjects have been taught at technical universities. The scientific bases of a designer's thinking process are already on such a level that we can explain this activity to a certain extent and we know which are the necessary supplementary subjects for design engineers in order to speed up engineering process and develop protectable design solutions.

5 Conclusions

The present paper describes the author's system involving those subjects and interactions, which were previously not considered essential for the training of designers and the development of engineering design. These subjects help to organise the discursive and intuitive thinking of the designer, necessary for achieving better and faster output and shortening the time spent for creating protectable solutions.

"The most important, and indeed the truly unique, contribution of management in the 20th century was the fifty-fold increase in the productivity of the MANUAL WORKER in manufacturing. The most important contribution management needs to make in the 21st century is to similarly increase the productivity of KNOWLEDGE WORK and the

KNOWLEDGE WORKER. The most valuable assets of a 20th-century company were its *production equipment*. The most valuable asset of a 21st-century institution, whether business or nonbusiness, will be its *knowledge* workers and their *productivity*" [10].

Based on the quotation above, the productivity of design engineers in the development of protectable design solutions should be increased by a half in the 21st century, which is not an easy task because the science of engineering design emerged less than a half of a century ago.

References

[1] Gokun, B.V., "Bases of technical development of manufactory in machine building", Moscow, 1960, pp.130 (in Russian)

[2] "Engineering Design and Creativity", WDK 24, Edition HEURISTA, 1996, preface

[3] Käosaar J., "Historical development of engineering design", M.Sc. thesis, Tallinn Technical University, 1992 (in Estonian)

[4] Hubka, V. and Eder, W.E., "Design Science", Springer, 1996, pp. 23

[5] Anohhin, P.K., "Survey of the physiology of functional systems", Moscow, 1975, pp. 44–45 (in Russian)

[6] Ulman, D.G., "The mechanical design process", McGraw-Hill, Inc., 1992, pp.12

[7] Käosaar, J. and Märtson, I., "Utility model – an essential part of product development", INTERNATIONAL DAAAM PROCEEDINGS OF THE 2nd INTERNATIONAL CONFERENCE 2000, ESTONIA, Tallinn, 2000, pp. 77 - 80

[8] Ponomarev, Y.A., "Development of psychological organisation of intellectual activity", Principles of development in psychology. Nauka, Moscow, 1978, pp.117 (in Russian)

[9] Käosaar, J . and Märtson, I., "Product designer, patent information and brainstorming", PROCEEDINGS, OST-99 SYMPOSIUM on Machine Design, Kungl Tekniska Högskolan, Stockholm, 1999, pp.13 - 19

[10] Drucker, P.F., "Management Challenges for the 21st Century", HarperBusiness, 1999, pp. 135

Jüri Käosaar,
Tallinn Technical University, Ph.D. candidate, (Patent Bureau Käosaar & Co), Riia Str. 185, 51014, Tartu, Estonia, Tel: 372 7 383 051, Fax: 372 7 383 055, E-mail: jyrik@park.tartu.ee

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A GENERAL FRAMEWORK FOR MODELLING OF SYNTHESIS – INTEGRATION OF THEORIES OF SYNTHESIS

H Takeda, M Yoshioka, and T Tomiyama

Keywords: design theory, knowledge-based engineering, descriptive models of design process, abduction, model-based reasoning

1 Introduction

Engineering design consists of a variety of thought processes such as analysis, synthesis, problem-solving, and decision-making. Among these, the most crucial process in design is "synthesis" or "synthesis-oriented thought process," because synthesis brings about creativity of design. In spite of the importance of synthesis, compared with analysis, synthesis is less understood and codified as a model.

In this paper, first we clarify the roles of synthesis in design, and then logically define design as a synthesis-oriented thought process in a logical framework as a reasoning process. The core part of the synthesis-oriented thought process is performed by abduction, but the overall process is logically realized by combination of abduction and deduction. Next, synthesis and analysis are formally distinguished.

Second, we introduce the object-dependent (or model-based) approach that has a power to naturally represent objects in the physical world. This combination of logical and object-oriented approaches is commonly used to empower a knowledge representation scheme for the sake of formality and expressiveness. Logic describes inter-model relations and general operations to models, while models describe the nature of design objects and their possible operations. We will redefine synthesis to deal with knowledge about design objects by introducing the concept of model-based reasoning in which modeling operations perform references to an extra-logical world in asserting logical formulae. In our formalism of combination of logical and model-based approaches, this explicit reference to non-logical facts in extra-logical (i.e., model-based) world naturally integrates logic and models.

Based on this concept, we introduce and operate a variety of knowledge on objects in an integrated way in a. reasoning framework for synthesis. The framework has the following three features, viz., duality of abduction and deduction, multiple viewpoints, and duality of logical and model-based reasoning. We also illustrate a prototype system of the reasoning framework of synthesis.

2 Synthesis and analysis

Synthesis can be identified in such activities as scientific discovery, design, and art including writing novels and painting pictures. While synthesis is obviously the core of these activities, its nature is almost unclear. This chapter overviews how we deal with synthesis and defines our approach to synthesis.

Synthesis is often counterposed to analysis and defined as an opposition of analysis, while analysis is defined independently of synthesis. Synthesis and analysis often collocate, but the patterns of occurrence of synthesis and analysis are different. For instance, in scientific discovery, synthesis very often appears after analysis. In contrast, synthesis and analysis appear alternately and repeatedly during design.

Synthesis and analysis are also often discussed logically. A typical interpretation is that analysis is deductive while synthesis is non-deductive; according to Peirce [1], synthesis is often explained by abduction. However, this correspondence might be confusing, because analysis (or synthesis) as human activities can contain activities other than purely logical reasoning. To clarify this, we distinguish *analysis* (or synthesis) *as human activities* from *analysis* (or *synthesis*) *oriented thought process*. Abduction and deduction are mainly used to refer to reasoning processes.

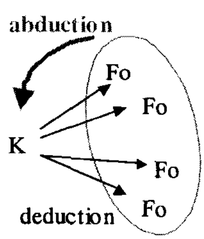
3 Logical formalization of synthesis

To logically deal with synthesis and analysis, we assume that synthesis and analysis are *rational thought processes based on theories*. Here a *theory* is a set of logical correspondence relations (among axioms, facts, and theorems) that gives explanations for a phenomenon described in facts based on axioms and theorems included in the theory. A *thought process based on theories* is a reasoning process in which theories are used to find axioms or theorems that explain given phenomena or to find phenomena as exemplars of axioms and theorems, and *rationality* of any given process means consistency of the statement to theories.

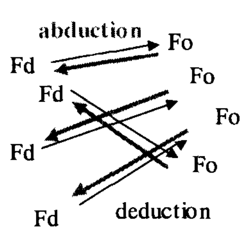
Then, analysis and synthesis can be logically associated with deduction and abduction [2]. First, we consider the following formula.

$\boldsymbol{A} \vdash \boldsymbol{Th}$

This formula means that under axioms $\boldsymbol{A}$, a set of theorems $\boldsymbol{Th}$ is proven. In this formula, finding theorems from the axioms is deduction, whereas finding axioms from given theorems is abduction. Axioms form the basis of a theory that can explain phenomena and facts, whereas $\boldsymbol{Th}$ consists phenomena and facts that are observed.



(a) Synthesis for generality



(b) Synthesis for individuality

Figure 1: Pattern of Process for Synthsis

Since we usually have distinctions between knowledge and facts, we divide axioms into defined facts ***Fd*** (that should be given prior to reasoning without explanation) and knowledge ***K***.

$$K \cup Fd \vdash Th$$

In contrast to ***Fd***, we can identify facts that should appear in theorem ***Th*** and be explained. They are called *observable facts* ***Fo***. With this distinction of ***Fd*** and ***Fo***, we can categorize abduction more precisely.

1. Finding ***K*** and ***Fd*** from a part of ***Th*** (=***Fo***)
2. Finding ***K*** from a part of ***Th*** (=***Fo***) and ***Fd***
3. Finding ***Fd*** from a part of ***Th*** (=***Fo***) and ***K***

These three different types of abduction play their own roles in different thought processes. For example, a scientific discovery process aims at obtaining knowledge ***K*** that should be *general* and therefore minimal, while given is a part of ***Th*** (=***Fo***) that is *individual* (see Figure 1(a)). Abduction in scientific discovery is types 1 or 2.Hypothetical knowledge (i.e., ***K***) proposed by abduction should be tested against observable facts ***Fo*** and this implies that deduction should be performed more often than abduction.

This stands in a sharp contrast to design in which most of synthesis is abduction of type 3. In a design process, the target is defined facts ***Fd*** that are *individual*, and given are knowledge ***K***

(a) Analysis-oriented Thought Process

(1) Observation of phenomena
A phenomenon is observed as observations ***O***.

(2) Extraction of facts
Observed facts ***Fo*** are extracted from ***O***.

(3) Formation of hypotheses or selection of axioms
Fo can be used to reason out hypothetical axioms ***Kh***. In obvious cases, a set of known axioms ***Ke*** is selected instead.

(4) Assuming definition facts
Initial definition facts ***Fd*** are assumed. Together with ***Ke*** (or ***Kh***), this will be used to derive theorems ***Th***. Usually, ***Fd*** contain such known facts as boundary conditions and initial conditions.

(5) Derivation of theorems from axioms
Theorems ***Th*** are derived from ***Ke*** (or ***Kh***) and ***Fd*** deductively. It may break down the original problem (i.e., derivation of theorems) into smaller subproblems (the "divide-and-conquer strategy").

(6) Verification of theorems against facts
The derived theorems ***Th*** are tested against the observed facts ***Fo*** to check the explicability of the theorems. If $Th \supseteq Fo$, this test is satisfied. Then the theorems are said to explain the extracted facts and the choice of ***Ke*** (or ***Kh***) was appropriate. If $Th = Fo$, then ***Ke*** is complete. If $Th \supset Fo$, then $Th - Fo$ signifies unobserved facts or undiscovered facts in the future or past. If $Fo - Th \neq \emptyset$, then unexplained facts remain.

(7) Verification of theorems against other known axioms
The derived theorems ***Th*** are again tested against other known sets of axioms ***K'***. This test verifies if the theorems are compatible with ***K'*** or at least if they do not violate ***K'***. If the hypotheses obtained in step (3) pass tests (6) and (7), they become axioms.

(b) Synthesis-oriented Thought Process

(1) Describing requirements
Requirements for the synthesis ***R*** are described as theorems.

(2) Extraction of requirements of interest
From ***R***, we only focus on interesting facts as ***Fo***.

(3) Selection of axioms
Axiom to be used is selected. Synthesis requires, various viewpoints to be considered. This means that the number or cardinality of ***K*** tends to be large.

(4) Derivation of solutions from requirements and axioms
Solutions ***Fd*** are derived as facts from ***K*** and ***Fo***. The basic reasoning is abduction logically, but other algorithms to arrive at solutions can be also used. The "divide-and-conquer strategy" might be used, but since the number (or cardinality) of ***K*** could be larger than analysis, trade-off and negotiation among different solutions are important.

(5) Derivation of theorems from axioms and facts
Theorems ***Th*** are derived from ***K*** and ***Fd*** deductively. This is the same as in the analysis oriented thought process. Deduction and the divide-and-conquer strategy are central.

(6) Verification of theorems against requirements
The derived theorems ***Th*** are tested against the requirements of interest ***Fo*** to check if the derived ***Th*** subsume the initial requirements ***Fo***; (i.e., $Th \supseteq Fo$). By doing so, we can check if the solutions ***Fd*** are satisfactory.

(7) Verification of theorems against other known axioms
The derived theorems are again tested against other known sets of axioms ***K'***. This test verifies if ***Fd*** (and accordingly ***Fo***) is compatible with not only ***K*** but also ***K'***.

Figure 2: Formalization of synthesis- and analysis-oriented thought processes

and observable facts ***Fo*** that is a part of ***Th***. Abduction must be performed as many as deduction (see Figure 1(b)), because both hypothetical facts (i.e., ***Fd***) and observable facts (***Fo***) are individual. Knowledge is also different in these two processes. While minimum knowledge ***K*** is desired in the scientific discovery process, a variety of knowledge ***K*** is required for design, because a variety of defined facts ***Fd*** need to be found by abduction.

From the discussions above, we can now formalize the synthesis-oriented and analysis-orient thought processes in Figure 2 [3]. While the analysis-oriented thought process (AOTP) is deduction-dominant, the synthesis-oriented thought process (SOTP) emphasizes the role of abduction. We can characterize SOTP in the following three dimensions in comparison with AOTP.

(1) Arbitrariness in problem definition: Enumeration of the requirements is less constraining than observation.

(2) Arbitrariness and complexity of viewpoints: Fewer axioms are preferred in AOTP while more axioms are preferred in SOTP.

(3) Complicated relationship between abduction and deduction: Both AOTP and SOTP need abduction and deduction. For instance, AOTP needs abduction to form hypotheses, while SOTP needs deduction to derive theorems in testing the facts.

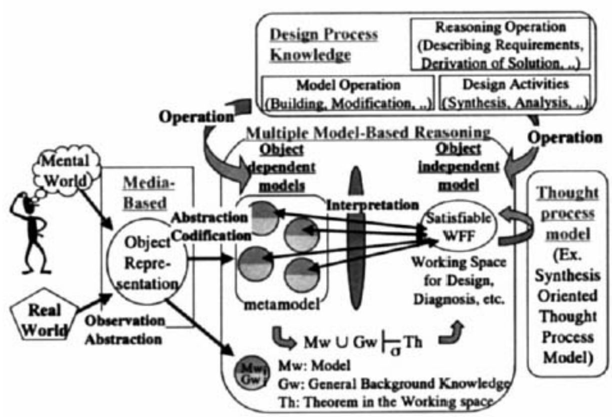


Figure 3: A computational Framework of Synthesis

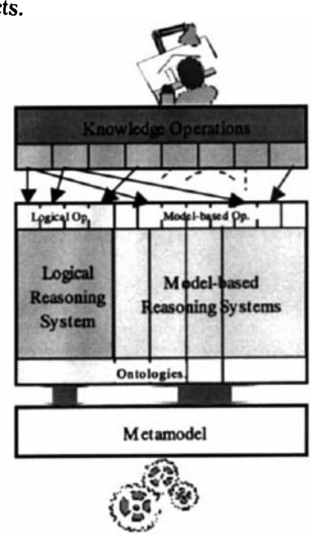


Figure 4: Two-way Integration of logical and model-based aspects

4 A computational framework for synthesis

The above discussion leads to the following two requirements for the framework to model synthesis.

1) Explicit controls for abductive and deductive reasoning processes over facts and knowledge are needed.

2) A variety of knowledge should be provided.

We propose a *computational framework of synthesis* to satisfy these requirements (see Figure 3). It has two main components; an object-independent model or a logical inference workspace in which abduction and deduction are explicitly controlled, and object-dependent models or a model-based reasoning workspace in which rich modeling knowledge is provided. The former is introduced to satisfy the first requirement, and the latter the second. Knowledge in a logical form has a clear syntax and therefore a sufficient computational power, but it lacks semantics that is fulfilled in real world. Model-based reasoning can offer such knowledge that is usually difficult to describe in logic.

The logical and model-based approaches are integrated as follows (see Figure 4). First, the metamodel mechanism manages multiple model-based reasoning systems and submits information about design objects to a common, logical workspace. This ensures smooth transfer of design object information between logical representation and models. Second, the designer's activities are represented by a set of knowledge operations that are decomposed into logical operations and modeling operations. The logical operations operate the logical inference workspace, while the modeling operations manipulate design object models through the metamodel mechanism. The multiple model-based reasoning performs assertion of logical formula in the logical workspace. In other words, a logical formula in the logical workspace is given a truth value only by referring to extra-logical models, if they are satisfiable. (Mathematically, this is called *model-theoretic* view as opposed to *proof-theoretic* view.) This explicit reference to non-logical models naturally integrates logic and models.

5 Knowledge operations

This paper is based on a knowledge-centric view of design activities. We provide a set of operations in the knowledge level to describe designers' activities. These knowledge operations are decomposed into logical and modeling operations, so that they become computable assuming the reasoning framework depicted in Figure 3. For example, *analysis activity* is a typical use of knowledge, and *activity of exchanging information with other designers* is a knowledge handling activity for knowledge acquisition.

As operations in the logical reasoning system, we define the following operations. First, we provide two types of object-level reasoning based on object knowledge; i.e., (l-1) deduction of properties of objects from objects (design solutions), and (l-2) abduction of objects from properties of objects. Here, requirement specifications are included in the properties of objects. Second, we also define meta-level reasoning as one up higher operations with operands such as objects and properties of objects in the object level; i.e., (l-3) setting of objects, (l-4) setting of requirement specifications, (l-5) setting of design knowledge, (l-6) consistency checking of knowledge, and (l-7) operations on the current set of design knowledge.

We provide the following eight operations as modeling operations. Among these eight, five operations concern individual models; i.e., (m-1) building a model, (m-2) reasoning with the model, (m-3) modification of knowledge, (m-4) modification of the model, and (m-5) reference to the model. The rest three are operations for maintenance of multiple models; i.e., (m-6) introduction of model-based systems, (m-7) selection of model-based systems, and (m-8) maintenance of consistency among different models.

We define the following seven knowledge operations as combinations of logical and modeling operations. First, to manage knowledge, there are five operations; i.e., (k-1) knowledge/information acquisition, (k-2) knowledge/information reorganization, (k-3) information confirmation, (k-4) conflict resolution, and (k-5) knowledge/information revision. Second, to utilize knowledge, there are two operations; i.e., (k-6) solution synthesis and (k-7) object analysis. Below, we describe these knowledge operations in detail.

(k-1) *Knowledge/Information acquisition:* The objective is to acquire knowledge and information related to the problem. There are two types of knowledge acquisition. One is to introduce a knowledge source and is formalized as introducing a new model-based system (m-6). The other is to add a new piece of knowledge to a particular model-based system and is formalized as modification of the knowledge base of a model-based system (m-3). The corresponding logical operation is the operation on the current set of design knowledge (l-7).

(k-2) *Knowledge/Information reorganization*: The objective is to reorganize knowledge and information for a task. Knowledge reorganization is a process to reorganize and maintain a set of model-based systems that are used for a task. So, this process is formalized as selection of (one or more) model-based systems (m-7) or maintenance of models in different model-based systems (m-8). The corresponding logical operation is an operation on the current set of design knowledge (l-7).

(k-3) *Information confirmation*: The objective is to confirm information in one knowledge source by testing it against another information source. In this operation, the designer confirms information in the logical level through mappings between models in the object dependent level and the logical level. The metamodel mechanism should provide such mappings and it is formalized as reference to the model (m-5), and setting of design knowledge (l-5), and consistency checking of knowledge (l-6).

(k-4) *Conflict resolution*: The objective is to resolve conflict among different model-based systems; for example, one modeling system says that value of an attribute of the design object should be 10, while another says 10.5. The logical reasoning system detects this type of conflicts by integrating mode-based systems and solves by modifying relationship among different mode-based systems or models in particular model-based systems (see k-5). It corresponds to maintenance of consistency among different models (m-8) and consistency checking of knowledge (l-6).

(k-5) *Knowledge/Information revision*: The objective is to revise knowledge or information to keep consistency with multiple models. It corresponds to modification of knowledge in model-based systems (m-3) or modification of models (m-4), setting of design knowledge (l-5) and setting of requirement specifications (l-4).

(k-6) *Solution synthesis*: It is synthesis in the narrow sense or abduction, i.e., to suggest a new solution for the problem. Following SOTP in Figure 2 (b), first, the selection of axioms for synthesis is formalized as selection of a model-based system (m-7) and description of the problem is formalized as building a model (m-1). Then, the designer proposes new solution candidates by reasoning about a model (m-2). From the logical aspect, it corresponds to setting of requirement specification (l-4), setting of design knowledge (l-5), and abduction of objects from properties of objects (l-2).

1. Derive the neighborhood system of a solution candidate from one axiom set A_1.
   a. Set requirements that can be treated by axioms A_1 as theory (Th_1) in formula (1).
      i. $A_1' \cup F_1 \vdash_\sigma Th_1$
      ii. $\{e_1 \rightarrow p_1, e_1 \rightarrow p_2, .., e_2 \rightarrow f_1, e_2 \rightarrow f_2, ..\} \cup F_1 \vdash_\sigma \{p_1, .., f_1, ..\}$
      where e_i is an abstract entity concept, p_i is an attribute concept, and f_i is a function concept.
   b. Derive $F_1=\{e_1, e_2, \ldots\}$ by abduction (purely logically) with the closed world assumption.
   c. Analyze the neighborhood system of F_1 in the attribute space and the function space by deduction using a modeler that corresponds to Axioms A_1. This will enrich Th_i.
2. Apply previous procedures with another set of axioms and make the attribute information and function information richer.
3. Compute $F_n \cap F_{n+1} \cap \ldots$ for narrowing the solution space to reach a solution.

Figure 5: Algorithm for model-based abduction

(k-7) *Object analysis*: After proposing new solution candidates, the designer should test the candidates against other knowledge sources. This operation follows AOTP shown in Figure 2 (a). Model-based operations for this are the same to those for solution synthesis (k-6). The difference is the reasoning mode in model-based systems, i.e., deductive or abductive. From the logical aspect, the first and second steps are the same to solution synthesis (k-6), and the third step is to deduction of properties of objects (l-1) instead.

6 Model-based abduction

The logical and model-based approaches are mutually integrated by knowledge operations that can associate operations in the both workspaces. This integration enables new types of reasoning smoothly combining different kinds of ontologies that are embedded in various model-based reasoning systems. Here, logical level abduction is enhanced by model-based abduction. Model-based abduction is an inference mode in which models are operated and inferred to incorporate new extra-logical statements into logical reasoning. In other words, various knowledge bases that are based on different ontologies cooperate each other to arrive at richer design solutions than pure logical level abduction. Model-based abduction can provide various methods based on particular models that are used as heuristics in design such as generate and test, catalog retrieval, case-based reasoning, computational model, and optimization techniques instead of pure logical abduction.

The process of model-based abduction is formalized as iterative exploration of candidates by applying different theories (see Figure 5). In order to apply this algorithm, we first need relationships between the ontology of each model-based system and the metamodel ontology to translate representation of a solution in one aspect to another. We also need semantic categorization of facts into entities, functions, and properties in those ontologies to interpret knowledge on models as formula shown in Figure 5.1.a.ii, and distribute representation of a solution into either facts or theorem. We discussed the details of model-based abduction in [4].

7 Conclusions

We discussed the nature of synthesis in design and showed a model that combined both logical and model-based representations. By doing so, we could clarify the over-all structure of the synthesis-oriented process as well as a general computational framework for modeling of synthesis that performs new types of inferences, such as knowledge integration by model-based abduction.

Synthesis is not mere abduction but an appropriate combination of abduction and deduction. Since actual abduction and deduction are deeply dependent on domain knowledge, we should model synthesis with knowledge operations. The proposed computational framework allows integration of logic level reasoning and model-based reasoning that have different knowledge bases with different ontologies.

We conducted case studies about how our framework can be applicable to actual design processes [5][6]. We could explain core parts of the processes including analysis and synthesis, as well as preparatory parts such as knowledge acquisition. Based on this result, we are now building a prototype system and testing it [7][8]. The system is an environment to provide designers with various kinds of design knowledge and abductive capabilities. This research is supported by the Research for the Future Program of the Japan Society for Promotion of Science in the project of "Modeling of Synthesis," JSPS-RFTF96000701

References

[1] Hartshorne, C. and Weiss, P. eds., 1931-1935, "The Collected Papers of Charles Sanders Peirce, " Vol. I-VI, Harvard University Press.

[2] Takeda, H., Veerkamp, P., Tomiyama, T. and Yoshikawa, H. "Modeling design processes", AI Magazine, 1990, Vo.11, No. 4, pp.37-48.

[3] Tomiyama, T., Murakami, T., Washio, T., Kubota, A., Takeda, H., Kiriyama, T., Umeda, Y., and Yoshioka, M., "The Modeling of Synthesis – From the Viewpoint of Design Knowledge," ICED '97, 1997, pp. 97-100.

[4] Yoshioka, M. and Tomiyama, T., "Model-based Abduction for Synthesis", Proc. 2000 ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference, ASME, DETC2000/DTM-14553 (CD-ROM), 2000

[5] Tsumaya, A., Takeda, H. and Tomiyama, T., "A Synthesis-Related Analysis Method of Design Protocol," ICED '99, 1999, Vol. 3, pp. 1949-1952.

[6] Tsumaya, A., Nomaguchi, Y., Yoshioka, M., Takeda, H., Murakami, T. and Tomiyama, T., "Verification of a Model of Synthesis -The methods for verification and results-", Proc. ICED 2001, (to appear)

[7] Yoshioka, M. and Tomiyama, T. "Toward a Reasoning Framework of Design as Synthesis," Proc. 1999 ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, ASME, DETC99/DTM-8743 (CD-ROM), 1999.

[8] Yoshioka, M., Nomaguchi, Y. and Tomiyama, T, "Proposal of an Integrated Design Support Environment Based on the Model of Synthesis", 27th Design Automation Conference in 2001 ASME Design Engineering Technical Conference & Computers and Information in Engineering Conference, 2001 (to appear).

Hideaki Takeda
National Institute of Informatics, 2-1-2 Hitotsubashi Chiyoda-ku Tokyo 101-8430, Japan, Tel: +81-3-4212-2543, Fax: +81-3-3556-1916, E-mail: takeda@nii.ac.jp, URL http://www-kasm.nii.ac.jp

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DESIGN ACTIVITY MODELLING – A PERFORMANCE VIEWPOINT

F J O'Donnell and A H B Duffy

Keywords: Descriptive models of the design process; process modelling; design management; design development; performance.

1 Introduction

Design activity modelling has received significant attention in research over the last 30 years with a focus on both descriptive and prescriptive models. This has resulted in the development of models offering different viewpoints of the design process, such as the description of the process in terms of activities/stages [1], the cognitive nature of design as described by Smithers [2] and those relating design within an overall model of product development [3].

These models focus primarily on the activities required to create a design solution, i.e. design activities, in isolation of the activities involved in managing the process by which that solution is developed, i.e. design management activities, and the relationship between them. This paper presents a novel formalism describing activities focused both on the design and its development, i.e. the *design development* process. The model describes how outputs of design and design management activities are evaluated within a model of performance measurement and management in design development.

2 Current activity models in design development

When we refer to performance in design it can be related to different areas, e.g. the performance of the design solution (artefact) in terms of its properties, such as the top speed of a car, or the performance of the process by which the solution was created in terms of duration or cost. These areas of performance are related in some way but the relationship is unclear.

Existing models of activities in design are almost exclusively focused on the performance of the design (artefact) and not the performance of the activities required for its development. For example:

- Radcliffe [4] highlights the importance that designers place on design management activities within the design process in his protocol analysis but the analysis does not identify how design and design management activities are linked/related. Indeed, throughout the collection of papers from the Delft Workshop, Analysing Design Activity [5], the analysis is restricted to the achievement of design (artefact) goals.

- Smithers [2] presents a model of the design process which treats design as a knowledge based exploration task. The model illustrates the role of knowledge in design. However, all of the activities described in the work are focused on processing knowledge of the design. Design management activities, such as scheduling and control are not included within the work.

- The model of Pahl and Beitz [1] is representative of a number of phase/stage based models and provides a step-by-step method to be followed in design, which supports scheduling of the activity within discrete phases. However, the tasks outlined in this model are focused on design goals and there is no reference to activity goals and the need to manage the design process in relation to both design and activity goals, e.g. the trade-off between cost of design development and quality of the design. Similarly, French [6] suggests that evaluation is carried out continually within the design process but this evaluation is focused on the initial need in terms of the artefact.

- Authors such as Andreasen [3] and Hales [7] provide more insight into the (business) context in which design is carried out. Andreasen identifies the need for greater efficiency in product development while also ensuring better results in terms of the artefacts produced. The concept of efficiency as defined by Andreasen identifies the trade-off between what is being achieved in product development and the costs (and implicitly time) incurred. However, the author provides a viewpoint, identifying the need for managing such a trade-off, and does not relate this within an activity/process model to further illustrate how it might be achieved.

The design activity and process models discussed provide significant insight into the activities, stages, etc. in design. There is a reasonable consensus on the main types of activities involved in design, their sequence, etc., and the evaluation of the output in relation to the design goals is a key component of the models discussed. However, the analysis of performance in relation to the activities carried out in design is restricted to literature addressing the management of design at the project level, e.g. [7]. However, it is proposed here that management activities are carried out at every level in design and not just at a project level and therefore there is a requirement to analyse performance in relation to activities at all levels.

3 E^2 performance model

An activity model is presented here (Figure 1) focusing on knowledge in design. This model is based on IDEF∅ [8], one of the Integrated Computer Aided Manufacturing Definition (IDEF) techniques, which was specifically created to model activities, processes or functions.

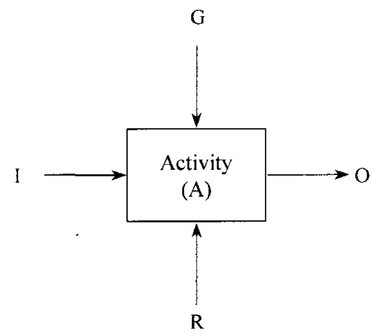


Figure 1. Knowledge Processing Activity

Design may be seen as the processing of knowledge [9], i.e. knowledge is continuously evolved as a result of specific activities between extremes of abstract versus concrete and general versus specific [10-12]. The design activity uses resources to transform input to output under the direction of goals and constraints. Figure 1 illustrates such an activity and the key categories of knowledge that relate to it. All inputs and outputs may be represented as forms of knowledge, e.g. a designer is represented in this model as a knowledge resource *(R)*, the state of the design prior to the activity may be described as the knowledge input *(I)*, etc. Four categories of knowledge are identified here:

- Knowledge Input *(I)*: the knowledge present prior to the activity;
- Knowledge Output *(O)*: the knowledge present as a result of the activity taking place;
- Knowledge Goal *(G)*: the knowledge which directs and constrains the activity;
- Knowledge Resource *(R)*: the knowledge which acts on the input to produce the output.

The category in which an element of knowledge resides is not fixed, but derived from the context of the model, i.e. the activity to which it is related. For example, an output of one activity may act as a constraint on another.

The performance of activities in design is described within the concepts of *efficiency* and *effectiveness* [13] and their relationship is further elaborated in [14]. Efficiency (η) refers to the relationship between the knowledge gained in the activity and the cost (in terms of time, money, etc.) of resources required to achieve that gain. Efficiency describes the inherent behaviour of activities in design but does not directly indicate goal achievement. The degree to which the result (output) meets the goal may be described as the activity effectiveness. Therefore, to obtain a fully informed view of activity performance both efficiency (η) and effectiveness (Π) are evaluated (Figure 2). That is:

$$\text{Design Performance} \\ \equiv \text{Efficiency } (\eta) \text{ and Effectiveness } (\Pi)$$

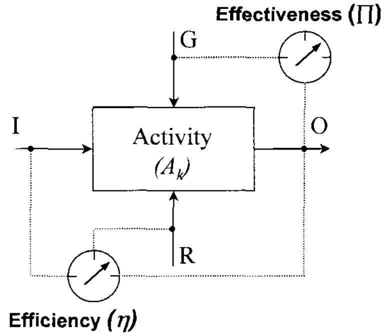


Figure 2. E^2 Performance Model

4 Design and Management

The knowledge goal *(G)* identified in Figure 1 may be related to either the *design (DG)* e.g. reliability, aesthetics, or the *design activity (DAG)* involved in creating that design, e.g. time consumed, labour costs, etc. The design and design activity goals may be managed intuitively by the designer in what has been presented in Figure 1 as one activity. However, it is proposed that there are two types of activity taking place; *design activities* (A_d) and *design management activities* (A_m). Design activities are focused on the design goals *(DG)* while design management activities are concerned with design activity goals *(DAG)* and managing the trade-off between achieving design and design activity goals to ensure best overall performance.

At a design project level these activities are often defined separately and are generally carried out by different people e.g. the designer/design team and the design manager [7]. However, the distinction between these activity types exists even at the level of individual design activities. For example, during sketching a designer may glance at their watch to evaluate the time elapsed in relation to an implicit or explicit time goal before proceeding. This represents a change of activity, i.e. from a design activity, focused on producing a sketch in accordance with a design goal *(DG)*, to a design management activity focused on ensuring a design activity goal *(DAG)* is achieved, e.g. sketch is completed on time.

Given the basic design activity representation presented in Figure 1 and the distinction between design and design management presented above, a further model is introduced in Figure 3 to describe design and its management. This Design Activity Management (DAM) model represents a *managed activity* i.e. any activity in design aimed at achieving design and design activity goals. The categories of input *(I)*, output *(O)*, goal *(G)* and resource *(R)* knowledge, presented in Figure 1, are decomposed to reflect categories related to either design or design management activities as follows:

$$I \rightarrow DI \text{ and } DAI$$
$$O \rightarrow DO \text{ and } DAO$$
$$G \rightarrow DG \text{ and } DAG$$
$$R \rightarrow DR \text{ and } DAR$$

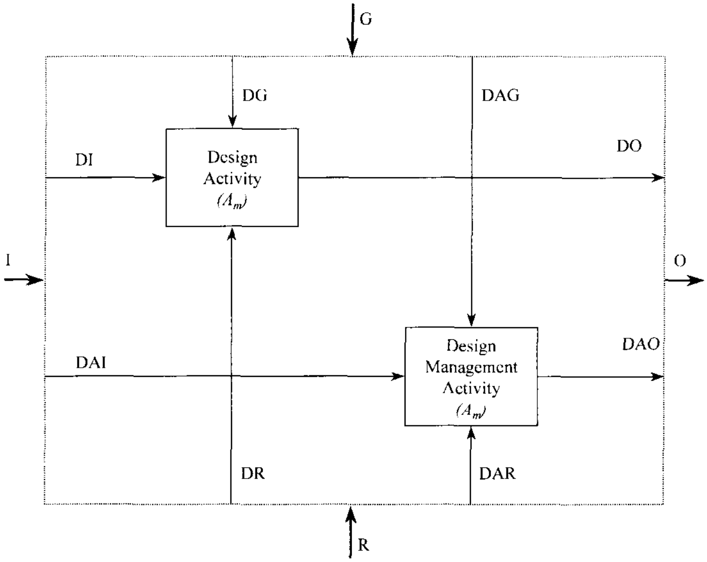


Figure 3. Design Activity Management (DAM) model

The managed activities described above are the fundamental elements of the design process, i.e. the design process consists of a number of managed activities with relationships such as those based on information dependencies and described as dependent, independent or interdependent [15]. Having established the design and design activity goals, perhaps through planning activities, the focus subsequently moves to ensuring these goals are achieved [11], i.e. optimising overall effectiveness. This overall effectiveness is composed of *design effectiveness,* illustrating how well the design goals have been met, and *design management effectiveness,* indicating if the design activity goals, such as resource cost, have been met.

In an informal sense, a designer will continually evaluate the effectiveness of his/her activities, e.g. checking their watch to assess time elapsed (design management effectiveness), evaluating the aesthetic strengths of a particular concept (design effectiveness), etc. More formally, effectiveness may be reviewed through simulating product behaviour and evaluating results at specific stages as represented within many of the phase models of the design process.

4.1 A Model of Performance Measurement and Management (PerMM)

The measurement of design and design management effectiveness is presented here as a critical part of controlling a managed activity within a process model for Performance Measurement and Management (PerMM) in design development. The description below focuses on a typical sequence of events in evolving the state of the design highlighting the main decision points.

1. The design activity (Ad) takes DI as input and, directed by knowledge of the specific design goal (DG), produces an output (DO) aimed at meeting the goal. This output will be compared against the goal to determine the level of design effectiveness, $\Pi(Ad)$, achieved in the activity (Figure 4).

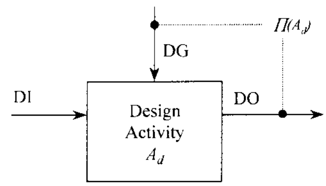


Figure 4. Design Effectiveness

2. The resulting level of design effectiveness $\Pi(A_d)$ is used as an input of control knowledge into the design management activity (Figure 5). The description of design effectiveness may describe how well a design goal has been met or whether a constraint has been satisfied or not.

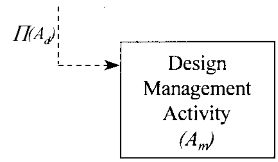


Figure 5. Effectiveness Input

3. The design management activity analyses design management effectiveness, $\Pi(A_m)$, using knowledge (including meta-knowledge) of the resources being used in both the design and design management activities. This knowledge is primarily time and cost based i.e. it refers to the time consumed or cost incurred during a particular activity-resource relationship. This is compared against knowledge of the design activity goal *(DAG)*, e.g. *to achieve a design lead time of 1 month*, to determine the level of design management effectiveness (Figure 6).

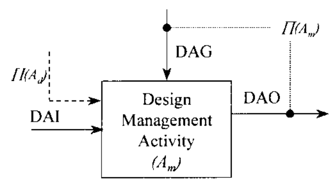


Figure 6. Design Management Effectiveness

4. Utilising design activity resource (DAR) knowledge the design management activity evaluates the relationship between design and design management effectiveness and decides on the controlling action, if any[1], which must be taken as an attempt to optimise overall effectiveness. This controlling action will typically involve changing the goals or resources in order to achieve a change in effectiveness.

Figure 7 is evolved from Figure 3 to illustrate the decision points and flow of control knowledge (shown as dashed lines) within a managed activity and serves to summarise the steps described above. That is, the model describes the process of measuring and managing performance in relation to both design and design activity goals. The following outlines the types of controlling action, aimed at optimising overall effectiveness, that may result from the evaluation of design and design management effectiveness:

[1] It may be desirable to take no controlling action, i.e. to maintain all goals, resources, etc. as they currently are and allow the managed activity to continue.

- At decision point c_i the decision options are to terminate the activity having established satisfactory levels of design and design management effectiveness *or* to continue with the activity.

- At decision point c_j the decision options are to redefine goals *and/or* alter resource allocation.

- At decision point c_k the decision options are to redefine design goals *(DG) and/or* design activity goals *(DAG)*. For example, the outcome of the design management activity may be to set a new launch date for the project. In contrast, it may be more appropriate to reduce the targets specified in some design goals, e.g. life in service, while maintaining the original planned launch date.

- At decision point c_l the decision options are to alter design resources *(DR) and/or* the design activity resources *(DAR)*. For example, the outcome from the management activity may be to allocate additional design resources to achieve increased design effectiveness with a probable negative impact on design management effectiveness.

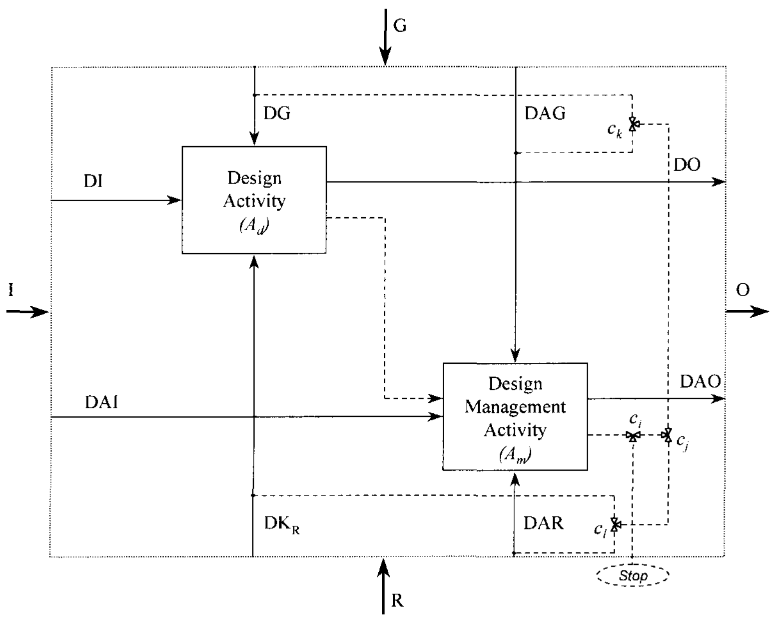


Figure 7. Performance Measurement and Management (PerMM) Process Model

5 Discussion and conclusion

Existing activity models of design fail to comprehensively capture the performance of both the design and the activities involved in its development. A design and activity management

(DAM) model has been presented as a novel means to illustrate the relations between design and its management. The E^2 model has been presented as a formalism of performance and provides a basis for describing the measurement and management of performance in design development. Using E^2 and DAM as a basis, a model of Performance Measurement and Management (PerMM) clearly distinguishes, yet relates, performance of the artefact and of the activities. The models presented here were evaluated using data from the Deft protocol studies [16] and through expert appraisal with industrial experts in areas such as performance management and design development. These are more fully reported in [17].

References

1. Pahl, G. and W. Beitz, *Engineering Design: A systematic approach.* 2nd Revised Edition ed, ed. K. Wallace. 1996, London: Springer-Verlag.
2. Smithers, T., *et al.*, *Design as Intelligent Behaviour: An AI in Design Research Programme.* Artificial Intelligence in Engineering, 1990. **5**(2).
3. Andreasen, M.M. and L. Hein, *Integrated Product Development,*. 1987: IFS (Publications) Ltd and Springer-Verlag London UK.
4. Radcliffe, D.F., *Concurrency of Actions, Ideas and Knowledge within a Design Team,* in *Analysing Design Activity*, N. Cross, H. Christiaans, and K. Dorst, Editors. 1996, John Wiley and Sons.
5. Cross, N., H. Christiaans, and K. Dorst, eds. *Analysing Design Activity.* . 1996, John Wiley and Sons: Chichester, West Sussex. 463.
6. French, M.J., *Conceptual Design for Engineers.* 3rd ed. 1998, London: Springer.
7. Hales, C., *Managing Engineering Design.* 1993: Longman Scientific and Technical.
8. Colquhoun, G.J., R.W. Baines, and R. Crossley, *A state of the art review of IDEF0.* International Journal of Computer Integrated Manufacturing, 1993. **6**(4): p. 252-264.
9. Persidis, A. and A. Duffy. *Learning in engineering design,*. in *Selected and Reviewed Papers and Reports from the IFIP TC/WG5.2 Third Intelligent Workshop on Computer Aided Design Osaka Japan September 1989.* 1991. Elsevier Science Publishers B.V. (North-Holland) Amsterdam The Netherlands.
10. Andreasen, M.M., *Modelling - The Language of the Designer.* Journal of Engineering Design, 1994. **5**(2): p. 103-115.
11. Ullman, D.G., T.G. Dietterich, and S.L. A, *A Model of the Mechanical Design Process Based on Empirical Data.* AI EDAM, 1988. **2**(1): p. 33-52.
12. Zhang, Y., *Computer-based Modelling and Management for Current Working Knowledge Evolution Support*, in *Department of Design, Manufacture and Engineering Management.* 1998, University of Strathclyde: Glasgow.
13. Duffy, A.H.B. and F.J.O. Donnell. *A Model of Product Development Performance.* in *Designers: Key to Successful Product Development.* 1997: Springer.
14. O'Donnell, F.J. and A.H.B. Duffy. *Performance Management at Design Activity Level.* in *International Conference on Engineering Design (ICED 01).* 2001. Glasgow.
15. Eppinger, S.D., *et al.*, *A model-based method for organising tasks in product development,*. Research in Engineering Design, 1994. **6**(1): p. 1-13.
16. Cross, N., H. Christiaans, and K. Dorst, eds. *Analysing Design Activity.* 1996, John Wiley and Sons: Chichester, West Sussex. 463.
17. O' Donnell, F. J., *A Methodology for Performance Modelling and Analysis in Design Development.* PhD Thesis, CAD Centre, University of Strathclyde, Glasgow. December 2000.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

APPLICATION OF DISSIPATIVE STRUCTURES TO IMPROVE THE GENERATION AND SELECTION OF NEW PRODUCT IDEAS

C Marxt

Keywords: Complexity management, dissipative structures, idea generation and selection

1 Introduction

In a time where the customer's voice is the most important in many companies and fulfilling the customer's needs is a condition precedent, product innovation should be one of the main foci of the general management. Especially the early stages of the new product development process are crucial - idea generation, preliminary investigation and building of a business case [1] [2] [3].

On the one hand, many companies rely very much on the ideas from their development department and on the other hand they use surveys to anticipate what the market will accept open handed. But this process of generating a lot of ideas and then selecting the "right one" has its flaws. Depending on the study only 0.1-14 out of 100 new product ideas have a chance to result in a new product in the market [1] [4] [5]. The selection of this one product is the task of the management team either on the department or on a company wide level. But according to Deschamps and Nayak approx. 54% of senior managers think that there is a lack of a formal process for idea generation and evaluation, which results in a lack of really innovative new concepts [6, p. 200]. Usually the management or decision team strives for meeting internal restrictions as well as the market needs. Doing this can result in a typical equilibrium decision, and the outcome is often not as radical as expected.

Therefore, this paper aims at

- describing the idea generation and selection process and showing its relevance for the entire new product development (NPD) process,
- showing why the usually used selection criteria don't work as expected
- comparing the concept of dissipative structures to the idea generation and selection process
- and finally giving first ideas how to implement the concept in day to day work.

In a first step the paper describes the idea generation processes with its stages and decisions. It ends with showing some of the flaws of today's idea selection process. The next chapter describes the original concept of dissipative structures by Ilya Prigogine [7] and its preliminary applications in management. The next and main part of this contribution lies on first, basic reflections on how to adopt the concept of dissipative structures for the idea generation and selection process. Finally, some implications for the practitioner conclude the paper.

2 The idea generation and selection process

As idea generation, capture and selection have been recognised as an important task for any company, these activities are described in detail by many authors [1] [8] [2]. Figure 1 shows a highly aggregated model of the idea generation and selection process. Usually, this process consists of four phases and as many decisions which have to be taken. In stage one the idea itself is generated in an individual's mind. The individual itself always decides on whether it will communicate the idea to someone else or not (D1), even if the generation takes place in a team.

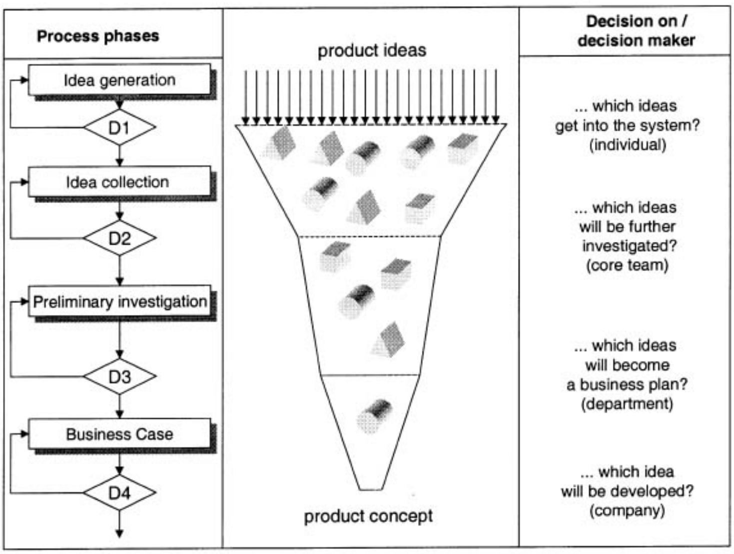


Figure 1: Idea generation and selection process

The next step is the collection and structuring of the ideas. This usually takes place in a team of experts, which know the relevant areas and are able to prepare the next decision. This core team will judge on the ideas and select several for a closer investigation (D2). There is a basic distinction found in literature between an incremental and a radical idea [9] [10]. The third step is a preliminary investigation on whether the concepts fit to the strategic goals, have enough market potential, and are technically feasible. The decision on that is taken in D3 by the department heads. Finally, a business case is prepared and goes to the board of executives who will decide on necessary investments (D4).

When selecting new product ideas a company is usually trying to answer the following questions [11] [1] [12]:

- Does the product fulfil the customer's needs?
- Is the feasibility and potential for realisation realistic?
- Does the idea fit to strategic goals?
- Can the necessary market volume be reached
- Is the market and technology risk bearable?

- Are the costs, the quality and the timing right for the target market?
- Is there an affinity to the rest of the product range?
- What is the competitor doing?

The effect of choosing ideas according to this set of questions is that the whole system (e.g. company) is kept in a fine tuned balance so that no big changes in the department or company will occur. But is this the right way to find and realise radically new ideas? The next chapter introduces the concept of dissipative structures and its use in social sciences.

3 The concept of dissipative structures

3.1 Dissipative structures in thermodynamics

Up to the end of the 19^{th} century or even the middle of the last century the world was seen as static and the equilibrium, although sometimes disturbed, was the main goal a system would try to reach. But this perception changed dramatically. Among others, Prigogine regarded the world as a dynamic system where the classical laws of a Newtonian era only apply in a minority of all situations [13]. Change and transformation of a system only occur in non-equilibrium states of the system. These conditions far from equilibrium are subject to different laws. The system is regarded as a highly complex network of non-linear relationships between its elements. Therefore the behaviour of the system is not determinable by knowing the input parameters. If the system moves very far from the equilibrium it might occur that the structure of the system breaks down and the whole system becomes chaotic. In this situation simple rules applied to the system might result in new, totally different structures. At the point where the system breaks down it opens up its boundaries and starts importing energy and exporting entropy. This kind of system is called a “dissipative system”.

3.2 Dissipative structures in social sciences

The idea of applying the concepts of non-linear dynamic systems to the social sciences is nothing new. Several authors have described how non-linear concepts could be applied to social systems. This resulted in management concepts and ideas such as the concept of technological discontinuities by Tushman and Anderson [14], the fractal company by Warnecke [15], the punctuated equilibrium concept of organisational transformation by Romanelli and Tushman [16], or in the use of mathematical concepts of chaos theory in strategic planning [17].

The concept of dissipative structures is rarely used and if, it describes either rather sociological aspects of transformation [18] or organisational change on a very high level of aggregation [19]. MacIntosh and MacLean for example use the concept to explain certain effects in the management of organisational transformation. They point out that only the handling of deep structures (e.g. organisational culture) can give the management some influence over the chaotic phase of the system, when it moves away from the equilibrium [20].

Developing new products is also some kind of transformation. Knowledge and experience combined with other information is transformed into business or product ideas which are brought to the market. The next part shows, how the concept of dissipative structures can be applied to the process of idea creation.

4 Dissipative structures in idea generation and selection

4.1 Rigid structures and open systems in the product development process

The community innovation survey of the European Union [21] has looked into approx. 90.000 companies. The idea was to give a state of the art description of the innovation system in European companies One of the areas of the survey covered the questions how many companies had problems with their product innovation and why.

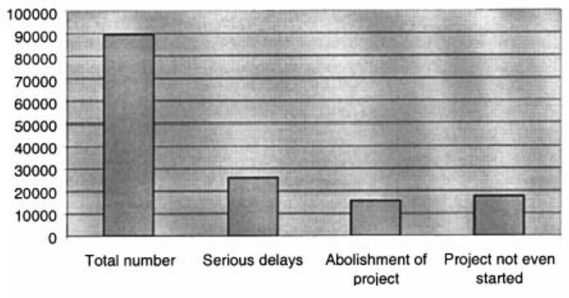


Figure 2: Problems with innovation projects in comparison to total numbers of projects

Analysing the reasons for this problems the survey has shown that besides lack of financial resources, excessive project risks, lack of customer responsiveness to products, lack of qualified personnel or state of the art technology especially organisational rigidities have led to problems in projects. Depending on the type of delays the following chart (figure 3) shows how many projects have been delayed due to organisational rigidities.

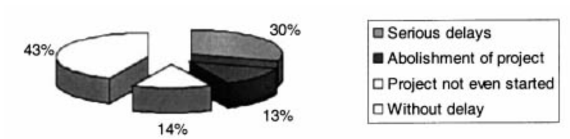


Figure 3: Number of innovation projects with problems due to organisational rigidities in comparison to total numbers of projects delayed (in %)

This means that approx. 30% of all projects which are seriously delayed are delayed due to rigid organisational boundaries. These restrictions occur in every phase of the new product development process. Approximately 50% of the companies rely very much on information from outside their own companies during their innovation process. One possible interpretation can be that the companies have to open their system boundaries to get to the necessary information. Therefore a new structural approach, such as dissipative structures, might be used to explain the surveyed results. A first step is to identify equilibria in the development process.

4.2 Dissipative structures and decision making

Many companies want the idea generation process to run smoothly. Running smooth means nothing else than as close to the equilibrium as possible, because most of the members of the organisation know what kind of ideas are expected (due to their internal socialisation). Even if the process itself generates radical new ideas the decision structures (see figure 1) usually filter radical ideas out of the system, because any disturbance in the system results in complex additional or even unwanted work. This means applying dissipative structures to idea generation means to apply it to decision making in the idea generation and selection process.

The following figure shows which type of idea – radical or incremental – is chosen depending on the ability to implement these ideas. An individual such as an inventor or an engineer in a development department is often the nucleus of radical ideas, because he usually has no constraints but his own. In many cases he on the other hand is not able to implement his ideas, due to a lack of resources. A team in a company can push ideas with a much greater force but the team has already some constraints either by internal agreement or by restrictions from the management. The next decisions are made on departmental level. Usually the ideas chosen are less radical because the system as a whole is seen as more static and therefore less able to cope with radical ideas. Anyway on this level the ability to implement is rather high. On an organisation wide level ideas can be realised, but due to constraints from many individuals and subsystems the type of idea is usually incremental. The shaded area in figure 2 shows the optimum and the small arrows indicate the direction where improvement can take place. D1, D2, D3 refer to the decision points in figure 1.

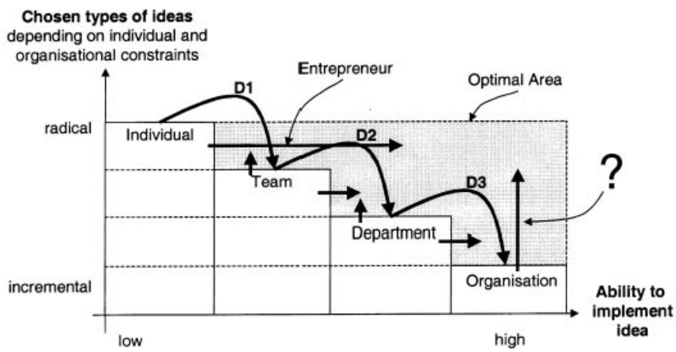


Figure 4: Idea selection in function of the ability to implement

The best available solution would be to have radical ideas and to be able to implement them. Looking at an Entrepreneur is a good way to illustrate the idea of the picture. Not only are his ideas of a radical nature but he also has an ability to implement an idea. He therefore comes close to the optimum. But many companies rely on the competencies and management present and not all of the managers have the abilities of great entrepreneurs. Hence the question is, how to enable an organisation to generate and especially select radical ideas?

4.3 Implications for the practitioner

From an organisational point of view there are four levels where non-equilibrium states can be induced:

- individual level, e.g. further education, job rotation, sabbaticals, etc.
- team level, e.g. diversity of team members, changes in team formation, new processes, etc.
- department level, e.g. restructuring, new distribution of resources (money, manpower), strategic choices, etc.
- organisational level, e.g. cooperative new product development, strategic alliances, mergers and acquisition, etc.

To generate radically new ideas new ways of work are necessary. Cooper for example describes ways to generate great new product ideas, such as running "pizza-video parties", allowing time off for technical people, making customer brainstorming sessions part of the plant tours, running group creativity sessions, setting up liaisons with key researchers, etc [1, p. 111]. All these things aim at inducing instability and by doing so opening the system boundaries and get more "energy" into the system, just as suggested by the concept of dissipative structures.

If a company lacks radical ideas on an organisation wide level, the induction of instability is very important. One of the ways to do that is to reorganise the whole company or else for example to start a cooperation with partners in new product development. The following chart shows areas of instability as they can occur in cooperative new product development. The results are drawn from a survey in the Swiss Industry [22].

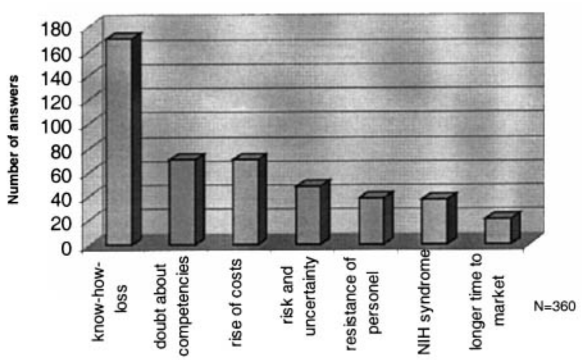


Figure 5: Selected areas of instability in collaborative new product development as expected by the companies

It is important that this partnership is not only based on loose contacts but that it is a very intense cooperation, e.g. by exchanging collaborators. Only this high degree of instability leads to radical ideas.

5 Conclusions

The paper tried to show initial ideas on how the concept of dissipative structures could be used to understand certain effects concerning the type of ideas generated and selected in specific organisational settings. Idea generation and selection is a stochastic process in a complex system and therefore underlies the same effects as systems in thermodynamics. The more instability is induced and the further the system moves from equilibrium the more chaotic the system gets. But only out of this kind of chaos new radical ideas will evolve. One way to generate a high amount of instability in early stages of the new product development process is to make extensive use of cooperative arrangements. This will disturb the responsible departments or even the whole organisation in such a great extent that the possibility of the evolvement of radical ideas rises.

Therefore, if a manager wants to improve radically on the products in his or her company he should generate as much instability as possible and furthermore select only ideas which are inducing as much instability as possible. If incremental improvements are requested, minor disturbances in the idea generation and selection process are sufficient. Currently we are working on project to proof the implications of this paper

6 References

[1] R. G. Cooper, *Product leadership*. Cambridge: Perseus Books, 1999.

[2] M. D. Rosenau, "The PDMA Handbook of New Product Development," . New York: John Wiley & Sons, Inc., 1996.

[3] A. Khurana and S. R. Rosenthal, "Integrating the Fuzzy Front End of New Product Development," in *Sloan Management Review*, vol. 38. Cambridge, Massachusetts: Sloan Management Review Association, 1997, pp. 103-120.

[4] G. Specht and C. Beckmann, *F&E Management*. Stuttgart: Schäffer-Poeschel, 1996.

[5] T. P. Hustad, "Reviewing current practices in innovation management and a summary of selected best practices," in *PDMA Hanbook of New Product Development*, M. D. Rosenau, Ed. New York: John Wiley & Sons, 1996, pp. 489-511.

[6] J.-P. Deschamps and R. Nayak, *Product juggernauts: how companies mobilize to generate a stream of market winners*. Boston: Harvard Busniess School Press, 1995.

[7] I. Prigogine, *Vom Sein zum Werden : Zeit und Komplexität in den Naturwissenschaften*. München, Zürich: Piper, 1985.

[8] R. Boutellier and R. Völker, *Erfolg durch innovative Produkte: Bausteine des Innovationsmanagements*. München, Wien: Hanser, 1997.

[9] R. Henderson and K. Clark, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and Failure of Established Firms," *Administrative Science Quarterly*, vol. 35, pp. 9-30, 1990.

[10] J. Hauschildt, *Innovationsmanagement*, 2. Auflage ed. München: Vahlen, 1997.

[11] M. Mussard and H. Wyss, *Innovationsleitfaden*. Zürich: SIA, Schweizerischer Ingenieru- und Architektenverein, 1993.

[12] C. W. Kim and R. Mauborgne, "Knowing a Winning Business Idea When you see One," *Harvard Business Review*, pp. 129-138, 2000.

[13] I. Prigogine, *Die Gesetze des Chaos*. Frankfurt/New York: Campus, 1995.

[14] M. L. Tushman and P. Anderson, "Technological Discontinuities and Organizational Environments," *Administrative Science Quarterly*, vol. 31, pp. 439-465, 1986.

[15] H.-J. Warnecke and M. Hüser, *The fractal company : a revolution in corporate culture*. Berlin: Springer, 1993.
[16] E. Romanelli and M. L. Tushman, "Organizational Transformation as Punctuated Equilibrium: an Empirical Test," *Academy of Management Journal*, vol. 37, pp. 1141-1166, 1994.
[17] D. Levy, "Chaos theory and strategy: theory, application, and managerial implications," *Strategic Management Journal*, vol. 15, pp. 167-178, 1994.
[18] C. Smith and G. Gemmill, "Change in the Small Group: A Dissipative Structure Perspective," *Human Relations*, vol. 44, pp. 697-716, 1991.
[19] R. Leiffer, "Understanding Organizational Transformation Using a Dissipative Structure Model," *Human Relations*, vol. 42, pp. 899-916, 1989.
[20] R. MacIntosh and D. MacLean, "Conditioned emergence: a dissipative structures approach to transformation," *Strategic Management Journal*, vol. 20, pp. 297-316, 1999.
[21] EUSTAT, "Community innovation survey database," 1996.
[22] C. Marxt, A. Staufer, and A. Bichsel, "Innovationskooperationen," *io management*, vol. 5, pp. 55-59, 1998.

7 Authors Title and Name

Dr. sc. techn. Christian Marxt
Swiss Federal Institute of Technology Zurich (ETHZ)
Department of Industrial Engineering and Management
Zurichbergstrasse 18
8028 Zurich
Switzerland
Phone: +41 1 632 05 56
Fax: +41 1 632 10 45
E-mail: christian.marxt@ethz.ch

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

DYNAMIC CONCEPT DEVELOPMENT - A KEY FOR FUTURE PROFITABLE INNOVATIONS AND NEW PRODUCT VARIANTS

S Ottosson

Keywords: Creativity, Invention, Innovation, Product Development, Product Variant, Project

1. Introduction

For companies (small and large) that want to maintain or establish market leading positions it is not enough to perform computer-based re-engineering of existing solutions - which many managers seem to think. The companies must also seek, create, develop and market new solutions i.e. innovations and new product variants. However the creation and early development of innovative new solutions is not much described in scientific papers. One reason for that is that classical research is difficult to perform when stability and predictability no longer are obvious, which is the situation in the early phases of new product development. Another is that it is difficult to objectively measure efficiency between different ways of performing product development. To find ways to overcome that problem, our research and empirical work on product development is mainly based on "participation action research" [1], which has its ground in quantum physics and complexity theories (see table 1).

The Quantum Paradigm	**The Classical Paradigm**
Statistic (probabilities)	Deterministic (controllable)
Holistic	Atomistic
Small effects important	Small effects neglible
Dis-continous	Continous
Focus on relations	Focus on objects
Many equally good solutions	One best solution
Chaos is the ground for development	Chaos is destructive
Both/and	Either/or
The observer always influences	The observer should not influence

Table 1 Some characteristics for the Quantum and the Classical (Newtonian) Paradigm

Based on our research we have seen that it is possible to cut down the early product development time typically in the region 25 – 50 % when what we call *Dynamic Product Development (DPD)* is used compared to when *Integrated Product Development (IPD)* is used. The advantage is gained in the early phases i.e. during concept development and the first product development of the concept. That is the therefore the scope of this paper. From the treatment some of the principal differences between DPD and IPD will hopefully be clear.

2. Definitions

By *new solutions* we mean not commonly known inventions, research breakthroughs, and revolutionary new ideas. 'Not commonly known' is a diffuse expression that we need to use as something that not even experts should expect when the solution is/was created. If the solution is not adopted for product development soon after it has been public it loses its newness and becomes a commonly known solution. If the solution is kept secret and not more than, say, ten people know about it and have promised not to tell anyone, the time factor is not limiting for the newness but can the solution be overrun by other solutions.

In order to be an *invention* the solution must be patentable even if a patent application is not filed. Therefore a technical height is needed in order for a technical solution to become an invention. If the invention leads to big steps in technical knowledge it is often said to be a *radical invention.* If the new invention is further developed from existing solutions it is often said to be an *incremental invention.* Inventions can be both product inventions and process inventions and also combinations of product and process solutions.

There are different definitions as regards the term *innovation.* By *technical innovation* we here mean an invention, research findings or a new idea that has been developed into a new product that has been sold to the market and has come into use. By a *new product* we mean a product that is new both from a technical point of view and from a market point of view. Technically, as a rule of thumb, the product should have at least 60 % new or redesigned parts and systems. From a marketing point of view, the product should generally be regarded as new by the "market".

By the term new *product variant* we here mean a new product based on an existing product "platform". Thus a new product that does not live up to the demands to be called an innovation in general is a new product variant.

The product development process from when a *product concept* exists, is today often performed as a project with a project leader and team members representing different specialities.

When a multidisciplinary project team consists of only product development people and process development people, the terms *concurrent engineering (CE)* or *simultaneous engineering (SE)* are often used. When marketing people and other experts also take part in the project, the terms IPD and DPD are used.

IPD was mainly developed for re-engineering situations while DPD was mainly developed for innovative development for which uncertainty and unlinearity are important factors to cope with. DPD builds further on IPD but has the Quantum Paradigm as the main mental model while IPD has the Classical Paradigm as the main mental model (see table 1).

3. Where have the Studies been made?

With the aim of finding usable rules of thumb for product development, different aspects, tools, and guidelines of development have since the early 1990s been tested in as well industrial projects in Swedish companies as in university student projects. This paper is based on experiences gained from the work initially with the aim to speed up IPD and later to further develop DPD.

Our industrial tests have been performed in small and medium-sized enterprises (SME) as well as in large companies. Some examples are: Alert Invest AB (hospital equipment), Careva Systems AB (handicap aids), Daros AB (diesel engine piston rings), Edu-Euro AB (IT-products), Flygt AB (pumps), Frontec AB (mechanical products and machines), Handiquip AB (lifts), IUC West AB (region development), Medical Robotics AB (medical equipment), Prosolvia AB (VR-technology), SKF AB (transport equipment etc.), Swedelift AB (lifts & elevators), Tervix AB (business development), and Texsun AB (solar power). The development projects have resulted in many profitable innovations and new product variants. Also many patents have been granted - however not always leading to new innovations and new commercial product variants.

Most student projects have been performed at the Innovation Engineering education at Halmstad University (www.hh.se). On average about 70 half or full academic year industrial projects with 2–5 students per project were performed every year in 1994–1998. In 1998-1999 five half-year projects were performed at the Innovation Management education at IUC West

AB. In 2000, five student teams performed product development projects during three months at the Electronics Design education at Linköping University (www.liu.se). A number of patents and invention prizes have been the result of these student projects. In some cases innovations have also become a reality.

4. Creativity as an Economical Factor

It is a well-known fact [e.g. 2] that a large extent of the properties of the products are set already during the initial creative development steps of new products and product variants. It is also a well-known fact that the degree of control of the products decreases rapidly during the first steps of development. Therefore a large portion of the product costs in general has been committed only after a small portion of the development resources has been expended in this phase. Estimations [3] tell e.g. that 10-20 percent of the total time spent in the conceptual design phase determines 80-90 percent of the total product cost (see upper part of figure 1). As the development process moves through the planning, design and manufacturing phases, the product becomes increasingly more concrete. For economic as well as practical and mental reasons the management and development teams are thus later less able to change the properties of the product & production set early in the process.

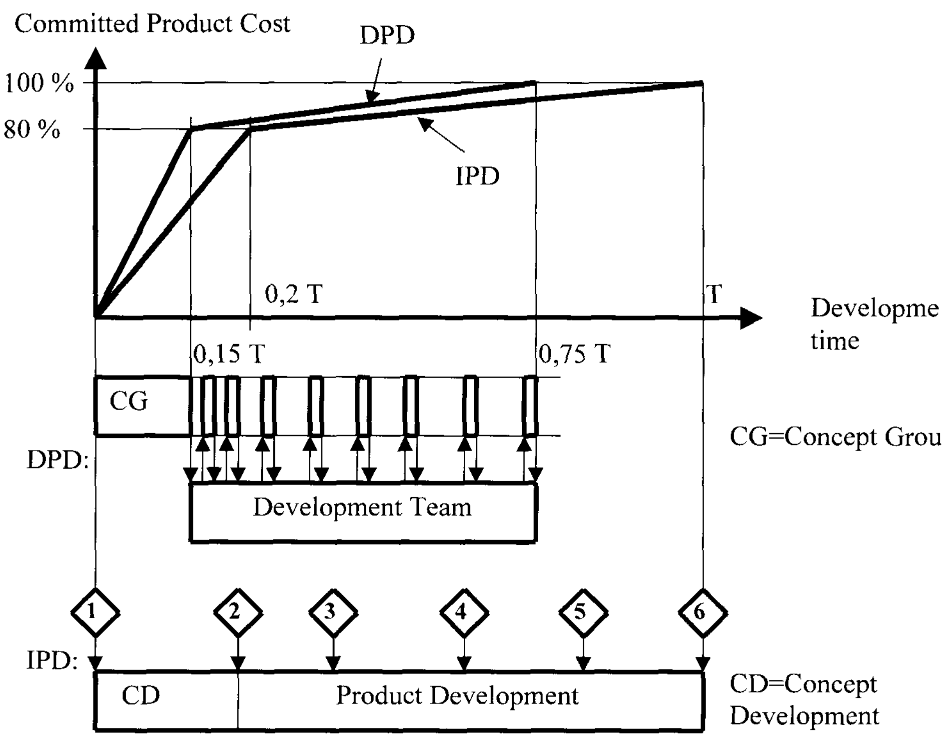


Figure 1: The product cost is largely committed early in the development process. For DPD the time to complete the product development is in general shorter than for IPD

As conceptual development in general has the single largest influence on the product cost and as it sets the premises for the other development phases conceptual development is sometimes regarded to be the most important design phase [e.g. 3]. The conceptual development performance is also extremely important if the product should have the possibility to come early into the market as a quality product or not. Coming early into the market with a new product means that the company will gain a pricing advantage and get ahead of competition on the learning curve. The sales life of the product is also extended which means improved profitability during the product life cycle. Through well performed conceptual development it is thus possible to achieve low development time, low costs, and good performance at the same time as creativity is maximised for the new product.

5. Concept Development

The concept development is in general in IPD only seen as the first phase in the product development process [e.g. 4]. The total development process is controlled by a *steering group* – which acts as an "outsider group" of the project. Controls are performed at well defined decision points (s.c. gates) where decisions are taken if the development shall stop or continue in any form. At the gats the project leader in general presents the situation and actions taken.

In DPD the concept is developed all the time the product development takes place until the development process is completed. This concept work is done by the *Concept Group*, which is an "insider steering group" of the project. One important team member in the Concept Group is the project leader. The Concept Group meets when there is a need for a meeting, which normally means frequent meetings early in the product development process and fewer meetings later in the process as figure 1 shows. The Concept Group normally is a small group (not more than six individuals), which consults resource people when that is needed from time to time. Initially e.g. the Concept Group is often expanded with some creative people to speed up and improve the initial creative work.

The concept development process starts in DPD with a *wish* and not only a *need*, which is the starting point in IPD. *A wish* is a wanted situation or solution in the future. The wish can be a need, a problem or simply a dream to be fulfilled. The wish has to be clarified and visualised and also complemented with other knowledge, e.g. from experts and from the market before the real creative process can start.

The creative process takes place individually or in a group and should hopefully result in at least one unique (radical) solution. We have found that dialogues are more useful than *brainstorming* when technical solutions are to be created. In contradiction to common opinions, we have also found that it is not beneficial to start looking at other solutions as they will block the individual creativity for some time. By looking at existing solutions one is also easily forced to improve details and to not think in product function and abstract terms.

In our work we have found that visualising in the form of making simple sketches is extremely important for fast and creative results. During group work it is beneficial if at least one participant is able to make simple sketches, preferably on a whiteboard, when more than four individuals co-operate simultaneously. Without sketches not everybody has the same vision for the work, which leads to misunderstandings and less efficient work.

For complicated situations it is often beneficial if the problem is divided into more parts to be solved and then be put together and further worked on to be a functional product concept. Time, space and structure are feasible grounds for such a division [5].

Due to our findings, first when a unique solution has appeared, it should be compared with other solutions (so-called *benchmarking*). Depending on the result of that activity, the initial solution should be modified and combined with other solutions in order to get an even better solution to work further from.

Sometimes two or more good solutions occur and it may be difficult to judge which has the best possibilities for the future. In such cases it can be advisable to develop them all simultaneously and to make a judgement later on. If good enough solutions do not emerge, the creative process must be started over again with other creative people participating.

To compare the concept development processes for IPD and DPD figure 2 is shown. If unique solutions are wished or needed, time and uniqueness is gained for DPD mainly as benchmarking and modification is not done before the creative process starts as the own creativity thus gets blocked for some time.

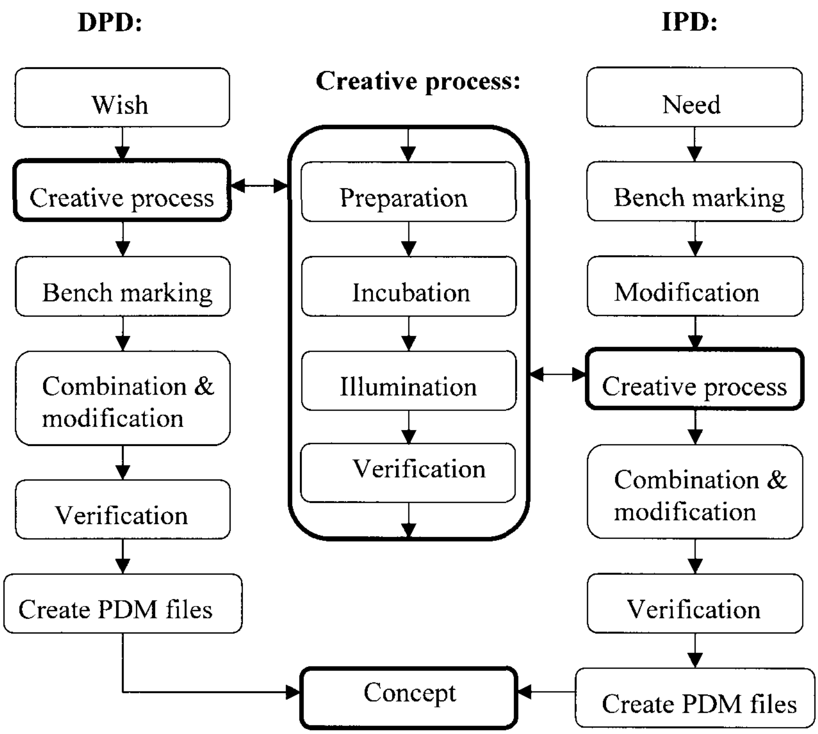


Figure 2: Concept development in DPD and IPD

The creative process follows the same principles for IPD and DPD. Thus when someone has shown an interest and willingness to engage in a wish or need, the chain preparation → incubation → illumination → verification [6] is performed. *Preparation* means active thinking (BAD – Brain Aided Design) combined with sketching (PAD – Pencil Aided Design) or modelling (MAD - Model Aided Design) [7, 8]. *Incubation* is related to the sub-conscious mind that works with a problem until useful solutions show up (the illumination). The time for an *illumination* to appear will differ depending on personal factors and external factors as well as on the problem complexity. Refreshment is needed when illumination does not appear. *Verification* means to check that the solutions meet the wishes (verification is further treated below).

It should be noted that in the creative process people often act on the basis of recognizing well known "patterns" and "characteristics". This method of working has developed simply because it is practical and effective. When we find ourselves in a new situation our brains look for patterns that fit the situation and we get guidance on how we should think and act. Much of this pattern-recognition takes place on an unconscious level. Sometimes the case is that we do not consciously understand a pattern other than through a feeling – which is why it is important to make models in the verification process. The more experience and the more solutions we have stored in our memory, the more solutions will we produce in general, if we have an interest and a willingness to do so [9].

The last step in the first concept development work is to document relevant data manually and electronically in EDM/PDM files (EDM=Engineering Data Management, PDM=Product Data Management). Note that computer aided tools up to this point are of limited use as CAD systems are based on geometric manipulation and reasoning and cannot deal with non-geometric information. Much of the design intent therefore has to be documented separately. It is also difficult to link the geometric solid models with the conceptual models as they are based on different representation schemes.

6. Early Product Development

The chain to perform development from a product concept until a wanted or wished result is reached can be performed as shown in figure 3. If the process any time meets with unsolvable problems occur , a new creative process (see figure 2) may be required to find a new solution to move further from.

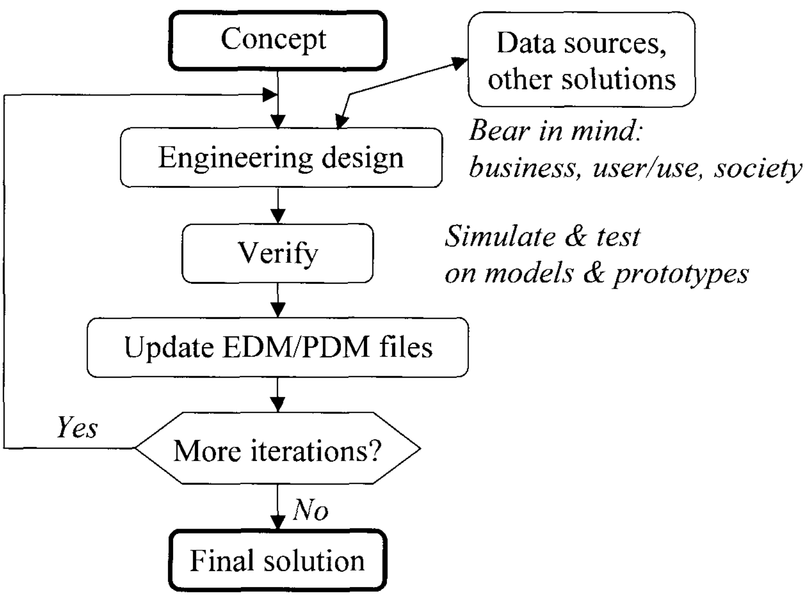


Figure 3: The engineering process from concept to final new product/product variant

If no model has been made in the concept work it is to be recommended that the engineering development start with making one or more models. After that has been done and from this point most work is in general computer based (CAE=Computer Aided Engineering).

The engineering design is very much a question of modifying and combining solutions. Models and prototypes can not be done too often until a final solution has been reality. For each iteration the EDM/PDM files must be updated.

There are some simple ways of performing modifications & combination that we have found to be especially usable. These are *enlargement* and *reduction* [10], which can be called "parametric design". *Inversing* the solutions, which means to do the opposite [10], is sometimes useful. To *reorganise* or *recombine* the solutions (also called "crossover" [11]) is closely related to inversing solutions. Reorganising is useful especially to save space – and is sometimes called "packing". We have found it to be more creative to *combine, substitute,* and *eliminate* solutions. In order to verify the solutions it is important to perform simulations and tests as often as possible (see next section).

7. Verification

As has been pointed out it is extremely important to often perform simulations & tests in the development process. Doing so we have found verification only with one human sense to be less reliable. This is especially the case if the only used sense is the eyesight since the eye often cheats our judgement even if the presentation is on paper in scale 1:1. Pictures transformed on the computer screen or transformed in a VR helmet never give the same impression as when holding a model or a prototype in one's hands. Therefore our experience is that it is important to produce models and prototypes that the product developers and test people can hold in their hands and touch. If models are practically impossible to make in scale 1:1, we have found the old recommendations of scales 1:2, 1:4, 1:10 to be recommendable.

The earlier in the development process, the softer material should be used, as simple materials are fast and easy to change. From our experience, clay, paper, wood, etc., should therefore be used before using harder material. Quote: "When a model starts to harden up, so does also the thinking." [12, p 79]. The strength of rough prototyping media is also that they encourage playing with ideas, possibilities, and potential at a low cost. To control that design intentions are met, rapid prototyping (e.g. FFF – Free Form Fabrication) is often useful when function and dimensions are decided after some idea iterations.

Our experience has shown that the more people who are involved in the development process, the more important is it that models and prototypes are given to as many as possible. This is due to the fact that models and prototypes are produced to answer questions and to give impulses for the development that cannot be described in written or spoken form. Important to bear in mind is also that prototypes and simulations always are "political" while managing prototypes and simulations is about managing power and influence.

The productivity of prototype-driven design measured in user satisfaction per man-hour, has shown to be "superior" [12, p 73]. One reason is that the product development team – and test users - have the same mental picture to work from when they all are able to touch the model (sometimes called mock-ups) and later the physical prototypes. For design studies – especially of large objects – digital mock-ups (DMU) serve the same purpose.

Models and prototypes are thus important visualizing tools for developers, managers, users, and customers. Models and prototypes help us to get a better understanding of ourselves and our priorities and help us to avoid mistakes and misunderstandings. As a rule, the more simulations, prototypes, and prototyping cycles used per unit of time, the more useful and technically perfect the final product will be. Note, however, that frequent prototyping easily leads to adding more and more features to make the prototype even better, so that the product in the end will have more features than the user actually needs or wants to pay for. This is a problem especially in software development.

To cut down development costs and to shorten Time to Market, many attempts have been made to cut down on the number of prototypes in the development process. Such business cultures are sometimes called specification cultures ("spec-cultures"), which is the opposite of prototype-driven cultures which DPD is an example of, as we cannot foresee everything in a theoretical way. Due to our observations it is not unusual that companies can spend even thousands of hours developing detailed specifications that are invalidated by the initial prototype. "Spec-driven cultures" draw heavily from market-research data before concepts are moved into the prototyping cycle. In prototyping cultures, prototypes are typically used to elicit market feedback well before final versions of the product are tested.

References

[1] Ottosson, S (2001): Participation Action Research – a Key to Improved Knowledge of Management, Article on review at Tehnovation
[2] Nevins, J.L., and Whitney, D.L. (1989): Concurrent Design of Products and Processes, McGraw-Hill, New York
[3] Hundal, M.S. (1998): Time-Driven Product Development, in Integrated Product Development Process (eds. Usher, J.M, Roy, U. & Parsaei, H.R.), John Wiley & Sons, New York (pp 59-84)
[4] Ulrich, K.T. and Eppinger, S.D. (1995): Product Design & Development, Mc Graw Hill, New York
[5] Altschuller, G.S. (1988): Creativity is an Exact Science, Gordon and Brreach Science Publ., NY
[6] Knellner, G.F. (1965): The Art of Science of Creativity, Holt, Rinhart and Winston, NY
[7] Ottosson, S. (1995): Boosting Creativity in Technical Development, in W.E. Eder (ed), WDK 24: Engineering Design and Creativity, Proc of the Workshop EDC, Pilsen, Nov 1995, Heurista (pp 35-39)
[8] Ottosson, S. (1998): Qualified Product Concept Design needs a Proper Combination of PAD-MAD before PDM, Journal of Engineering Design, Vol. 9 No.2, pp. 107-119
[9] Hertzberger, H. (1991): Lessons for Students in Architecture, Uitgeverij 010, Rotterdam
[10] Evatt, M.A.C. (1996): Structured Creativity: A Minsnorm?, Engineering Design and Creativity (Eder, E. Editor), Heurista WDK 24, Zürich
[11] Vajna, S., Bercey, T., Clement, S. & Mack, P. (2000): Autogenetic Design Theory – A Contribution to an Extended Design Theory, 2000 ASME Design Engineering Technical Conference, Baltimore, Maryland, USA (September 10 – 13)
[12] Schrage, M. (2000): Serious Play, Harvard Business School Press, Boston, USA

Stig Ottosson, Professor in Innovation Management
Linköping University
S-60174 Norrköping
Sweden
E-mail: stig.ottosson@itn.liu.se

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

MASS CUSTOMIZATION OF CREATIVE DESIGNS

J S Gero

Keywords: Mass customisation, creative designs, entropic measures, complexity.

1 Introduction

Among a nation's goals are competitive leadership in the international marketplace and excellence in industrial productivity. Superior design, a fundamental prerequisite for superior products and systems, is one of the keys for achieving these goals. Computer aids to design (computer-aided design) have the potential to unlock this key. Whilst the early work on computer-aided design concentrated largely on drafting and then geometric modelling, more recent research has focussed on the development of concepts and tools for design decision making [1]. Where all the design variables take numeric values the processes of simulation can be used during design analysis and evaluation. In some cases simulation can be used with symbolic valued variables. When, in addition, a set of goals or objectives or fitnesses can be expressed in terms of those variables, the processes of optimisation can be used and we have a design optimisation or optimal design problems [2]. Evolutionary systems have proven to be particularly useful in design optimisation [3]]. When the variables are symbolic rather than numeric, artificial intelligence methods have proven to be useful [4].

Whilst much of what we use and consume is designed for mass production, there is an increasing demand for individual products. Some of these we are familiar with, such as purpose-designed buildings; however, there is the potential today to direct design individualised artefacts based on direct marketing [5]. Thus, instead of buying a product that has already been designed and manufactured, it may increasingly become possible to specify individual requirements that result in individualised artefacts. This brings with it the need to be able to design such products. Whilst the current demand is for variations of existing designs there is expected to be a demand for innovative or creative designs that are not simple variations of existing designs. As a consequence there will be an increasing need to produce such designs on demand.

This paper presents a strategy, with appropriate techniques, to develop the capacity to mass produce creative designs – mass customisation of creative designs.

2 Mass customisation of design

2.1 Mass production

The concept of mass customisation of design is analogous to the concept of unit production within the framework of mass production. In mass production, the cost of the infrastructure for production is spread over increasingly larger runs of the same or very similar products. The components can be shared between families of artefacts in order to increase the size of production runs. Mass production has a history going back to the nineteenth century but is epitomised by the early twentieth century production of the Ford motor car. The same concept of the benefits of scale of production continue to underlie not only the motor car industry but many other industries including the computer chip and computer memory industries.

With the investment in infrastructure for mass production it becomes possible to move to a variant of mass production: batch production where the same production line is used for large batches of a single product within a family of products that draw components from the same set of components as all members of that family.

2.2 Mass customisation

It is possible to extend batch production down to the production of a single unit, hence unit production still using the same mass production facilities. Unit production offers the capacity for mass customisation [6], [7]. Each product is a variant from a family of products produced from the same components. Currently, a number of computer hardware suppliers offer the opportunity to "customise and build" to order.

2.3 Mass customisation of design

Mass customisation of design requires an analogous infrastructure in designing as exists in mass production – a production line for designs. Such an infrastructure exists in the form of design processes for parametric or variant design [8]. A variety of such processes exist ranging from design optimisation [2], through designing using knowledge-based systems [4]. More recently, robust search techniques have been introduced into the engineering design process [3].

In all of these approaches the design space is fixed at the outset. The design space can be thought of as being comprised of three subspaces: Function, behaviour and structure subspaces [9]. A description represents the product's elements and their relationships, which are labelled *structure*. In designing, behaviour may be viewed in two ways. There is the behaviour of the structure, which is directly derivable from structure; it is how a structure performs in its environment in a measurable way. Transforming function to expected behaviours, provides the second view of behaviour. This provides the expectations that the product has to meet. Function has been defined in another context as "the relation between the goal of a human user and the behaviour of a system" [10]. Figure 1 shows the relationship between these three subspaces. These three subspaces constitute the state space of design.

Thus, we are able to produce mass customised designs on demand once we have specified the requirements, turned them into expected behaviours and decided on the variables that will be used to define the structure. From this point on the problem is one of finding values for the structure variables that optimise or satisfice the behaviours. In general, this form of mass customisation of designs produces families of products [11]. The design process searches the space of possible structures. Whilst the space of possible structures is defined by the decisions about what are the structure variables and any constraints on the ranges of those variables, whether those constraints are explicit or implicit.

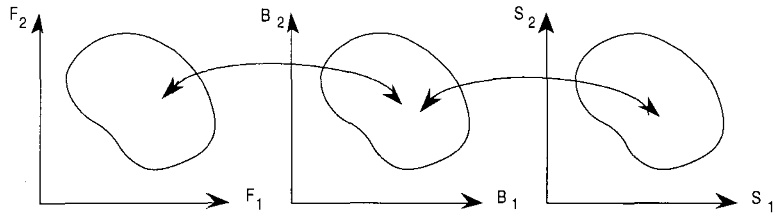


Figure 1. The three subspaces of function (F), behavior (B) and structure (S) which constitute the state space of designs, plus the locus of the transformations between them.

As we shall see later in this paper, the concept of a fixed space of possible structures plays an important role in our conception of creative designs. The space of possible structures is fixed by a human designer outside the formal design process in this form of mass customisation of designs.

3 The problem of mass design customisation of creative designs

It is convenient to characterise designing as routine or non-routine, although there are other ways of categorizing designing processes. *Routine designing*, in computational terms, can be defined as that designing activity which occurs when all the necessary knowledge is available. It may be more formally expressed as being that designing activity which occurs when all the knowledge about the variables, objectives expressed in terms of those variables, constraints expressed in terms of those variables and the processes needed to find values for those variables, are all known *a priori*. In addition, routine designing operates within a context that constrains the available ranges of the values for the variables through good design practice. Figure 2 show graphically the notion of the state space of routine designs being bounded by a set of a priori decisions and constraints.

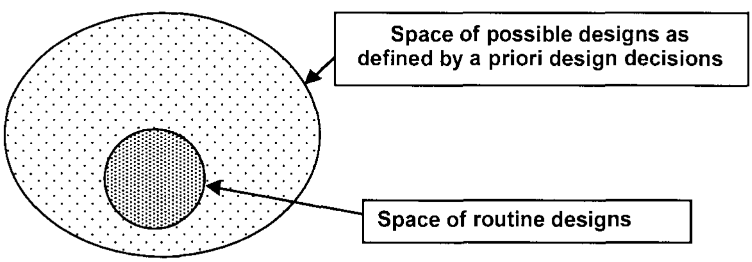


Figure 2. The space of possible designs is defined by the set of *a priori* decisions. The space of routine designs is a subset of those possible designs.

Non-routine designing can be subdivided into two further groups: innovative designing and creative designing. *Innovative designing*, in computational terms, can be defined as that designing activity that occurs when the constraints on the available ranges of the values for the variables are relaxed so that unexpected values become possible, Figure 3. This produces two effects, one for the design process and the other for the product or artifact.

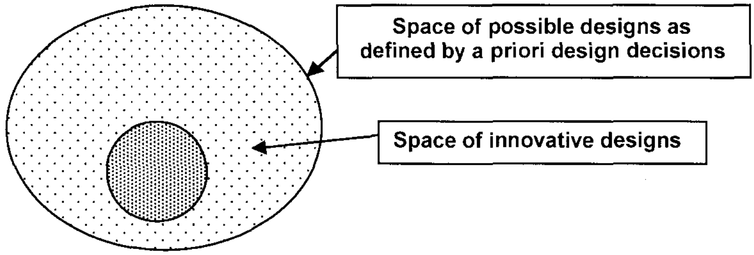


Figure 3. The space of innovative designs is a subset of the possible designs.

In terms of the design process, variable values outside the usual ranges have the potential to introduce unexpected as well as unintended behaviours that can only be brought into formal existence if additional knowledge capable of describing them can be introduced. For example, in designing a structural beam to carry a load across a gap there are standard depth-to-span ratios for different materials. If the depth of the beam is made much larger than these then there is the likelihood that the beam will buckle. However, if no buckling knowledge is applied to its design (and buckling is not normally considered in the design of such beams) then no buckling behaviour will be found. In terms of the artifact, innovative designing processes produce designs that recognizably belong to the same class as their routine progenitors but are also 'new'.

Creative designing, in computational terms, can be defined as the designing activity that occurs when one or more new variables is introduced into the design. Processes that carry out this introduction are called "creative designing processes". Such processes do not guarantee that the artifact is judged to be creative, rather these processes have the potential to aid in the design of creative artifacts. Thus, creative designing, by introducing new variables, has the capacity to produce novel designs and as a result extends or moves the state space of potential designs, Figure 4. In the extreme case a new and disjoint state space is produced that results in a new type of design. Creative designing has the capacity to produce such a paradigm shift.

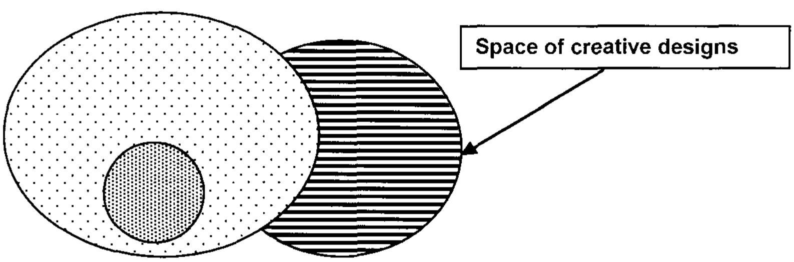


Figure 4. The space of creative designs is a superset of the possible designs, as defined by the set of *a priori* decisions.

The problem remains how is it possible to develop the infrastructure for the generation of creative designs so that it can be used for the mass customisation of creative designs. The mass customisation of routine designs is possible because we have such an infrastructure in the generic processes of design optimisation. Even though we might use different optimisation techniques we are using the same fundamental approach: searching a space of possible designs defined at the outset by the variables. In creative designing, since we aim to change the variables that are used to define the space of possible designs we cannot use such an approach. Further, in routine designing we know the performance of the techniques we apply. The performance is in terms of computational complexity of the search processes. In creative designing we need to be able to determine both the computational complexity of searching a novel state space and more importantly, the potential of a state space of possible designs in terms of the range of designs that can be generated within it.

4 Complexity measures for mass customisation of creative designs

Conceptually, modern design theory views the design process as a search in a predefined space of possible designs. This design space is implicitly fixed by defining its generator (a process that can generate any design in this space). Many choices for such a generic design generators are available – shape grammars [12], rule-based systems [4], and evolutionary systems [3] are examples. Thus, designing with a CAD system is then understood as running a search method coupled with this generator with decisions governed by a behaviour function.

This raises a fundamental question: how can design generators be compared? If the design generator is not fixed a priori but can be modified as part of the design process then a further question arises: how can two modifications of the generator be compared? Currently there are no adequate methods for this. Thus a test is needed which evaluates either the generator or the design space produced by that generator. This evaluation can produce a measure of the design potential of the corresponding space as well as a measure of the hardness of the typical search in this space. The higher the potential the more likely it is that creative designs will be generated (mass customisation of creative designs). The hardness of the search is the measure that describes the computational scalability of any search process. Once these measures are constructed then we are able to compare different design generators, to modify them so that they produces better design spaces, etc. A design space is typically a very large space (often unbounded) and the cost of constructing and evaluating any individual design in it can be high. A design space typically is also an ill-defined space, with no analytical models, no "good" properties available. This makes a direct evaluation of the space very difficult.

The approach we have adopted it to measure the complexity of a typical sample of designs in a state space produced by a creative designing process so as to use that as the basis for determination of the computational hardness and design potential of that state space of possible designs.

4.1 Complexity measures for designs

We take concepts from information theory [13] as the basis for a general approach to the development of measures of computational hardness and design potential both for individual designs and populations of designs. Design complexity measures have been used in axiomatic design [14], which postulates that the best design has the lowest information content. The structural design complexity measures, which were developed by the practitioners of axiomatic design [15], [16], are based on the application of the simplest tool of information theory - Shannon entropy - within a linear symbolic representation. This restriction to linear representations makes it inapplicable for a large majority of design problems, which have non-linear representations, where design descriptions are short, etc. As a consequence we will utilise some more modern tools of information theory (Lempel-Ziv complexity, approximate entropy) as estimators of design hardness and potential. Unlike Shannon entropy they are readily computable and meaningful even for fairly short descriptions.

4.2 Complexity measures for design spaces

Overall, we intend to use the notion of a family of design space generators. That is, a meta-generator exists, which has parameters that can be set to produce different generators. This comes from the work on families and individuals. A typical example of two families in say mechanical energy transmission design might be hydraulic systems and gear systems with spaces of individual designs within each family.

Once we have developed the means to measure the potential of design space and the hardness of designing in this space then we are in a position to control the modification of the generator so that it generates better design spaces. Thus a control strategy needs to be designed for generator modification that balances the increase in the hardness of design process against the gains due to increasing the potential of design space. Design processes are often modelled as randomised algorithms [17]. This gives rise to additional performance variability (beyond the performance variability caused by varying the design space) even when the design space they operate on is fixed and repeated trials are run on a single design space. This implies that there is an inherent risk associated with such design processes. This risk can be quantified, through an analogy with economic risk, as the standard deviation of its performance distribution. Hence, it is possible to interpret this situation as the estimation of the risk of getting a lower average performance in exchange for increasing certainty in obtaining a reasonable design.

4.3 Complexity measures as a basis for mass customisation of creative designs

We can work with individual designs and explore multiple representations of that design. Each different representation of a design produces a different complexity measure and hence a different potential. This notion of re-representation plays an important role in the production of novel designs [18]. Or we can work with a design space and sample designs within that design space. The complexity of individual designs can be measured from their qualitative (symbolic) representation [19]. The complexity can be calculated from both re-represented individual designs and from populations of designs. The results of these calculations can be used to measure both the hardness of future computations (ie how difficult it will be to search the space of possible designs) and the potential of the space of those designs (ie the variability that can be produced).

Then, based on either these measures we can carry out a statistical sampling of a range of representational spaces derived from individual designs or from a sampling of design spaces to determine which of them has more potential or we can use symbolic reasoning [4]. These measures can be used as one of the objectives or fitnesses for any generator that is a synthesis process capable of generating creative designs [20]. Once we have this we are capable of mass customising creative designs. Figure 5 shows an outline architecture of such a system where the complexity of the available designs in measured and used to control the generation of mass customised designs with individual requirements.

For example, the complexity of beam cross-sections generated using a rule-based system can be measured and that complexity used as a fitness in a genetic algorithm as the design generator. The complexity measure can be analysed to determine both the hardness and potential of the design space produced. If the fitness is set to maximise complexity then an increase in variability will result. As variability increases so does the potential for creative designs.

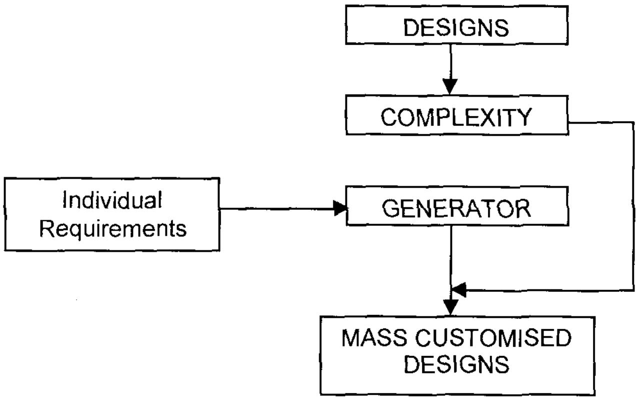


Figure 5. Outline architecture of a system for mass design customisation of creative designs

4.4 Discussion

Mass customising designs involves having the infrastructure for the mass production of designs so that individual designs can be produced on demand. In order to mass customise creative designs we need a richer infrastructure since we need to be able to determine the computational hardness and design potential of any design generator. This is too difficult a task, so we determine the hardness and potential of a space of designs by sampling individuals in that space and use measures of complexity as the basis of our results. We then use the results to control synthesis processes. As a consequence it becomes possible to produce creative designs on demand – mass customisation of creative designs.

4.5 References

[1] Mistree, F, Smith, W, Bras, B, Allen, J and Muster, D., "Decision-based Design: A Contemporary Paradigm for Ship Design", Trans Soc. Naval Architects and Marine Engineers, 16,1990, p565-597.

[2] Papalambros, P. and Wilde. D., "Principles of Optimal Design", Cambridge University Press, Cambridge, 2000.

[3] Bentley, P. (ed.), "Evolutionary Design by Computers", Morgan Kaufmann, San Francisco, 1999.

[4] Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M. B. and Gero, J. S., "Knowledge-Based Design Systems", Addison-Wesley, Reading, 1990.

[5] Anonymous, "Direct Hit", The Economist, 350, 1999, p57-59.

[6] Pine, B. J., "Mass Customization", Harvard Business School Press, Boston, 1993.

[7] Meyer, M., Tertzakian, P. and Utterback, J., "Metrics for Management research and Development in the Context of Product Family", Management Science, 43, 1997, p88-111.

[8] Suh, N. P., "The Principles of Design", Oxford University Press, NY, 1990.

[9] Gero, J. S., "Design Prototypes: A New Knowledge Representation Schema for Design", AI Magazine, 11(4), 1990, p26-36.

[10] Bobrow, D., "Qualitative Reasoning about Physical Systems: An Introduction", Artificial Intelligence, 24(1-3), 1984, p1-5.

[11] Jiao, J., Mitchell, M., Duffy, V. and Lin, F., "Product Family modelling for Mass Customization", Computers in Industrial Engineering, 35(3-4), 1998, p495-498.

[12] Stiny, G., "Introduction to Shape and Shape Grammars", Environment and Planning B, 7, 1980, p343-351.

[13] Cover, T. M. and Thomas, J. A., "Elements of Information Theory", Wiley, New York, 1991.

[14] Suh, N.P., Bell, A.C., and Gossard, D., "On an Axiomatic Approach to Manufacturing Systems", Journal of Engineering and Industry, Trans. A.S.M.E., 100, 1978, p127-130.

[15] El-Haik, B. M. and Yang, K., "The Components of Complexity Engineering Design", IIE Trans, 3(10), 1999, p925-934.

[16] Braha, B. and Maimon, O., "Measurement of a Design Structural and Functional Complexity", IEEE Transaction on Systems, Man, and Cybernetics, A, 28, 1998, p4 527-534.

[17] Gero, J. S. and Kazakov, V. "Adaptive Enlargement of State Spaces in Evolutionary Designing", AIEDAM, 14(1), 2000, p31-38.

[18] Damski, J. and Gero, J. S., "A Model of Shape Emergence Based on Human Perception", Information Technology in Design, Vol. 2, ICSTI, Moscow, 1994, p96-105.

[19] Gero, J. S. and Kazakov, V. "Entropic similarity and complexity measures for architectural drawings", Working Paper, Key Centre of Design Computing and Cognition, University of Sydney, Australia, 2001.

[20] Gero, J. S. and Maher, M. L. (eds) (1993). "Modeling Creativity and Knowledge-Based Creative Design", Lawrence Erlbaum, Hillsdale, New Jersey, 354pp.

Acknowledgements

This work is supported by a Sesqui Grant from the University of Sydney. Ongoing discussions with Vladimir Kazakov of the Key Centre of Design Computing and Cognition, University of Sydney are gratefully acknowledged.

John S. Gero
Key Centre of Design Computing and Cognition, Department of Architectural and Design Science, University of Sydney, NSW, 2006 Australia, Tel: +61-2-9351 2328, Fax: +61-2-9351 3031, E-mail: john@arch.usyd.edu.au, URL: http://www.arch.usyd.edu.au/~john/

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

PRODUCT INNOVATION ON DEMAND – FICTION OR TRUTH?

U Lindemann

Keywords: Drivers to innovation, innovative products, principles of innovation

1 Introduction

Product innovation may be generated on demand by using, combining and modifying different methods. For a successful product innovation, there has to be a well balanced network of processes like the product planning process, the concept process, the prototype process, the marketing process etc. A failure of at least one of these innovation processes can be caused by a number of reasons, concerning the structure of the company, the individuals or regulations. It is the purpose of this paper to demonstrate, that successful product innovation may be generated with help of a set of methods in conjunction with a well established team.

2 Product Innovation by new Product Concepts

Innovation is the successful transfer of requirements into products, which are successfully accepted by the market and include excitement needs following the Kano Model. Currently, many of literatures are available about innovation. Politicians like to talk about it as well as the top management in industry. There is a big variety of innovation types in processes as manufacturing, maintenance, marketing etc. as well as in the technology of products. Concerned to product innovation we are aware of new materials and material composites, electronics and sensors etc. Still, there is an additional door to innovation which is used far too seldom. The advantage of this way of innovation is the limited investment and limited risk compared to the development of new materials, new electronic equipment etc. This way is the systematic way of product development, which essentially means using all helpful methods and systematic tools available on the market. The key question is, weather the right measure of combination and adaptation of methods dependent on the specific situation.

The overall innovation process concerned to products consists of several sub-processes or partial processes. There is a requirement for a strategic planing process, a process of market introduction, advertising, service etc. The focus of the paper is mainly related to the process of the development of innovative product concepts.

By a number of different projects the situation specific mixture of methods was tested and the experience out of these projects is condensed in this paper.

2.1 Conceptual Design – the Classical Approach

The Industrial Approach

Industry and politicians claim a lack of innovation. Time pressure and cost pressure are recognized as their main problems. Outsourcing of development tasks to engineering companies or to sub-suppliers may help for a while, but reduction of competencies may be a resulting problem. We observed a number of development processes in industry and other authors have done it in a similar way. The results show there is a number of reasons beyond lack of time that prevent effectiveness of processes and innovative results [3, 4].

The Scientific Approach

There are different schools [1, 2, 6] with partly different approaches. Some suggest mainly the systematic way of development strategies and methods, others the creative way. The picture is not just like black and white, but these are two main focuses. A comparison was done by Lauche [5], she discussed the two different approaches and the specific advantages. The main streams of scientific discussion meet and discuss their views on international meetings, but a deeper understanding should be developed by common research and product development projects.

The Success of Innovation

Both approaches – the industrial and the scientific one – generate innovation, innovative solutions. But there are a number of questions to be answered. Why are there deficiencies in innovation? Why do innovation processes fail?

Nevertheless the above approaches exist side by side and the question is, if there might be a fruitful collaboration between scientists and industry to improve both approaches. This kind of successful co-operation already exists in a number of countries, and it may be improved by the given results.

2.2 The Experiments

Within the past 5 years, we have been working on more than hundred product development projects, many of them have been team oriented. These projects have had different volumes, the structure of the projects was differentiated. The development of new product concepts and the verification of these concepts concerning the main targets as function, cost, weight etc. were the normal task. Quite often, we ended up with a tested prototype, sometimes with a complete range of possible solutions. The main focus of this paper will be laid on team oriented projects. There were two kinds of projects – projects mainly run by students (student projects) and projects mainly run by a team of researchers plus industry members (research projects).

Student Projects

Typically, the teams existed of about 5 students (third year) as the working team plus one or two researchers as their trainers and coaches. In addition, there was a staff from the industrial partner, who supplied the specific knowledge of the product, the use of the product etc., and took over the role of the customer of the team. The usual time frame was about 5 months.

Research Projects

Usually two researchers from the university worked together with three designers from the industry and built a mixed team. These projects used to last in average of 2 years and the researchers invested more than 2000 working hours each. These figures are average values, to give an idea of the total size of the projects.

Example 1 – a Student Project [8]

- The Product

The ship's propulsion system allows exact navigation in every direction and includes turning on the spot. Thrust is generated by blades that are located underneath the ship's body. Those blades rotate around an axis rectangular to the water surface. During the rotation, the axes that are rectangular to the symmetrical axes of the blades intersect at one point, called the steering centre (N; see figure 1). If the steering centre is identical with the propeller centre (O; see figure 1) the system does not generate any resulting velocity, which then prevents the ship from moving. If the steering centre (N) is moved out of the propeller centre (O), the blades oscillate during rotation. The angle of attack from the flow on each blade is oscillating; therefore, a lift force is generated by each blade. The lift force changes in magnitude and in direction so a resulting velocity is generated in strictly one direction. Usually, two propellers, that are spinning differently, are located underneath each ship and the angular momentum from each propeller is balanced so the rotation of the ship on the spot is possible.

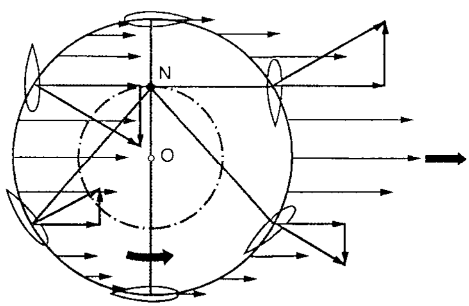


Figure 1. principle solution of the propulsion system

- The Project

Goal of the project was the development of innovative concepts for a ship's propulsion system, that will keep up the functional advantages of the existing Voith-Schneider-Propeller while improving its concept concerning weight, volume, cost, efficiency, maintainability and reliability.

- The Team

The team carrying out the concept project consisted in this case of seven students in mechanical engineering, each with a special knowledge in different sub-disciplines, and two research assistants. During weekly team sessions, it has proven to be effective to distribute different roles among the session participants. Those roles were the session management, the moderation, the record keeping and the time management. The team sponsoring was taken over by the two research assistants. The experiences with these strictly defined roles were very good.

- The Methods

In order to realise the required revision of the ship propulsion system within a short period of time and moderate cost, they followed a structured and methodical procedure. In the beginning of the project a list of requirements was developed. The requirements were documented as neutral as possible, including specific numerical values (if necessary and/or available) and with reference to the person who stated it.

To find new solutions, they used several innovation stimulating methods, e.g. brainstorming, the method 6-3-5, and parts of the TRIZ (Theory of Inventive Problem Solving) [9] method. Also they used catalogues with physical effects and different check lists. In the beginning a great number of new ideas was produced, but many of them were not corresponding with the list of requirements.

In order to find out better concepts for the further handling the next step was an evaluation and a pre-selection on the basis of different criteria. For this they used the portfolio diagram. Concepts with similar attributes (for example: the relevance to the customer, cost, time to market) are in the same area of the diagram. In order to prove the suitability of the different concepts in the early phases of the development process, different methods and tools were used, for example orienting attempts to examine relevant product properties. For this they used the equipment of the development lab of the institute, which contains different measuring and testing facilities. Therefore, they got a number of verified concepts at the end of the project.

- The Results

The original goal was the education in design methods and this was fulfilled in an excellent way. In addition, there was a very good result for the industrial partner: "That`s it, what we require", the design manager concluded.

By the use of several methods, new tools and different roles in the team about 50 different ideas had been elaborated within a short period of time. Now the Company is verifying our concepts and is going to select the best of them for realisation. As many of process related details are similar to the student projects, only the most important differences will be discussed here.

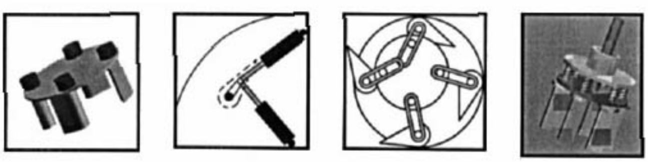


Figure 2. four concepts out of about 50 concept ideas

Example 2 – a Research Project [7]

- The Product

The product within this project was a safety brake based on dry friction steel versus steel. A typical application is industrial products like robots. The requirements of the market asked for new solutions with higher functionality, which the given product did not meet.

- The Project

The industrial partner and the institute of the university agreed to a common research project of in total about three years. The objective were new concepts to build safety brakes for the given application. During the total project, as additional goals, the improvement of the old concept as well as new product ideas with new concepts based on the given technology were added.

- The Team

The team consisted of one researcher (sometimes two) plus three participants from the industry. The team met regularly every four to six weeks for a two day workshop. The rest of the time was filled up with individual work plus other projects and tasks.

- The Methods

The methods used were similar as in the student projects, the range and the intensity were clearly higher.

- The Results

A number of improvements of the given product was defined and verified. Additional solutions based on different concepts were found and documented, some of them had been proven by prototype tests. New product ideas for new market areas based on the well known technology came up, some had been verified. Out of these results, the company started a project of strategic product planning.

Figure 3. safety brake – known concept and one of the new concepts [7]

Examples of Further Projects

- Connecting running devices of handicapped wheel-chairs - the result is an European patent, the company started series production.

- Snow board binding - at the end of the project series, production already started and sales numbers are higher than expected, the production costs are lower than planned.

- Safety valve – the number of parts was halved, the function was improved, the management was not interested because of other main focuses.

- Grinding machine –the prototype is assembled and tested in these days.

- Scanning device for the pharmaceutical industry - patent in preparation, device is built as a prototype and is tested.

Table 1 shows some examples of the projects and their results. Change of the management was one of the reasons, no specific interest was the other one. The weakness of implementation hindered successful innovation, too.

There was hardly no indication of any problems in the basic innovation process within the product development process.

project examples	type of project	main inventor	delays in realisation	problems in / with industry	patent	realisation	success
.....	...	...	...	...	...	...	
connecting device	student	student	yes change of management	no	yes	yes	yes
snow board binding	student	student	no	no	no	yes	yes
safety valve	student	student	yes	yes – not interested	open	open	probably no
brake with perm. magnet	research	researcher	no	no	yes (several)	yes	probably yes
grinding machine	research	researcher	no	no	open	yes	probably yes
scanning device sensor	student	student	no	no	in progress	yes	yes
ship propulsion system	student	student	yes change of management	no	no	still in progress	open
.....	...	...	...	...	...	...	

Table 1. overview of project results (excerpt)

2.3 The general experiences

We use the number of experiments as a set of case studies to be evaluated. The experiences we found in these projects have been found nearly independent of the students and researchers involved, the product to be developed, and the branch of industry. There was a large impact out of the companies culture and sometimes of the management within the company.

- The team: The positive aspects were the different background, the mixture of young and old, of people with and without product related experience, of team members from outside the company, of ideas from outside, of questions without history, etc.
- The methodical approach: The mixture of methods (brainstorming, brain-writing, gallery, functional analysis and other TRIZ-methods, QFD, design review ...) as with different intensity and adapted to the situation, intensive work in early phases.
- The coaching of the team (supported, not dominated): In student teams, the coach was intensively integrated, he suggested methods and strategies, he came up with additional questions, he looked for additional quality aspects.
- The "open view": As external participants students and researchers came up with questions and ideas without bothering to blame themselves.

- The motivation: The students and researchers were highly motivated, there were some pressure and competition within the team too.

The organisational conditions had an important impact, workshops were running without interrupts within the university facilities and there was a common understanding of problems, methods and processes.

2.4 The Results

Depending on the resistance to innovation resulting from different reasons, the success of all the projects was different. There may be discussion of four different classes of resistance.

- The insufficient availability of complete technical solutions of the partial systems of an innovative product may be a reason for a failed attempt of innovation. This happened, for example, during the invention of the air-bag in the 50's, when, for example sufficient sensors and gas generators were not available. This did not happen in all the projects we have worked on.

- It may happen, that the available qualification of the staff does not meet the requirements of the technology of the innovative attempt. This happened only in less than 10% of our projects. In these cases, the innovation process was stopped by our industrial partner at least for a couple of months or years, sometimes the changes in the management activated the innovation process again.

- Individuals and their social problems caused a number of accounts of resistance in different levels of the hierarchy of the companies. These difficulties were the most important problem we observed. e.g. NIH, lack of co-operation, lack of target orientation

- Regulations of the companies, their insurance companies, the public and laws were just few important reasons preventing innovation.

We identified a number of conditions, which are required for successful product innovation processes (fig. 4). We were able, to establish most of these conditions as discussed above. Within our projects we were not in a position, to manage the complete network of required processes beyond product development.

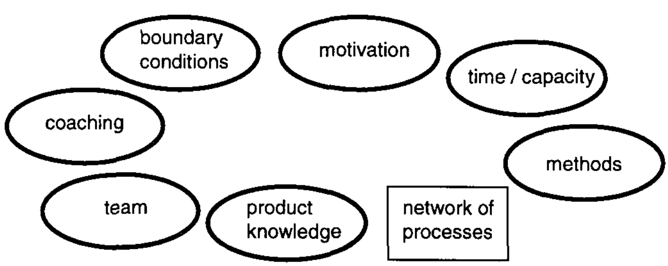


Figure 4. elements of success within our projects

Beside these difficulties for real successful innovation, we had a hundred percent positive result concerned to technical as well as market oriented objectives. Rules for selection, combination and adaptation were derived from the processes and will be discussed. Influences from the situation described by technical, project, individual, as well as organisational questions are taken into consideration.

3 Conclusion

It is possible to generate innovation on demand, if the boundary condition allows these processes. It is possible to select, combine and adapt methods in an objective oriented way. In addition to the innovative product development, a number of additional innovation processes like marketing, sales etc. are necessary for the overall success. The top management is responsible for the total innovation process. If this condition is fulfilled, then we are sure, that innovation on demand is possible.

References

[1] Bender, B.: Zielorientiertes Kooperationsmanagement in der Produktentwicklung. Dissertation TU München 2001. Reihe Produktentwicklung.

[2] Ehrlenspiel, K.: Integrierte Produktentwicklung. München, Hanser 1995

[3] Gausemeier, J et al: Kooperatives Produktengineering. Paderborn: HNI 2000.

[4] Grabowski, H.; Geiger, K.: Neue Wege zur Produktentwicklung. Stuttgart: Raabe 1997.

[5] Lauche, K.: Qualitätshandeln in der Produktentwicklung. Dissertation Universität Potsdam 2001

[6] Pahl, G; Beitz, W.: Engineering Design. London, Springer 1996.

[7] Schwankl, L.: Development Process of Innovative Brake Systems. In: Marjanovic, D.: International Design Conference 2000. Proceedings of the 6 th International Design Conference, Dubrovnik, 23.-26.05.2000, S. 693-698.

[8] Schwankl, L.; Pache, M.: Collaboration between University and Industry in Engineering Design. In: Scrivener, S. (Eds.): Adjunct Proceedings of Collaborative Design. CoDesign 2000, 11.-13.09.2000, Coventry. Coventry: Univ. 2000.(S. 185-190)

[9] Terninko, J., Zusman, A., Zlotin, B., "STEP-by-STEP TRIZ: Creating Innovative Solution Concepts", Ideation Int., Nottingham (New Hampshire), 1996

Prof. Dr.-Ing. Udo Lindemann
Lehrstuhl für Produktentwicklung, Technische Universität München,
D 85748 Garching, Boltzmannstr. 15, Germany
Tel: +49 – 89 – 289 151 30
Fax: +49 – 89 – 289 151 44
E-mail: lindemann@pe.mw.tum.de
URL: – www.pe.mw.tum.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

EXPERIENCE OF CREATIVITY ENHANCEMENT TEACHING IN INDUSTRY AND UNIVERSITY

G Thompson and M M Lordan

Keywords: Human creativity, psychology of creativity, creative education

1. Introduction.

Research into Creativity and Creative Problem Solving (CPS) has been carried out for many years. Important publications can be traced from Wallas (1926) through Osborn (1953), Gordon (1961), De Bono (1969), Parnes et al (1977), Rickards (1988), Kirton (1989), Parnes (1993), Isaksen et al (1993) to the research output in journals of the present date [1-8]. To write such a small list is to invite criticisms of omission, but the objective here is simply to highlight the many years of in-depth research that has been carried out.

However, much of this research has not found its way into engineering design education. Probably this is due to the fact that the terminology used is unfamiliar to engineers and that the ideas can appear vague when compared to precise engineering calculations. Also, the view often prevails in engineering circles that some people posses a natural creative talent whilst others have little or none at all.

The objective of this paper is to report experience in developing and delivering creativity enhancement workshops for engineers in industry. Additionally, experience of the use of the workshops with senior students in university is also reported. The workshops are based on extensive research into creativity enhancement [9].

2. Creative problem solving, creative style and solution requirements.

The teaching programme begins with the well established Creative Problem Solving approach originated by Osborn [2] with contributions by many researchers (notable from Buffalo State College, NY) and summarised by Isaksen [10]. The basic elements of the process are given in figure 1. Each stage of the process comprises a divergent-convergent thinking activity. The importance of divergent-convergent thinking and the use of suspended judgement in the divergent phase is underlined in the workshops. Throughout the workshop emphasis is placed on the need to use particular thinking skills at appropriate times in the CPS process . The problem solving process in figure 1 is compared with well known models of the engineering design process [11-12]. By drawing attention to the compatibility of the models, i.e., the ability to integrate the CPS activities into any stage of the engineering design process, no conflict with existing design education is introduced.

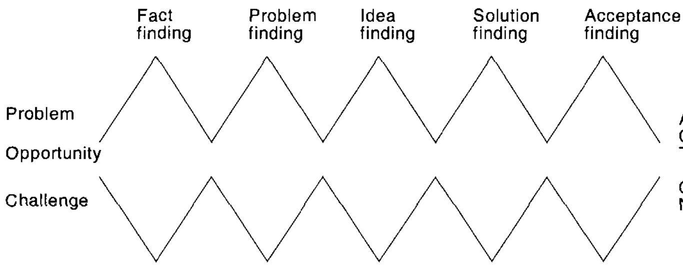


Figure 1 Creative problem solving process.

The model is not described as a rigid process, but rather as a means by which the right problem to be solved can be identified, a solution found and a means of implementing the solution sought. The existence of a series of divergent-convergent stages is more important than a precise number of stages and the aim of producing a practical outcome is important in an engineering context. Creativity is presented as a process by which practical outcomes are found and not as some zany open-ended ideas generating session.

Differentiating creative style from creative ability is a key aspect of the teaching programme. Following Kirton, [8], the teaching programme emphasises the difference between creative style and creative ability. Creative style is not an indicator of creative ability. Each person has a preferred style that can be measured on a continuum from extremely adaptive to extremely innovative. Innovators prefer solutions that break the paradigm whilst adaptors prefer incremental change. Innovators prefer to do things differently whilst adaptors tend to make improvements to existing solutions. It is a common misconception that innovators are seen as being more creative than adaptors. Innovators and adaptors have different creative styles which says nothing about their creative ability. Each person has a preferred creative style which is stable over the years. When working in a style which is not their preference then a person is subject to stress.

The use of 'flexing' in a controlled manner is advocated in the teaching programme. ‘Flexing’ is the ability of an individual to adopt a different creative style to their preferred style according to need. Of course, the person may become uncomfortable if they use a non-preferred style for a prolonged period. However, an engineering designer can do this more safely if flexing is performed with a knowledge of style and an understanding of the significance of their actions.

The reason for adopting a non-preferred style is to satisfy solution requirements. An engineering problem may require an innovative solution (one that involves a step change or a break from the norm), an adaptive solution (one that involves incremental change or an improvement to an existing solution) or a solution that is somewhere between these extremes. It would be inappropriate to pursue an innovative creative problem solving approach if the refinement of a product leader would be more profitable. Similarly, it is pointless to pursue improvements to an existing product if market analysis shows that a step change is required to leap ahead of competitors. Therefore, different solution requirements may require a more adaptive or innovative approach than a person's preferred style would normally produce. A person would normally use a method that fits their preferred style and thus their results may be ineffective.

Therefore creative style is a major part of the teaching programme. Most engineers recognise their personal style attributes and those of their colleagues. They also understand the usefulness of different styles and how different people can work effectively together in an atmosphere of understanding. Each person has their style measured using the Kirton Adaptor/Innovator inventory and this stimulates interest in the workshop. Matching style to problem solution requirements is a way to achieve high quality design. Additionally, by relating creativity to engineering problems and the engineering business in general, the notions of creativity as an 'abstract activity', 'only for game playing' or 'not for practical engineering situations' are dispelled.

3. Content of the teaching workshops.

It is not possible to give full details of the teaching programme in the confines of this paper. The outline programme to the workshops is given in figure 2. Typically, the workshops are of 7 hours duration. They have been run from 9 a.m. to 5 p.m. and from 1.30 p.m. into the evening. It was found that companies were reluctant to send their staff onto a creativity programme of longer than 1 day.

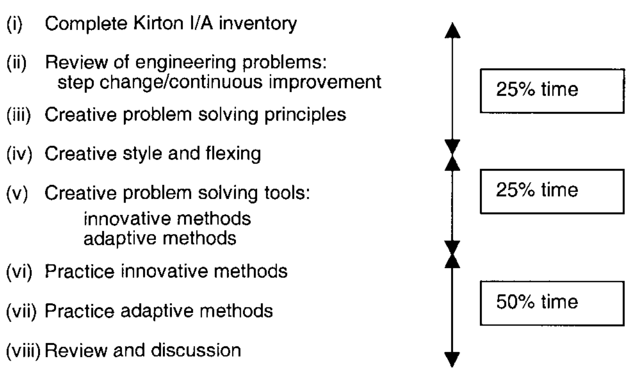


Figure 2 Outline of the creativity workshops

The workshops begin with each person completing a Kirton Adaptor/Innovator form to asses their preferred style, the results are fed back later. Then a review of engineering problems is undertaken to set the engineering tone of the workshop. Examples are given of step change and continuous improvements in a range of applications including product design and process industries. The economic significance of adaptive change is emphasised to dispel the notion that creativity is only associated with 'ground breaking' step change. The creativity principles taught include the CPS process, creativity = F(knowledge, imagination, evaluation), person-product-process-press and certain thinking skills.

The divergent creativity problem solving tools that have been used successfully in the workshops include brainstorming, brainwriting, forced relationships, invitational stems, morphological analysis and ladder of abstraction. The use of excursions has met with partial success. The convergent tools found useful are ALU^{O} (Advantages, Limitations, Uniqueness and Opportunities), voting, paired comparison analysis and ranking and weighting. The last method is commonly taught in engineering education and little time is spent on it. Paired comparison analysis is similar to Pugh's method [12] of concept evaluation which, in spite of it being extremely good, appears not to be known by many engineers in industry.

There are opportunities to undertake certain comparisons of methods within the teaching programme. A comparison of brainwriting with brainstorming has been carried out. A particular engineering problem, which can be understood by engineers of diverse backgrounds, has been presented to groups of graduate engineers. The problem is real and has two interesting solutions, one of adaptive style and one innovative. The usual method to tackle the problem for most engineers is brainstorming since they have used this method in their work or in undergraduate classes. In this teaching programme, brainstorming and brainwriting were used by groups of mixed adaptive/innovative style to tackle the same problem. In every case, brainwriting produced both solutions. Brainstorming tended to produce only the innovative solution, which would be expected from such an innovative style method. In the teaching sessions this result highlights that, if a method is used that favours a particular style, then solutions of predominantly that style will be produced. Therefore it is important to use the right method for the problem solution requirements.

In order to demonstrate the relevance of creativity methods, engineering problems are *always* used for the practice sessions. Usually the problems are provided by group members, except for the 'test' example mentioned in the preceding paragraph.

Each engineer on the programme has their adaptor/innovator preference assessed. In the practice sessions, the innovators are encouraged to use adaptive methods, and vice versa. In this way the ability to flex between styles is developed. Innovative style and adaptive style methods are stated to be of equal importance but, in order to counter the common perception of equating innovative style with creative ability, the importance of adaptive style creativity is stressed. The inclusion of adaptive methods helps promote the acceptance of CPS for use in engineering applications.

The role of individual creative work is discussed. Creative work in industry is carried out at an individual level and in group activity. The usefulness of the process and tools to the individual is emphasised which helps gain acceptance of the methods. Too often creativity courses are group exercises devised by innovators for innovators and so fail on two counts: they neglect the individual and they neglect the adaptive style of work.

The teaching workshops are designed to *enhance* creative ability. The aim is to ensure that everyone recognises that they have the potential to be creative and that their creative ability is expressed through a particular style. The workshops are intended to develop an individual's ability within their own creative style and to help the individual adopt another style when appropriate. Thus, a person's all round creative ability is enhanced. No attempt made to measure creative ability, which is itself a controversial subject. The objective is to emphasise that all have a creative contribution to make and that the workshop teaching programme is intended to help everyone present. There is no purpose served by trying to measure how good someone is at the workshop since comparisons will inevitably be made between individuals which would be counter productive. The effectiveness of the teaching programme was assessed by feedback and by an impact study to asses retained improvements in creative behaviour.

4. Response to the workshops.

4.1 Industry.

Creativity workshops have been provided for medium to large engineering companies in manufacturing industry. In total, some 50 design and design/development engineers have attended the workshops. Following each workshop a questionnaire was used to gain immediate feedback and the following comments summarise and reflect the responses received.

Benefits:
- The use of analogy was found to be helpful.
- An understanding creativity principles and habits was illuminating and useful.
- The importance of the need to suspend judgement in divergent thinking was recognised.
- The destructiveness of criticism was realised.

Difficulties:
- A sense of insecurity was felt by some participants.
- A tendency to censor ideas prematurely remained in spite of an understanding that this should not be done
- It was hard to distance oneself from the detail of problems.
- The need to keep ask questions during ideas generation sessions was felt.
  (Engineering education and practice is based on asking questions, seeking more information and challenging ideas. There is a need to justify one's opinions in the workplace. It is not surprising therefore that this difficulty prevails in creativity.)

Feedback: personal.
The following personal comments on the workshops were also reported.
- Innovative approaches were seen as stimulating.
- Adaptive methods were generally accepted as being useful.
- Adaptors realised the benefits of brainstorming when conducted properly, but they need time to become comfortable with the technique.

- The use of creative style was received very well. Participants, almost universally, warmed to the concept and recognised the attributes of their colleagues. It proved to be a very important way to help adaptors recognise that they have a valuable creative contribution to make.
- The idea of flexing was very well received, in particular the matching of creative style to the innovative/adaptive solution requirements.

Specific information was sought concerning flexing by the use of feedback questionnaires and plenary discussion. The following questions were put and the replies obtained were:

	Yes	Maybe	No
Do you consider flexing to be useful in design?	67%	33%	0%
Has your capacity to flex style increased after training	64%	27%	9%

Impact study

An impact study was carried out to determine what lasting effect the creativity workshops had after three months from attending the workshops. Specific questions were put and the following replies were received.

	% of engineers reporting a retained improvement in this behaviour/ability
More flexible and open minded	75%
More searching for ideas before converging	75%
Being less critical of others' ideas	70%
Increased readiness to re-define problems	45%
Seeking ideas and stimuli from others	45%

It was disappointing (but interesting) to note that, whereas their personal approaches to creativity had improved, only 15% of the participants felt that the improvements were useful in their companies' design activities.

The participants had found the workshops to be stimulating. They reported that time was well spent describing general principles of creativity and in stressing the mental discipline that is required in CPS. The teaching programme was seen to be different from other creativity courses they had attended in that it dealt with engineering problems and the application of the methods to real working situations. It was not an 'away day' kind of experience that might be fun but of little practical use.

4.2 University postgraduates.

Teaching workshops have been held for second and third year research postgraduate research engineers who are taking an Engineering Doctorate programme. Their research is industry based and the work undertaken is similar to that of engineers carrying out advanced work in industry. Also, the teaching workshops have been given to MSc students in Mechanical Engineering Design and to final year undergraduate students who are taking an advanced multi-disciplinary design option (electrical, mechanical and civil engineers). The teaching workshops have been run over a four year period, approximately 45 postgraduates and 35 final year undergraduates have participated in the programmes.

Particular feedback from the sessions has yielded the following responses.

- The concept of innovative and adaptive creative style are received well and clarify to many the vague ideas of the attributes they recognised in others (this should lead to better working relationships in the future).
- The realisation that people with an adaptive style can make a significant creative contribution is seen as a major revelation. Adaptors clearly gained esteem.
- The creative problem solving process was received well when presented in an engineering context, but not otherwise. In the case of undergraduates, applications were often required in their 'home' discipline.
- CPS tools were received best when presented in a manner that appears directly relevant to engineering practice.
- Flexing is readily accepted as a useful ability. Innovators find it relatively easy to use adaptive style techniques. Adaptors will use brainstorming if the 'no judgmental comments' rule is strictly enforced. *They are very effective during the third wave of ideas which produces valuable material.*
- Brainstorming produces an interesting response. Everyone has heard of it but very few have practised the method correctly. Many are surprised by the strict emphasis on 'no comments' as the ideas are generated. After the sessions, remarks are usually heard about how useful it has been compared to their past experience when brainstorming has been practised without suspended judgement. Osborn recommended brainstorming as a method by which individuals can develop the ability to suspend judgement when generating ideas and this has been borne out.

Follow up studies to see how the ideas have been used in design work has revealed that brainstorming and morphological analysis have been adopted. Working alone, individuals have adopted divergent-convergent thinking skills and used suspended judgement.

4.3 Creativity assessment.

The subject of creativity assessment is complex and a number of different measures have been proposed in the literature, reference 13 reviews 41 methods of creativity assessment. Different instruments have been developed to assess different aspects of personality, e.g. the Torrance Test of Creative Thinking, Myers-Briggs Type Indicator and the Basadour Simplex®, and there are correlations between different instruments [14]. Many of the methods of creativity assessment rely on the completion of self-assessment questionnaires. Quantitative measures such as the number of patents claimed, or the economic benefits gained, depend on other factors in addition to the person: the product, environment, economic resources and opportunity. Therefore, when assessing individual personality and behaviour one has to rely to a great extent on self assessment using focussed questions to individuals. The workshops described in this paper aim to improve particular aspects of creative behaviour in people therefore the impact studies carried out focussed on those aspects, as discussed in 4.2 and 4.3 above. Student behaviour was also observed in class. The value of the workshops may be assessed to some extent in this way.

5. Conclusions.

Creativity enhancement teaching workshops have been developed based on creative problem solving principles. The workshops have been used with engineers in industry and with senior students in university. The workshops have been received well, participants have warmed to the principles and methods and feedback has revealed that beneficial creative behaviours have been retained.

The most significant aspect of the workshops is the use of creative style and flexing. Creative style enables innovators and adaptors to appreciate each others' qualities which makes for better working relationships. Also, adaptors realise that they have a valuable contribution to make and are encouraged to think creatively. The teaching programme emphasises the importance of both styles and encourages flexing between styles according to solution requirements. Choosing appropriate creativity tools, e.g. an adaptive ideas generation session when improvements to a product are required, is an important part of effective creative design.

References.

1. Wallas, G. (1926) *The art of thought.* New York: Franklin Watts.
2. Osborn, A.F. (1953) *Applied Imagination.* New York: Charles Scribner's Sons.
3. Gordon, W. J. J. (1961). *Synectics.* New York: Macmillan.
4. De Bono, E. (1974) *Eureka: An illustrated history of inventions from the wheel to the computer.* New York: Holt Rineheart and Winston.
5. Isaksen, S.G., Murdock,M.C.,Firestein,R.L. and Treffinger,D.J. (eds) (1993) *The emergence of a discipline (vols 1 & 2).* Norwood, N.J: Ablex
6. Parnes, S.J., Noller, R.B. and Biondi,A.M. (1977) *Guide to creative thinking.* New York: Charles Scribner's Sons.
7. Rickards (1988) *Creativity at work.* Brokkfield, VT: Gower Publishing.
8. Kirton, M.J. (1989) *Adaptors and Innovators: Styles of creativity and problem solving.* New York: Routledge.
9. Lordan, M (1998) *Methods of enhancing creativity for engineering design.* PhD Thesis, UMIST, U.K.
10 Isaksen, S.G.), Dorval,K.J and Treffinger,D.J (1994) *Toolbox for creative problem solving.* Creative Problem Solving Group - Buffalo, Williamsville, NY 14221
11 Pahl G and Beitz I (1996) *Engineering Design a Systematic Approach.* Springer
12 Pugh, S. *Total design.* (1991) Addison-Wesley
13 Puccio, G.J. and Murdoch, M.C. (1999) Creative Education Foundation, Buffalo, N.Y.
14 Isaksen, S.G. and Puccio, G.J. (1988) *Adaption-Innovation and the Torrance Tests of Creative Thinking: The level and style issue revisited.* Psychological reports, 63, 659-670.

Dr G.Thompson C.Eng. F.I.Mech.E.
Department of Mechanical Engineering, UMIST, PO Box 88, Manchester, M60 1QD, UK, Tel: (+44)161 200 3743, Fax: (+44)161 200 3848,
E-mail: graham.thompson@umist.ac.uk, URL http://www.me.umist.ac.uk/
Alex F. Osborn Visiting Professor, Buffalo State College, NY

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

CLASSIFICATION AND EFFECTIVENESS OF DIFFERENT CREATIVE METHODS IN DESIGN PROBLEMS

E Mulet and R Vidal

Keywords: Problem solving techniques, creativity methods, solutions finding, effectiveness.

1 Introduction

This paper focuses on the study of creative methods, with the aim of evaluating their effectiveness. Many of the most important creative methods, also known as black box or intuitive methods, appeared in the 50s. From then on, not many modifications were seen until the improvement in computer performance, which allowed for the implementation of other kinds of problem solving techniques, such as axiomatic design. In addition, there is an increasingly wide pool of techniques available, although designers have only limited criteria to help them decide which ones to use.

In line with this, our study contributes to improving the selection criteria for a particular creative method. The objectives of this article are: to present a classification of the most important creative methods; to propose new contributions for the development of evaluation methods for measuring the effectiveness of creative methods and to obtain results that show the effectiveness of these methods.

First of all, the methodology applied is described. This methodology consists of a classification of creative methods, a description of the creative method analysed in this study and a proposal for effectiveness evaluation. Secondly, a description of several experiments carried out to investigate the method's effectiveness is provided. And finally, the results obtained are presented and discussed.

2 Classification of creative methods

In this study, a classification of creative methods is presented. We are interested in analysing the effectiveness of creative methods. Therefore, we considered the operational mechanism of the method to be the most appropriate criterion to define this classification. In this vein, the classification includes the categories defined by Roozenburg [1], namely association and creative confrontation methods. The other categories were set according to previous classifications [2], in which several criteria proved to define the groups of methods. Table 1 shows the categories of this classification.

Association methods are those which encourage relationships to be spontaneously established for the generation of ideas. Creative confrontation methods consist of an enforcement of the establishment of a relationship that did not previously exist. When the method consists of a rearrangement of the initial information in order to encourage new ideas, we are talking about the initial information rearrangement category. Assumption checking methods are those which question every aspect of the problem that is taken for granted. Finally, exhaustive problem exploration consists of searching for solutions in a series of defined aspects.

Even though there is a general operational mechanism in a creative method, if a deeper level of detail is analysed, it is possible to observe the ways in which an idea is set during the generating alternatives phase. From now on, these "ways" will be called idea-finding fonts. They have been determined by the observation and simplification of the possible ways in which an idea is set during a creative process. Furthermore, these idea-finding fonts are a way of categorising the method results. In addition, a relationship between these fonts and general operational mechanisms of methods was observed, which can be seen in Table 1.

Spontaneous idea-finding fonts are those in which an idea appears without a clear, intentional purpose. When one idea follows on from another, it is known as an evolution font. Sometimes an idea is the result of joining two or more previous ideas, which is known as a synthesis font. Opposite fonts are those in which an idea appears as the result of denying or changing the direction of a previous idea. And finally, when an idea comes as an intentional comparison to something that seems to be a solution, it is called a comparative font.

Table 1. Classification of creative methods

Idea-finding fonts	Category of methods	Methods
Spontaneous	Association	Brainstorming, brainwriting pool, 6-3-5, brainmapping, reversal brainstorming, brainsketching, Gordon method, free association, mindmapping, Lotus Blossom, Delfos
Comparative	Creative confrontation	Synetics, forced analogy and random stimulus, ideatoons, bionics, forced relationship, inspired approach, transformation
Evolution and Synthesis	Initial information rearrangement	Problem division, problem inversion, methodical doubt
	Exhaustive exploration of the problem	Check lists, attribute listing
Opposite	Assumption checking	Why?, assumption smashing

This study focuses on brainstorming methods, but other groups of methods may be analysed in the future. In this paper, we will show how many different idea-finding fonts take place together when applying an association method such as brainstorming.

3 Brainstorming methods

The creative method used in this study is a kind of brainstorming method which is slightly different from the traditional one [1]. One difference is that the idea given to the rest of the people in the group is written down. In order to simplify the analysis, the number of group members is reduced to five. Finally, the evaluation step was ignored in this study because we are only interested in evaluating the solutions finding process. In this variant of brainstorming, when a member comes up with an idea, he/she has to tell it to the rest. Immediately, he/she writes it down on a card and places it in the centre of the table. Everyone has to listen to the idea and add something new to develop it. When there are no more comments, someone has to start off with a different idea and thus, the cycle is repeated. Reading the cards will help to find new ideas for a previous or a new solution.

Three types of brainstorming methods were analysed. These options consisted of providing different supports to generate alternatives. The first type is the one described above, which we will call B1 or verbal brainstorming. In the second case, participants have to write, but also to draw each solution on a card or piece of paper; we call this B2 or visual brainstorming. It is a type of brainstorming based on traditional brainstorming with added sketches [3]. And finally, we have defined a new type in which participants have to write and physically construct the solution with pieces and tools. There is no history of brainstorming analysis with characteristics of this type, which we will call B3, or physical brainstorming. Through these types, it is possible to analyse the influence of different material supports on the effectiveness of a brainstorming method. Figure 1 shows an implementation of B3 method.

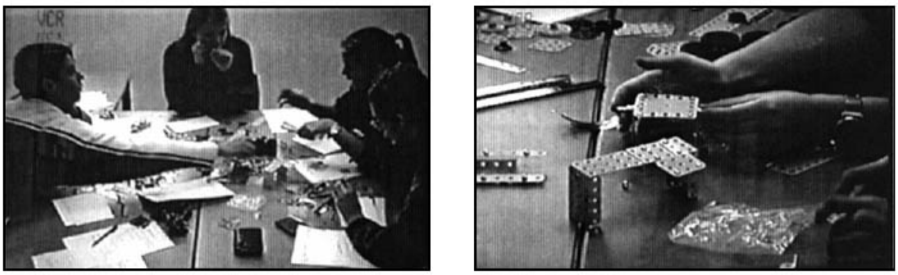


Figure 1. Brainstorming with pieces and tools

4 Evaluation techniques

The quantity and the quality of ideas generated through brainstorming methods have been analysed in previous research work [6]. Here, the purpose consists in evaluating the effectiveness of creative methods relating them to the way in which the idea has been obtained (idea-finding font). The proper use of the mentioned relationship is then used as a new evaluation technique. With these results, it is possible to expand on the works of Van der Lught [3], who has studied the performance of variants of brainstorming with different visual aids. The methodology used is the analysis of the protocol.

The protocols are studied in order to identify the **ideas or sub-ideas**, and the global ideas. A **global idea** is a group of ideas that refer to the same solution. An idea or sub-idea is each new contribution to the solution of the design problem. It is equivalent to a move in Goldschmidt's work [4], [5]. Three parts of the evaluation method applied can be differentiated.

4.1 Validity levels

A way to obtain a measure of the effectiveness of the method is proposed in this study. It was decided that analysing the validity of each idea in relation to the specifications is an appropriate way to evaluate the solutions finding process. Three levels are defined to evaluate validity: **valid idea**, if the idea contributes to the solution of the problem; **rejected idea**, if the idea is related to the problem specifications but it is not a viable solution or it does not match the specifications; and the third level, **non-idea,** or an idea not related to the problem.

The number of each kind of idea is determined. The composition or proportion of each kind of idea is also an interesting way of analysing differences between methods.

4.2 Process variables

A second part of the evaluation is the determination of several process indicators. First of all, a linkograph [3] [4] [5] is constructed that represents the relationship between the global ideas. From this linkograph the link density [3], which is equivalent to the link index [4] [5] is determined. The **link density** is the ratio between the number of relations and the number of ideas. With this ratio, we will compare our results with previous research carried out [3] [4] [5].

We defined an index that is the **idea-global idea ratio** (I/IG), to measure the proportion between the number of ideas and the number of global ideas. Our purpose is to determine whether there is a relationship between I/IG ratio and the effectiveness of the method and what relationship this is. The idea of measuring this ratio came from the observation of the protocols, from which we observed important differences on this question.

4.3 Idea-finding fonts determination

The last part of the evaluation is the study of the use of idea-finding fonts in a brainstorming method. This is done in order to analyse the presence of idea-finding fonts characteristic of other groups of creative methods, with the purpose of analysing the relationship between process effectiveness and idea-finding fonts.

The protocols are then studied and an idea-finding font is assigned to each idea. The assignment is made by analysing the content of the idea and determining whether or not it has a relationship with a previous idea and the kind of relationship it is.

5 Description of the experiment

In order to meet the objectives, certain experiments were ruled out. These experiments consisted of generating alternatives for a design problem following the steps of the brainstorming method described. Each experiment group was made up of five first year Industrial Design Engineering students from the Universitat Jaume I in Spain. These students were already familiar with design methods.

The problem to be solved was a **functional design problem**. It consisted of the design of a drawing table that took up minimum space when it was not in use. Other specifications were that the table had different inclinations, that the cost was as low as possible, and several other essential specifications for a drawing table.

A total of twelve experiments, four of each type was carried out. During the experiment, all the members had to "think aloud". Each experiment was videotaped in order to analyse the protocol. The organisation of the experiments was as follows: first of all, an agreement was reached in which the students agreed to follow all the instructions; secondly, all the method instructions were explained. Finally the design problem to be solved was defined. At this moment the experiment started and lasted one hour. Six experiments were done simultaneously; and a second turn was done immediately, in order to avoid the transfer of information between students.

Up to now, only one protocol of each type has been analysed and a statistical analysis will be carried out in order to validate the results and get new findings.

6 Results and discussion

During the experiments we observed how the participants motivation and the quality of ideas were better in B2 than in B1, and much better in B3 than in the other two. The following results confirm these impressions quantitatively.

6.1 Analysis of the effectiveness

From the analysis of a protocol of each type of brainstorming, each idea and global idea is identified. Following that, the adaptation of the ideas to the specifications is studied. Table 2 shows these results.

Table 2. Effectiveness results of the analysis

BRAINSTORMING TYPE	**B1**	**B2**	**B3**
NUMBER OF IDEAS	102	75	110
NUMBER OF VALID IDEAS	16	57	86
% VALID IDEAS	15.7%	76.0%	78.2%
% REJECTED IDEAS	6.9%	14.7%	18.2%
% IDEAS NOT RELATED TO PROBLEM	77.5%	9.3%	3.6%

From the results, it can be deduced that applying a brainstorming method with the aid of pieces to construct the solutions increases its effectiveness. This follows from the number of valid ideas in each experiment: while in type 3 there are 86 valid ideas, in type 2 there are 57 and in type 1 only 16. The percentage of valid ideas is quite high in B2 and B3, but in B1 less than 16% of the ideas are valid. It is particularly interesting that 77.5% of the ideas in B1 have nothing to do with the problem specifications. We think that this result is especially high due to the fact that an hour is a long time to look for solutions by verbal or sentential brainstorming.

In conclusion, the most effective method is brainstorming type 3, followed by type 2 and finally, B1. It is important to take into account that the design problem in these experiments is a functional problem. These results could be very different if a design problem with a formal component had to be solved.

6.2 Process variables results

Other indicators studied are the link density (LD) [4][5] between the global ideas and the idea-global idea ratio (I/IG ratio). Table 3 presents these results.

Table 3. Design process measures.

BRAINSTORMING TYPE	B1	B2	B3
NUMBER OF IDEAS	102	75	110
NUMBER OF GLOBAL IDEAS	55	24	16
RATIO IDEAS/ GLOBAL IDEAS	1.9	3.1	6.9
LD IN GLOBAL IDEAS	0.61	0.71	0.94

Figure 2 graphically shows the differences in the number of global ideas and in the quantity of ideas per global idea for each type of brainstorming. Each symbol illustrates how many global ideas have the number of sub-ideas that are indicated in the horizontal axis. In this manner, one global idea is represented by the symbol O; the symbol ɸ means two global ideas and an additional line jutting out the circle means one more global idea.

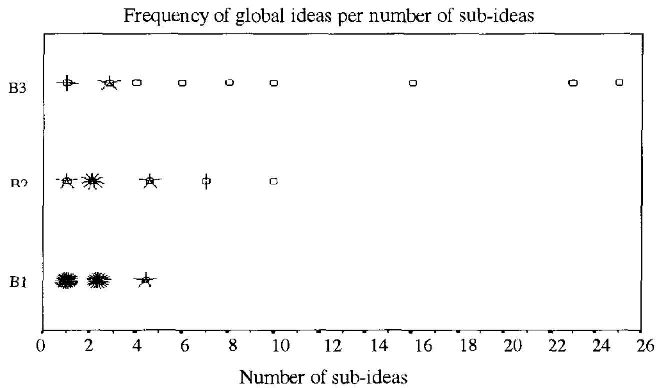


Figure 2. Differences between types of brainstorming

The number of global ideas is very high in B1 in comparison with B2 and B3. The I/IG ratio is very low in B1. In B2 this ratio increases by 60% and in B3 this ratio is much higher than in the other types of brainstorming. The link density between global ideas is also higher in B3. The lowest link density corresponds to B1 again. This happens because in B3 the ideas are qualitatively on a deeper level of detail than in B1. This causes a higher I/GI ratio and also a higher number of valid ideas. The LD is lower in B1 due to the major dispersion of the ideas obtained in comparison with B2 and B3. B2 is placed in an intermediate position.

These results coincide with Goldschmidt's [4][5], in which the greater the link index, the more productive the design process. In B1 the number of ideas is high, but there is a low relationship between them, and there is not a good development of the ideas in comparison with B2 and especially with B3.

6.3 Idea-finding fonts results

Additionally, an analysis of the idea-finding font was carried out. To achieve this, the origin of each idea is analysed, and it is assigned it to one of the defined fonts. With these measures, it is possible to analyse the influence of the idea-finding font on the effectiveness of the solutions finding stage in a design process. The results of the analysis of the idea-finding fonts can be seen in Figure 3. The codes to express each of the fonts and the percentages in each experiment are found in Table 4. Table 5 illustrates data extracted from one experiment in which the type of idea finding font has been analysed. When a sub-idea is analysed, the font is labelled as 5E4, which means that sub-idea number 5 comes as the evolution (E) of sub-idea number 4.

Table 4. Idea-finding fonts codification and percentages

CODE	FONT	B1	B2	B3
S	Spontaneous	35.3 %	9.3 %	11.8 %
E	Evolution	58.8 %	66.7 %	78.2 %
J	Synthesis	3.9 %	4.0 %	0.0 %
O	Opposite	0.0 %	8.0 %	4.5 %
C	Comparative	2.0 %	12.0 %	5.5 %

Table 5. determination of some idea finding fonts.

Nº of sub-idea	Student Code	Protocol	Idea finding font
5	A2	In a part of the desk one take a board with the required size and then it is positioned over the desk.	5E4
6	A2	Using the wood strips or the bars, the selected bending angle is adjusted. When one do not use the board it is removed.	6E5
7	A3	The one mentioned by A2. I would prefer to assembly it to the desk allowing its extraction in the same way as the keyboard in a computer desk. This would allow hiding it.	7C4
8	A1	Installing a guide system?	8E7
9	A1	Making it a fold-away desk?	9S

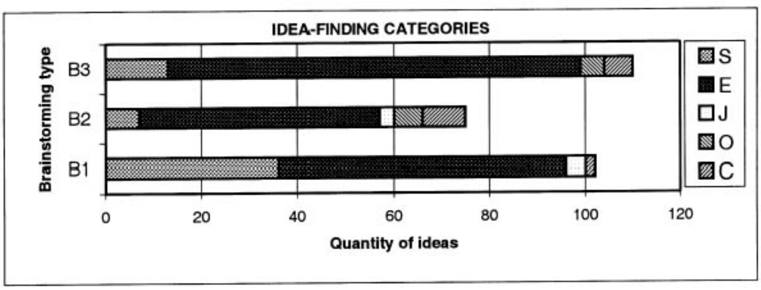


Figure 3. Composition of idea-finding fonts

Analysing these results, it is possible to study the relationship among the effectiveness and the composition of idea finding fonts. We observe that in all the experiments, over half of the total number of ideas was arrived at by evolution. In B2 and B3 the percentage is very high. The order in effectiveness of the experiments is the same as the order in percentage of ideas obtained by evolution.

The quantity of spontaneous ideas in B1 is higher than in the rest. In this case, B2 is the brainstorming type with the lowest number of spontaneous ideas. Thus it is not possible to obtain a relationship between the percentage of spontaneous ideas and effectiveness from the present data. However it seems that the presence of a high number of spontaneous ideas is negative in terms of effectiveness. A third observation from the results is that in B1 there are no ideas obtained by opposition, and the number of ideas obtained by comparison is very low.

The origin of all the ideas was identified with the idea-finding fonts defined. Therefore, the analysis of the protocols verifies our fonts definition; we have not found an idea which does not come from one of these fonts.

Finally, these results show that a spontaneous association method does not work exclusively with spontaneous relationships. On the other hand, the spontaneous association is not the most effective method, as shown in Figure 3.

7 Conclusions

In a functional design problem, variants of brainstorming methods that use physical aids as pieces improve the effectiveness of the method. This effectiveness is also related to the idea finding font. In this vein, a high percentage of ideas obtained by the evolution of previous ideas increases effectiveness.

The results found coincide with the initial classification. The evaluation techniques are coherent with the effectiveness results and answer the initial impressions, so the evaluation method has been successful. It is also remarkable that brainstorming with materials and tools met with great approval from the participants of the experiments. This is an interesting point to take into account when applying or teaching brainstorming.

References

[1] Roozenburg N.F.M., and Eekels J., "Product design; fundamentals and methods", Wiley, Delft,1995.

[2] Zusman and Zlotin. "Overview of creative methods" TRIZ journal. July 1999.

[3] Van der Lugt, R. "Developing a graphic tool for creative problem solving in design groups" Design Studies, Vol 21, 5, 2000, pp. 506-522.

[4] Goldschmidt, G. "The designer as a team of one", En Cross N., Christiaans H., & Dorst K. (Eds) Analysing design activity, Wiley,Chichester, UK:, 1996

[5] Goldschmidt, G., "Criteria for design evaluation: a process-oriented paradigm", Evaluating and predicting design performance, Kalay Y.E., Wiley, Chichester,1992, pp.67-79

[6] Ziegler, R.; Diehl, M.; Ziljstra, G. "Idea production in nominal and virtual groups", Group Processes and Intergroup Relations. Vol 3, 2, 2000, pp.141-158.

Elena Mulet
University Jaume I, Department of Technology, Address: Campus de Riu Sec 12071 Castellon, Spain, Tel: +34 964718188, Fax: +34 964728106, E-mail: emulet@tec.uji.es,
URL: http://www.tec.uji.es/Residuos/residuo.html

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

INNOVATION IN THE TENSION OF CHANGE AND REUSE

M Gerst, C M Eckert, P J Clarkson, and U Lindemann

Keywords: Drivers of innovation, Psychology of creativity, Innovation methods

1 Introduction

Most new products in engineering are designed by modification from existing products for sound economical reasons. Yet most discussions of innovation in the literature take a totally fresh look at the design problem. This paper investigates the vital role of core-requirements in identifying suitable areas of innovation and analyses how innovation can be integrated with design by modification. It proposes a combination in a two stage approach where techniques are used to identify a set of starting-requirements and to map out the solution space. An analysis of the possible changes to the existing product and their impact is used to reduce the number of possible solutions and to carry out a trade-off between fulfilling requirements and the impact the changes will have on the product.

2 Innovation

All companies must innovate to maintain competitive advantage. In practise the majority of all new products are designed by modification from existing products, reusing product properties, functionality, solution principles, parts and components. This paper addresses the fundamental question that all companies must address: "How can we know what can be changed and what should be reused?"

2.1 Traditional views on innovation

Common models of innovation processes distinguish between "technology push", where new technical developments allow the generation of a new product for which a market needs to found, and "market pull", where companies tend to understand innovation as a new or reinforced market-oriented way of thinking. Methods like Quality Function Deployment (QFD) or Target Costing support this way of thinking, by translating customer requirements into technical product characteristics with appropriate parameter values and associated allowable costs. Innovation is driven by the aim to increase customer satisfactions by meeting requirements better and to maximise profit through meeting customer requirements in a more economical way. These methods largely leave the considerations of technical realisation aside. They create a possibility for technical innovation while making the risk hard to estimate.

Established creativity methods, such as brain storming or method 6-3-5 [1] are aimed at the development of new product ideas. These methods aim explicitly to free the designer from the bias of past designs. They enable designers to generate many new ideas, which later need to be carefully evaluated. In this situation, designers are rarely short of ideas, but many of them are not appropriate for realisation later in the design process. A successful idea is one that leads to a working prototype and ultimately a production model which performs well in the market place.

In industry development rarely aims at totally new products, more often focusing on the improvement of existing products to meet new customer requirements. Rather then designing a new product, designers mostly modify some components or subsystems of an existing product while leaving others intact.

2.2 Stable balance between requirements and product properties

A product that has been successfully introduced into the market can be seen as a stabile balance between its properties and its requirements. If new core-requirements for the system arise, all requirements have to be redefined, which in turn leads to changes to the product and the system is thrown out of balance (see Figure 1). In complex products designers rarely understand all the implications of a change to the product:

- *Propagation of changes to product.* A change to one part of the product can affect many other parts. The cumulative effect of several changes is especially hard to predict [2], and hence the likely costs of change are in turn hard to predict. In the extreme, change may give rise to further money- and time-consuming changes, or a decrease in the quality of the product [3].

- *Effects on buyers behaviour.* The market can find radical changes (innovation) difficult and might show an ambivalent reaction. A previously used concept has a higher credibility for the consumers, because they feel that the idea is proven. This view is justifiable as new innovations are often brought to the market without being fully tested and customers are inconvenienced when they fail, are recalled or changed.

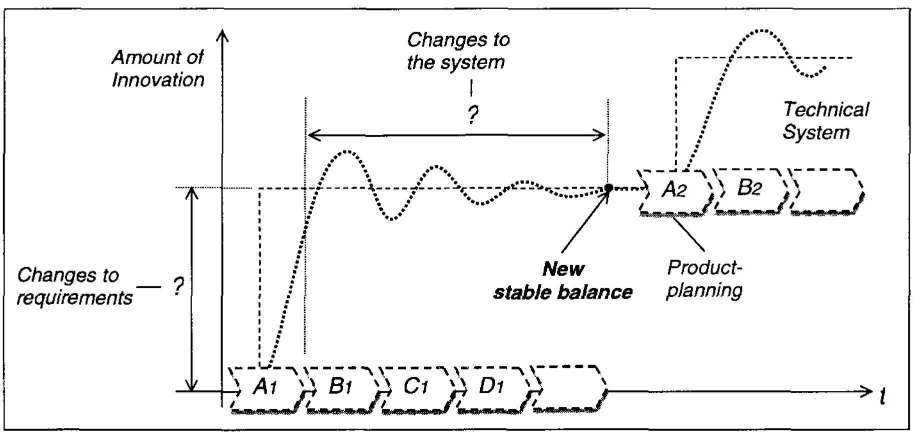


Figure 1. Stable balance of states

Innovation is found in the constant tension between change and reuse. Why should something that works be changed? One reason could be in reaction to another change. Designers must be aware of the implication of a possible change or a possible reuse. The reuse of existing solutions, components, manufacturing techniques and equipment makes sound economical sense. It reduces design time, saves on development and testing costs, enables companies to reuse specific tooling and allows bulk ordering. Change might be necessary to maintain a competitive edge, not just for this product, but as part of an innovation strategy.

2.3 Our study

This paper reports on ongoing collaborative work between Institute of Product Development at the Technische Universität München (TUM) and the Engineering Design Centre (EDC) at the University of Cambridge. Both groups have longstanding experience in innovation research [4, 5] and are researching change and customisation processes [2, 3, 6]. This paper draws in particular on process modelling and the findings of 20 interviews with engineers and engineering managers in an automotive company.

3 Innovation as a response to changed requirements

Product innovation is driven by core-requirements that define the desired behaviour of the new product. In our case study the automotive company had developed 6 core-requirements for a new car through market research and internal discussion specifying for example the new car should be "more sportive" and have "better handling". This wish list was broken down into technical core-requirements such increased acceleration by x %.

Conventional design methods stress the importance of a full requirement analysis, but without having clearly specified core-requirements the designers are in danger of loosing sight of their goal and can't prioritise their requirement [7]. Borrowing the term from marketing, we define *product strategy* as the core-requirements and important technical and commercial requirements. In response to the product strategy designers can develop an *innovation strategy*, which identifies those product innovations that are desirable; and can re-evaluate other requirements in the light of the innovation strategy.

3.1 Identification of a set of starting-requirements

A design process must begin with establishing a product strategy and is driven by it. Most fundamentally core-requirements aim to increase customer benefit by offering a more attractive product or to increase profits, for example by decreasing manufacturing costs or building up a new skill base in the company.

Core-requirements are those that give a company an edge over its competitors and their fulfilment is vital to the company. Cost savings are core-requirements in most development projects, however cost increases can be justified by customer needs. Other core-requirements can be derived from the main reasons why a customer would buy the product (*Purchase Promotion Factors*). Further requirements can arise from the needs of a particular organisation, such as adequate use of internal resources.

Core-customer requirements can be found through market or customer analysis as well as internal discussions. Customers rarely know the technical requirements that would encourage them to buy the product and don't understand long-term market and technological trends. Some companies use a "Product inventory" to look at the range and history of their products, identifying good and bad points, in order to find the factors that define the brand identity. Then can then use this information to evolve the brand in an appropriate way.

The product strategy includes not only those decisive requirements that make the customer buy a particular product, but also basic-requirements for the successful and safe use of the product. It also includes constraints on the core-requirements, such as limitations on the applied solution principles that stop the designers from scaling up existing designs without major changes to the fundamental design of the product.

The distinction between core-requirements, basic-requirements and constraints are important in the development of an innovation strategy for a particular product and across the entire company (see Section 4). Typical requirements lists distinguish between compulsory "must have" requirements and optional "nice to have" requirements. However, it is often the optional requirements that form the starting point for the product strategy.

From a customer's point of view the flip side of the purchase promotion factors are the *purchase rejection factors*. These stop the customer from buying a product. Hence such factors can be used to define the basic-requirements for the product. Without knowledge of the rejection factors, the product may not sell. However, they do not give the product a competitive advantage. Industrialists tell us that it is often easier to find out what customers do not want, than to find out what they really want and what makes them excited about the product.

Further restrictions leading to basic-requirements can originate from the company itself, competitors or suppliers. They may arise because of technical, economical, legal, social or ecological reasons. Frequent sources of restrictions are companies' or suppliers' design or production processes, for example time pressure, cost pressure or standardisation as well as competitors patents.

3.2 Classification of requirements

To identify a set of starting-requirements designers need to be able to prioritise requirements. This can be achieved, for example, through use of the following classification of requirements. Kano [8] differentiates between excitement-requirements, performance-requirements and basic-requirements, each reflecting different levels of customer satisfaction. Excitement-requirements lead to high customer satisfaction, since they are not expected and satisfied even when they are not completely met. Basic-requirements do not lead to high customer satisfaction as they are strongly expected (even though not explicitly) and their absence causes dissatisfaction. Performance-requirements range between excitement and basic-requirements and can best be expressed by the customer, reflecting the customers' expectation of the product.

To prioritise requirements we classify them by the way they affect solution generation (see Figure 2). In an extension to the Kano model, requirements can be classified as:

- *vital requirements* including excitement-requirements, which capture those features likely to gain unexpected customer satisfaction, identity-requirements, which maintain the identity of the brand, and core-requirements;
- *important requirements* are the Kano performance-requirements, which are expected by the customer;
- *boundary requirements* are basic-requirements and constraints, which need to be fulfilled to have a viable product.

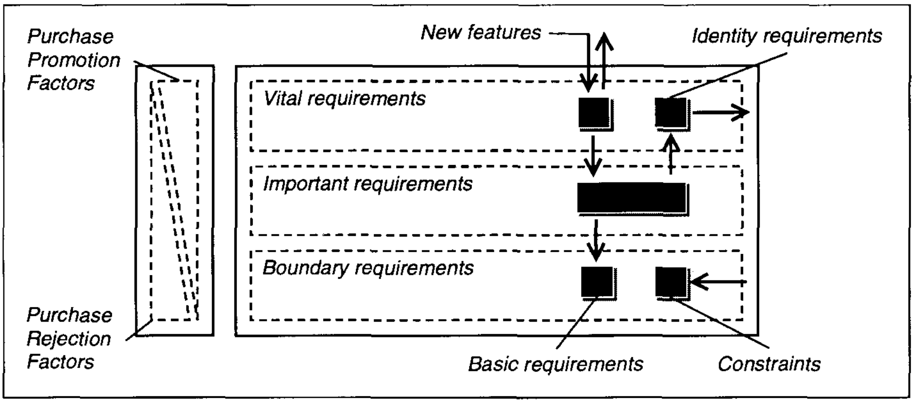


Figure 2. Classification of requirements (according to Kano-Model)

As time passes requirements are increasingly taken for granted by the customers. For example, at first safety belts were excitement-requirements, then they become performance-requirements for all good cars and now they are basic-requirements, without which nobody would buy a car. Performance-requirements that have been maintained and updated over many years in company become identity-requirements.

4 Handling the risk of innovation

An important question at the beginning of most design projects is to what extent innovation becomes necessary and how it should be applied. Figure 3 describes a procedure that can help designers to build up an innovation strategy by defining parts, components or product functions whose change becomes necessary and vital for further product success.

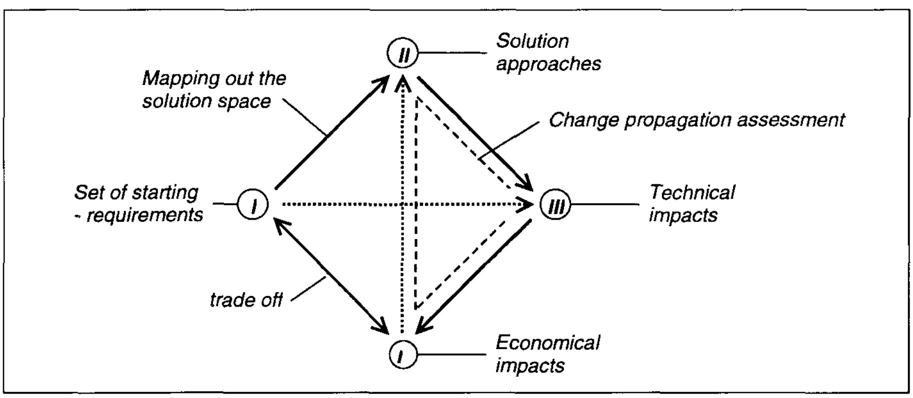


Figure 3. Requirement change cycle

The selection of suitable areas of innovation in a product depends on the risk they carry. The risk must be assessed with regard to the effect of an innovation on the entire solution, the chance of changes propagating through the system, the effects of the changes on the design and manufacturing process and the effects on the customers' purchasing behaviour. Even though risks can be taken at the level of single components or parts, the total risk of the development project must remain manageable and commensurate with the effects of a possible failure.. In the following section we describe a procedure showing how, starting from a set of requirements, necessary innovations can be identified and possible solutions evaluated according to their technical and economical impact (see Figure 3).

4.1 Mapping out the solution space

To work out an innovation strategy it is important to map out the solution space that is available for the fulfilment of the core-requirements. From the core-requirements key parameters can be derived. This analysis establishes the degrees of freedom in the solution space and identifies a number of solution approaches [7]. A systematic analysis of the solution space is necessary since designers tend to be satisfied settling for the first solution that they find and are subject to fixation on old solutions [9, 10] even if they consciously try to innovate. Designers cope with the complexity of design by being example driven in their thinking style and discourse, which leads to more conservative designs [11]. An example from our case study is given in Figure 4.

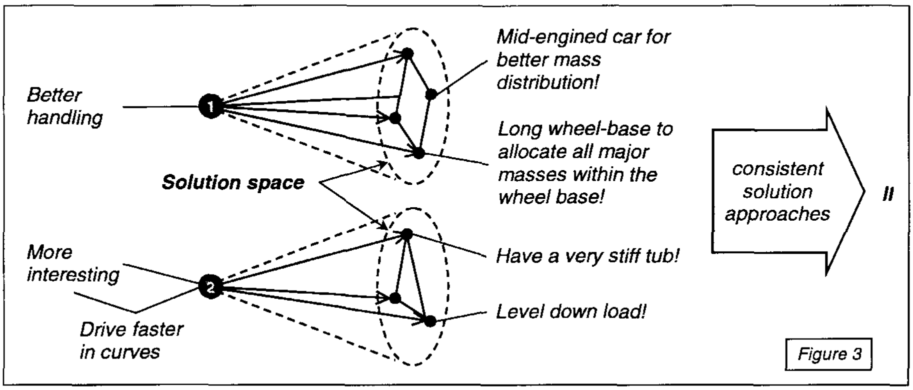


Figure 4. Example for mapping out the solution space

For each core-requirement designers need to establish its implications for the entire product. They have to analyse the relations between requirements and product characteristics. This includes the relationships between requirements and product characteristics that can be directly influenced, such as material, number of parts, dimensions, as well as the relationship of the product characteristics with each other. This analysis requires a vision of an "optimal system" (e.g. according to the laws of physics) and knowledge of the trade-offs that will need to be made between parts of the product. Only designers with great experience can make these assessments and have to be guided by their instincts, because the understanding of the connections is often tacit. While various design methods, such as QFD, can contribute to the understanding of these dependencies there are limitations to their applicability [12]. Therefore it is more important to bring teams of experts together to map out the solution space.

Once the space of possible solutions has been found, individual solution approaches can be selected. This has to be an iterative and creative task, since conflicting requirements and product characteristics have to be identified. Solution characteristics must be weighed against each other, against basic-requirements and constraints to assess the adequacy of the solution approach. In the example given in Figure 4, it turned out that, because of the location of the engine, passengers had to be seated further forward and because of requiring a stiff tub, the frame had to be strengthened. As a consequence it became difficult to get into the car with a conventional door - a basic-requirement (purchase rejection factor). Identifying and solving these conflicts can lead to new solution approaches and innovation. In this example the car was designed with a novel door swinging upwards.

4.2 Evaluation of possible solutions

Once a solution approach has been identified, the designers must select a suitable starting design and assess the changes that are required to this design. A careful assessment of the impact of these changes on other parts of the product gives a greater understanding of the problems involved and allows a rating of the possible solutions. This rating can also include a long-term view about planning updates to the product. The assessment of the implications of change is a two-stage process:

- *Propagation of change to other parts of the product (II – III):*
  Clarkson *et al.* [6] assess the risk, in terms of likelihood and impact, of a change propagating from one system to another based on an a priori experience value captured from experts. Such an approach gives a first indication of likely change propagation. However, change propagation predication is tricky since the likelihood that change will propagate from one system to the next depends critically on the tolerance margins of key parameters linking these systems [2]. Kleedörfer proposes the estimation of change propagation by team discussion [3].

- *Change impacts to economical impacts (III – IV):*
  Typically only a small number of cost factors are considered to assess the impact of a change, while the consequences of a change in terms of design time, flexibility of the solution and quality of the product are not considered. A balanced analysis to select a solution approach should include all factors relevant to the business and not just cost [3].

4.3 Trade-off between requirements and impacts

Every new design solution will inevitably require a trade-off between fulfilling all requirements and the impact the changes will have on the product. Designers have two choices. They can reduce their requirements, assessing whether the violated requirements are vital, for example bringing out a less attractive car with a comfortable but conventional door. Alternatively, they can develop a new solution that does fulfil both vital and basic-requirements. However, the time and resources to do so are limited. In consequence most products are compromises.

5 Conclusions

In industry most new products are developed by modifying existing ones. Innovation is therefore typically a change to a product. Before the design of a new product can begin designers need to carefully assess the requirements for a new solution. Innovation must be targeted where requirements make it necessary, keeping successful solution aspects unless they would have to be changed later as part of a long-term innovation strategy. We propose a two-stage strategy involving a mapping out of the possible solution space based on the requirements and a selection of solution approaches based on an assessment of the impact of the change on other parts of the product, thus assuring the most economical solution.

Acknowledgements

The authors are grateful to the UK Engineering and Physical Sciences Research Council (EPSRC), GKN Westland Helicopters Ltd and Lotus Engineering for their support of this project.

References

[1] Gausemeier, J, Lindemann, U, Reinhart, G, Wiendahl, H-P (2000), "Kooperatives Produktengineering," HNI, Paderborn. (HNI-Verlagsschriftenreihe; Bd. 79)

[2] Eckert, C M, Clarkson, P J and Zanker W (2000), "Customisation in Complex Engineering Domains," University of Cambridge, CUED/C-EDC/TR101.

[3] Lindemann, U and Reichwald, R (1998), "Integriertes Änderungsmanagement," Springer, Berlin.

[4] Blessing, L T M and Yates, I R (1999), "Design and development capabilities of small and medium sized enterprises in the UK: an analysis," in 12th International Conference on Engineering Design, Munich, Germany, 1, pp119 –124.

[5] Gerst, M, Seidemann, H and Kanitz, F (2000), "Kooperatives Produktengineering. Strategische Produkt- und Prozessentwicklung," in: ZWF 95, no. 7-8, pp369-372.

[6] Clarkson, P J, Simons, C S and Eckert, C M (2000), "Change Propagation in the Design of Complex Products," in Engineering Design Conference 2000, Brunel University, UK, 27-29 June, pp563-570.

[7] Lindemann, U, Demers, M, Gerst, M and Wulf, J (2001), "Lösungssuche mit Hilfe von Zielformulierungen," in: Konstruktion, n. 3.

[8] Kano, N, Seraku, N, Takahashi, F, Tsuji, S (1984), "Attractive Quality and Must – be Quality," in: Journal of the Japanese Society for Quality and Control, **2**(14): pp39-48.

[9] Jansson, D G and Smith, S M (1991), "Design Fixation," Design Studies, **12**: pp3-11.

[10] Purcell, A T and Gero, J S (1996), "Design and other types of fixation," Design Studies, **17**: pp363-383.

[11] Eckert C M and Stacey M (2000), "Sources of Inspiration: A Language of Design," Design Studies, **21**(5): pp523-538.

[12] Eiletz, R (1999), "Zielkonfliktmanagement bei der Entwicklung komplexer Systeme – am Beispiel PKW-Entwicklung," Shaker, Aachen.

Dipl.-Ing. Manuel Gerst
Institute for Product Development
Technische Universität München
Boltzmannstr. 15
D-85748 Garching
Phone: 0049 89 289 15153
Fax: 0049 89 286 15144
E-mail: gerst@pe.mw.tum.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

TOWARDS HYBRID METHODS FOR SYNTHESIS

A Chakrabarti

Keywords: human creativity, innovation methods, principles of invention, concept generation.

1 Introduction

Design synthesis research is often seen as having produced a set of competing approaches, falling largely under two seemingly conflicting philosophies. One is the design reuse philosophy which prescribes retrieval and reuse of past designs as much as possible, and suggests adaptation of past designs to the current purpose if complete reuse is not possible. The other is the compositional synthesis philosophy that prescribes development of a design by composing a set of building blocks.

Under each of these philosophies, the approaches available are either designer-led methods, or automated tools. For instance, the function-structures approach [1] is primarily meant to be a designer-led approach for compositional synthesis, while the functional synthesis approach [2] is one of its automated approaches.

Within the design reuse philosophy, similar divisions can be seen within its two major sub-areas: retrieval of designs, and change of a design to suit the current purpose. For instance, design catalogues by [3] is a designer-led approach to retrieval, while retrieval using analogy [4] is one of its automated approaches. Similarly, the support proposed based on a function model [5] can be taken as a designer-led approach to change and repair, while the evolutionary approach to architectural layout design [6] is one of its automated approaches.

The aim of this paper is to address the issue of polarity and specialism in the area of design synthesis research, and identify the scope of blending the current, apparently conflicting and competing approaches into more powerful synthesis schemes that should help make major advances in the capability of the area as a whole.

2 A review of the main approaches to synthesis in design

The main approaches to synthesis – compositional synthesis and design reuse (which includes retrieval and change) – are discussed below, with the aim of identifying their relative significance, and the main issues involved in terms of them providing creative designs that satisfy a given function.

2.1 Compositional synthesis

Compositional synthesis approaches start off with the overall function to be fulfilled by a design, and combine building blocks to fulfil these functions [2, 7, 8]. Usually compositional synthesis is seen as the process of generating compositions of building blocks by combining them from a pre-defined set. However, identification of building blocks itself is part of the broader, creative task of synthesis [9]. In order to be able to synthesise well, one must be able to identify appropriate building blocks; central to this are the issues of what building blocks to use, and how composite or basic they should be [10, 11].

Compositional synthesis has been hailed as having the potential of generating innovative designs, but at the cost of having high potential risks associated with them (as with innovative designs in general) and having a resource-intensive development process. Case based design approaches [12], which are based on retrieval and adaptation (change), are seen to be more resource efficient [13] and less risky [14] than compositional synthesis.

2.2 Design Reuse

Within a design reuse approach [4, 15, 16, 17], a design is retrieved and changed if necessary. The change may be due to (i) the need to generate alternative designs [4, 15], (ii) to modify an existing design to fulfil the current purpose [16], or (iii) to develop an optimal design [17, 18].

Retrieval

Retrieval is the first step in the design reuse approach. Often the central issue is how to identify the most appropriate cases for the current purpose. When purpose can be expressed in a quantitative way, the main task has been to identify whether a retrieved design is capable of providing the function [19, 20]; change in parametric values of the components of the retrieved designs are performed to test this. In others, where product functions are qualitative, analogy is often used as a means of retrieval, which can be at several levels such as functional, behavioural and structural [4].

In general, retrieval of complete designs is likely to lead to the least innovative designs, but has less cost associated with the development process. The more different the source and target domains are, the more innovative and risky the designs, and the more resource-intensive their development process are likely to be.

Change

One way of changing a design is to identify a set of rules the application of which creates different but valid designs. These can then be subjected to test to see if they satisfy the given function, which can be quantitative or qualitative, based on which further modifications are made. Change processes can be used to generate alternative designs [14]. They can also be used as a means of repair. Knowledge-lean processes such as simulated annealing can be used to make the changes [16], but so far they have proved more successful where the functions can be quantified and components used are similar in characteristics to each other, e.g., structural optimisation. In cases where functions are hard to quantify, knowledge-intensive processes (e.g., qualitative model based reasoning [15]) have been relatively more successful.

In general, for designs with functions that are hard to quantify, change processes for repair are likely to produce designs that are more innovative and risky than those directly retrieved, but less so than compositionally synthesised designs, especially if compositional synthesis is done using less composite elements as building blocks. The resource involved in development is also usually in-between, although not always, as seen from a comparison between a compositional synthesis process and a retrieval-based modification process [21].

3 Analysis

Possible combinations of the three processes underlying the main sub-approaches - compositional synthesis, retrieval and change - are systematically enumerated in this section. The change process is seen to take place in terms of two alternative ways: retrieval of a design part of which needs to be replaced, or creation of a design using compositional synthesis in order to use that as a replacement. Each combination enumerated can be viewed as a possible synthesis scheme. These combinations are then put together into a minimal graph that is a union of the combinations. Since this minimal graph has the individual combinations as its subsets, this graph can be treated as a generic synthesis framework from which all these synthesis approaches can be derived.

3.1 Combinations starting with compositional synthesis

Compositional synthesis (CS) starts with a function and no structure, and ends with one or many alternative structures – combinations of building blocks – that fulfil the function.

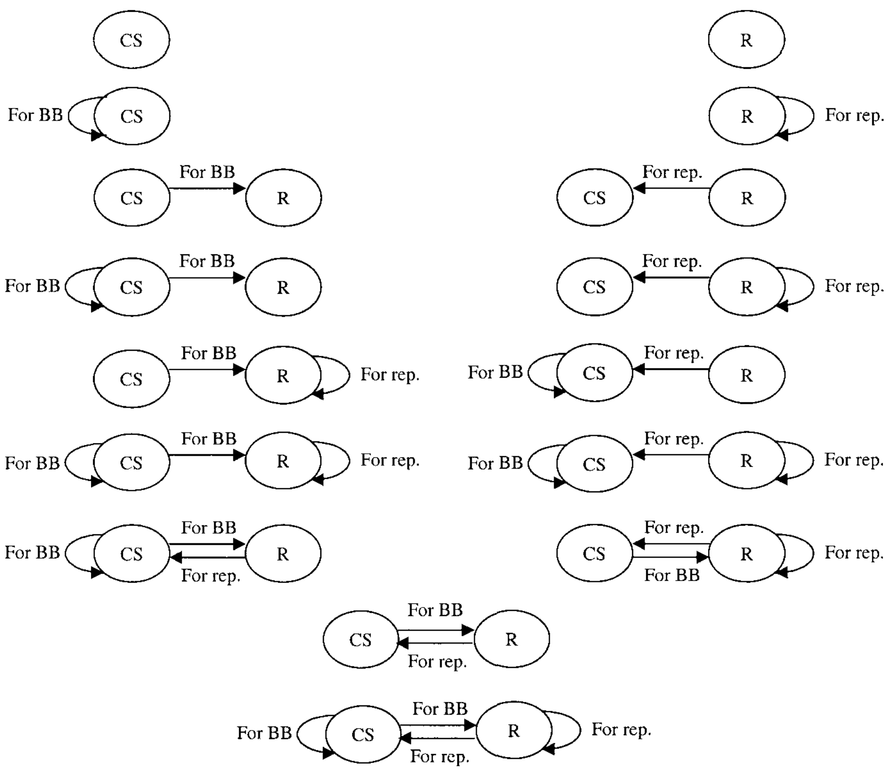


Figure 1. Hybrid synthesis possibilities

The first combination possibility is where a structure that can fulfil a given function can be developed using, or is constrained to remain within the set of building blocks with which synthesis begins. This is pure compositional synthesis, shown as the top left option in Fig. 1.

The second is the possibility where the existing building blocks are not sufficient to fulfil a certain sub-function, at least within the constraints imposed (e.g., the number of building blocks to be used in a solution). Some of these building blocks, therefore, are combined into more composite building blocks to be used in the synthesis (2nd down from top left, Fig. 1).

The third possibility applies where the existing building blocks are not sufficient to fulfil a certain sub-function. An existing design is retrieved (R) as a new building block, to be used in the composition process. This is a combination of synthesis and retrieval (3rd down from top left, Fig. 1).

The fourth possibility applies when the building block retrieved does not fulfil the function entirely and a change, to fulfil the remaining function for instance, is necessary. This is done by developing a building block using compositional synthesis (4th down from top left, Fig. 1).

The fifth is similar to the fourth, except that in this case, instead of creating new building blocks using compositional synthesis, the retrieved solution is changed using further retrieval for replacement of parts of it that do not work (5th down from top left, Fig. 1).

In the sixth case, new building blocks are created and used in compositional synthesis using both compositional synthesis and retrieval, and yet the function is not fully satisfied. So, the retrieved solution is further amended using further retrieval (6th down from top left, Fig. 1).

The next (next down) is similar to the sixth, except that instead of using further retrieval as the process of further amending the solution retrieved, retrieval uses compositional synthesis for this purpose.

The eighth case (last but one in the figure) starts off with compositional synthesis as the process to change the solution originally retrieved, and retrieval as the process for identifying further building blocks necessary in the compositional synthesis process.

The ninth (last down in the figure) case is similar to the eighth, but also includes processes for generating further building blocks using compositional synthesis, and processes for further change using further retrieval.

3.2 Combinations starting with retrieval

Retrieval (R) starts with a function and no structure, and ends with one or many alternative structures – retrieved from a case-base – that fulfil the function.

The first possibility is that a structure that can fulfil a given function is obtainable directly from the case base, shown as the top right option in Fig. 1.

The second is the possibility in which the solution originally retrieved is not sufficient to fulfil the function and is changed by replacing parts of it using further retrieval (2nd down from top right, Fig. 1).

The third is similar to the second, except that the change process for replacement is done using compositional synthesis rather than further retrieval. This is a combination of synthesis and retrieval (3rd down from top right, Fig. 1)

The fourth possibility involves change processes using compositional synthesis as well as further retrieval. This would be the case where the one done first does not quite fulfil the function and therefore further changes are necessary (4^{th} down from top right, Fig. 1).

The fifth is similar to the fourth, except that in this case all the changes are done using compositional synthesis, which uses compositional synthesis to develop further building blocks necessary (5^{th} down from top right, Fig. 1).

In the sixth case, further retrieval is tried first as the change process for amending the solution originally retrieved. However, this is not sufficient to fulfil the function in this case, and further changes are made using compositional synthesis that uses further compositional synthesis to develop building blocks (6^{th} down from top right, Fig. 1).

The next (next down) is similar to the sixth, except that instead of using further compositional synthesis as the process of developing further building blocks, compositional synthesis uses further retrieval for this purpose.

The eighth case (last but one in the figure) starts off with search for building blocks using retrieval, and its amendment using compositional synthesis.

The ninth (last down in the figure) case is similar to the eighth, but also includes processes for generating further building blocks using compositional synthesis, and processes for further change using further retrieval. Note that these last two cases are common in both the processes that start primarily as either compositional synthesis or retrieval.

3.3 Overall Framework

The cases described in Section 3.2 illustrate two things. The first is, since a complete design as dug out by a retrieval process can also be used as a building block, retrieval can be used as a sub-process of a compositional synthesis approach. Now, when a design retrieved by a retrieval process does not quite provide the intended function, it needs to be changed, and thus change approaches can be used as sub-processes of a compositional synthesis approach.

The second is, when a design is retrieved and needs changing, a possible change process may be a compositional synthesis approach, which may use a design reuse approach as one of its sub-processes. In other words, the two approaches can bootstrap, leading to more symbiotic, capable and possibly more complex schemes, in which the two original, apparently distinct and conflicting approaches can be sub-processes of each other.

This bootstrapping is what is indicated in the bottom four cases in Fig. 1 in all of which arrows connect compositional synthesis and retrieval in both directions. This symmetry allows the process to start primarily as a compositional synthesis or retrieval. This, for instance, can be done by re-interpreting the seventh case down from top left in Fig. 1 (originally interpreted primarily as compositional synthesis) as primarily a retrieval process. In this new interpretation, the process starts as retrieval which uses compositional synthesis as its sub-process for replacement of parts of the solution retrieved, which in turn uses further compositional synthesis as well as retrieval as sub-processes to develop further building blocks. Similarly, the seventh case down from right in the figure (originally interpreted primarily as retrieval) can be re-interpreted primarily as a compositional synthesis process, which uses retrieval to generate its building blocks, which in turn uses further retrieval as well as further compositional synthesis to amend the building blocks.

It should be noted that the case lowest down in Fig. 1 has all the other cases subsumed in it, and is the minimal graph that includes them all. This graph is taken here as the generic synthesis framework (see Fig. 2).

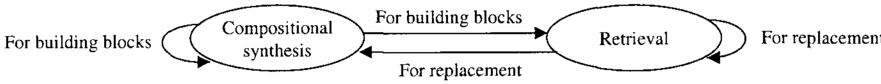


Figure 2. Generic synthesis framework

If the two ovals are considered in isolation, they show compositional synthesis and retrieval as alternative direct approaches to synthesis. However, if direct retrieval does not work, the retrieved design may need further retrieval for replacement (shown by arrow pointing back to retrieval, on right) or compositional synthesis in order to create the replacement of some of its parts (shown by arrow at the bottom). Similarly, if direct compositional synthesis does not work, it may require developing further building blocks (BB), generated by further synthesis (arrow on right), or further retrieval of designs to be used as building blocks (arrow on top). The process can go on several times, and the arrows together can be seen to represent iterative or recursive cycles.

4 Examples from existing synthesis approaches

The first two cases – of direct compositional synthesis (top left in Fig. 1) and retrieval (top right in Fig. 1) have many examples in design research literature [2, 4, 7, 8, 19, 20]. Retrieval and adaptation by retrieval (2nd down from top right, Fig. 1) can be seen for instance in [6] where design prototypes are retrieved from case memory, on which GA operators are applied to generate new designs. An example of limited case adaptation using both compositional synthesis and further retrieval (fourth down from right in Fig. 1) can be seen in [22]. However, there seems to be no example of the other combinations enumerated, which points to the gap in current synthesis methods and the scope for developing them along these lines.

5 Conclusions and further work

The main conclusions are:

- There are two apparently distinct and contrasting philosophies in design synthesis research: design reuse (retrieval and change) and compositional synthesis.
- There is a variety of approaches under each, both design-led and automated.
- It is possible to describe change in terms of synthesis and retrieval.
- The approaches can be explained as subsets of a generic synthesis framework consisting of synthesis and retrieval operations, with iterative or recursive cycles between the two.
- None of the existing approaches presently allow such seamless combination; however evidence suggests that designers already do some of these and it would help if approaches to help designers synthesise also had this flexibility.

This leads to a host of research issues: how can each such hybrid approach be supported? Where is each one applicable? How far should iterative cycles go on until it is understood that there can be no solution possible within a base of knowledge? The two potential challenges are knowledge representation and development of rules that are necessary to get to the relevant knowledge in an efficient way. These are future directions for this work.

References

[1] Pahl, G., and Beitz, W. Engineering Design: A Systematic Approach, Springer, London, 1996.

[2] Chakrabarti, A. Designing by Functions, PhD Thesis, Department of Engineering, University of Cambridge, 1991.

[3] Roth, K. Systematik der Maschinen und ihrer Mechanischen Elementaren Funktionen, Feinwerktechnik, 74, 453-460, 1970.

[4] Qian, L. and Gero, J. S. Function-behaviour-structure paths and their role in analogy based design. AIEDAM Special Issue on Representing Functionality in Design, A. Chakrabarti & L. Blessing (Eds.), 10(4), 289-312, 1996.

[5] Ullman, D.G. A New View on Function Modeling, Proc. International Conference on Engineering Design, 1, 21-28, The Hague, 1993.

[6] Gomez de Silva Garza, A., and Maher, M. L. A Process Model for Evolutionary Design Case Adaptation, J.S. Gero (Ed.) Artificial Intelligence in Design'00, 393-412, Kluwer Academic Publishers, 2000.

[7] Ulrich, K.T., Computation and Pre-Parametric Design, PhD Thesis, AI Laboratory, Massachusetts Institute of Technology, Cambridge, 1988.

[8] Bracewell, R. H., and Sharpe. J. E. E. Functional Representations Used in Computer Support for Qualitative Scheme Generation – Schemebuilder, AIEDAM Special Issue on Representing Functionality in Design, A. Chakrabarti & L. Blessing (Eds.), 10(4), 333-346, 1996.

[9] Smithers, T. Personal communication, 1999.

[10] Chakrabarti, A. and Bligh T. P. Exploring Basicness of Building Blocks for Supporting Innovative Design of Mechanical Systems using the Building Blocks Metaphor, Technical Report No. CUED/C-EDC/TR61-January 1998, University of Cambridge, 1998.

[11] Liu, Y.C. A Methodology for the Generation of Concepts in Mechanical Design, PhD Thesis, Department of Engineering, University of Cambridge, 2000.

[12] Kolodner, J. Case Based Reasoning, Morgan Kaufmann, San Mateo, 1993.

[13] Pu, P., and Purvis, L. Formalizing Case Adaptation in a Case Based Design System, J.S. Gero and F. Sudweeks (Eds.) Artificial Intelligence in Design'94, 77-91, Kluwer Academic Publishers, 1994.

[14] Hammond, K. J. CHEF: A Model of Case Based Planning, Proc. AAAI-86, Philadelphia, PA, 267-271, 1986.

[15] Finger, S., and Rinderle, J. R. A Transformational Approach for Mechanical Design Using a Bind Graph Grammar, EDRC Report No. 24-23-90, Carnegie Mellon University, USA, 1990.

[16] Sycara, K., and Navinchandra, D. Integrating Case Based Reasoning and Qualitative Reasoning in Engineering Design, J.S. Gero (Ed.) Artificial Intelligence in Design'90, Computational Mechanics Publications, Springer Verlag, 231-250, 1990.

[17] Shea, K. Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Direct Stochastic Search, PhD Thesis, Canegie Mellon University, USA, 1997.

[18] Liu, J-S., Parks, G. T., and Clarkson, P. J. Metamorphic Development: A New Topology Optimisation Method for Truss Structures, The 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Paper AIAA-99-1387, St Louis, MO, USA, April 12-15, 1999.

[19] Medland, A. J., Mullineux, G., Twyman, B. R., and Rentoul, A. H. A Constraint Based Approach to the Design and Optimisation of Mechanisms Systems, IUTAM Symposium on Optimisation of Mechanical Systems, D. Bestle and W. Schiehlen (Eds.), Kluwer Academic, Netherlands, 1996.

[20] McGarva, J. R. Rapid Search and Selection of Path Generating Mechanisms from a Library, Mechanisms and Machine Theory, 29(2), 223-235, 1994.

[21] Chakrabarti, A., and Bligh, T.P. A Scheme for Functional Reasoning is Conceptual Design, Technical Report CUED/C-EDC/TR60, Cambridge University Engineering Department, January 1998.

[22] Hua, K., and Faltings, B. Exploring Case Based Building Design - CADRE, AI EDAM, 7(2), 135-143, 1993.

Corresponding author's name: Dr Amaresh Chakrabarti

Engineering Design Centre, Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, UK, Tel: +44 1223 766961, Fax: +44 1223 332662, E-mail: ac123@eng.cam.ac.uk, URL: http://www-edc.eng.cam.ac.uk/people/ac123.html

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

'INNOPLAN' – AN ADAPTATION OF THE METAPLAN-TECHNIQUE FOR A NOVEL COMPUTER SUPPORTED METHOD OF TEAMWORK

T Kennel, A M Kunz, and S Müller

Keywords: Creativity and Innovation, innovation methods, psychology of creativity, computer supported cooperative work;

1 Abstract

The moderation-method (Metaplan-technique) is a popular and established method to create innovative ideas in a team session as well as to structure and evaluate them.
Although computer aided work is commonly used in lots of proximate development processes, there is hardly no use of information technology for creativity-techniques. Effective teamwork is hindered, because so far it is impossible to feed a computer simultaneously with multiple input-devices.

The development of a novel work environment as well as the use of modern in/output-devices) permits in combination with special software a new form of collective interaction and an adaptation of the Metaplan-technique. Scope of the paper is the further development (in the following named InnoPlan) of a known method and to describe the required infrastructure, as well as to present first results of an IT-based teamwork..

2 Introduction

The future success of a new product is based on the work in the early stages of product development processes. In this phase, ideas are created, problems and tasks are defined, solutions are searched, visualized and structured.
Shorter innovation cycles and increased market dynamics require efficient and flexible planning processes. Specially in the early stage of the product development process, the band width of the problem situation and the resulting essential knowledge necessitate both an interdisciplinary as well as a team based procedure.
One important part of the planning process is the moderation method (Metaplan-technique) that has been proved for more than 30 years. In moderated team sessions all members contribute with own written cards. Together all these cards are discussed, clustered and pinned on a board. The result are thematically structured visualizations on posters, which can be used for further discussion or photographed and attached to a journal of a Metaplan-session.
A complete Metaplan session consists of three to four steps an will take several days. This procedure can be varied under certain aspects. It depends on the composition of the group, on the tasks and on the creativity of the moderator. Since this method consists of independently combinable modules, it can be used very flexible. Using this method requires a lot of accessories, which are conventionally available (e.g. flipchart...)

Although the techniques and procedures of the moderation-method had been refined since their beginnings and are still an established manner, the handling is not comfortable at all. Transportation of posters and moderation equipment, pinning and gluing of cards on flipcharts, as well as photographing or scetching of the posters in order to take the protocol is not state of the art, bulky and has to be improved. However the method to write down ideas, to cluster them, to develop and structurize them, in order to get an overview of the processes still has lot of advantages

Previous attempts to support Metaplan technique and other moderated teamwork by means of the information technology were not successful. The touchsensitive whiteboards (Smartboard, Team-Board ...) are designed for presentation purposes and there vertical orientation forces a frontal interaction. Since only one input is possible per time interval , a parallel and simultaneous working is impossible and thus prohibits an effective teamwork. The seriel input of ideas is time consuming and blocks up all spontaneity which is elementarily important for the success of this method. A horizontal orientation of the touchsensitive boards together with an ergonomical handwriting causes quantisazion errors and is thus inapplicable. At the present time the advantages of the digitalized inputs do not compensate the disadvantages from the above. In addition formerly elaborated posters are only available by scrolling backwards and can not be compared and attached to the wall as a reminder.

3 Motivation

A time and cost efficient product development presumes that all relevant product data are available and accessable to all members of the team. All data that are collected during the lifecycle of a product fee the *Digital Product.* The usage of information technology in the early stages of the product development process enables the "birth" of the digital product already with the first ideas.
The adaptation of the well known Metaplan technique as well as their redesign in aspect to novel hard- and software allows an synenergetic usage of the information technology and the paperbased teamwork. An intuitive handling of the whole system should enable the users to be creative and innovative.
Working with new technology and functional software, together with an attractive user interface, increases the motivation of all team members and smooth the way for novel applications.
Nowadays the protocolling of team sessions is time consuming and needs additional technical equipment (cameras, scanners, copiers, digitizers...). Information technology simplifies protocolling, managing and distributing of elaborated work. Due to a automatic „history-function" the time flow of a team session can be browsed and thus creates a new quality of protocolling.

After a successful completion of the development the whole system, method and infrastructure will be used as an IT-based planningsystem for moderated teamwork in industrial applications. Further optimization and development will follow after first experimental results of this new system (project name: InnoPlan-System).

4 Contributions

Adaptation of the Metaplan-technique into an IT-environment:
A successful transformation of the Metaplan-technique into an IT-based surrounding requires a perfect combination of all functional parts. This are methodology, moderation equipment (conference room with integrated technical infrastructure) and moderation, which together form the novel InnoPlan-system.

Methodology:
The Metaplan-technique is subdivided in single steps, that are rebuild with means of information technology. Synenergy will result from the usage of novel technologies and the adaptation of the moderation technique. Several procedures are identical both in the InnoPlan and Metaplan technique, others are optimized to be technical supported. This will allow an easy and comfortable usage of the new system. The goal of the transformation and adaptation of this method is an simple and intuitive application as well as its transparency and reproducability.
One important aspect of the proposed work is the design of the software and of thus the man-machine-interface (MMI). For individual software designers and ergonomists commonly use the metapher of a desktop. Teambased applications ask for new metaphers. The combination of individual in/output devices (InnoPlan pads) together with collective workspaces and reminders requires a consistent design of the grafic user interface GUI in order to realize an intuitive usage by untrained persons. Tasks like writing or scetching on cards, exchanging them between InnoPlan-pads and publishing them on collective workspaces or reminders, teambased clustering or editing as well as new ways of taking a protocol should be easily handled after a short training. This enables the user to concentrate on his creative work.
An other important aspect of this work is the design of an ideal work environment. The architectual and interior design of the conference room under ergonomic and psychologic aspects as well as the optimal integration an combination of the needed IT-infrastructure, are the main focuses.

Conference room:
The maximum number of persons for moderated team sessions with the InnoPlan method is about 10 to 12. Larger groups need different techniques based on different interaction and teamwork. The conference room has to support a teamwork in plenum as well as in smaller subgroups. Therefore a flexible furnishing is needed and a room size of at least 40 m2. A fixed desk contains the central infrastructure (like server, printer, document camera, video tape recorder, wireless LAN-receiver ...). Other desks can be moved around an combined according to different team sizes. It has to be considered that the collective workspaces and reminders has to be visible from all positions. Relevant data are projected on two interactive boards (1,5m x 1,1m) and two reminders (equal size) by four data projectors mounted at the ceiling. The images can be randomly arranged and exchanged on the four projection screens. Also a simultaneous projection of an identical content on all screens is possible in order to avoid the shading by the presenter.
It has to keep in mind that architectural and interior Design (illumination, temperature, noise, smell...) affects the creative environment as well as the offer of refreshments (coffee, water, snacks...).

The "novel" characteristic of the moderation environment:
As a basic requirement the conference room has to provide a motivating environment to a temporarly gathered team. In addition the InnoPlan-system includes the complete technical infrastructure, allows the communication with distant teams or specialists, facilitates a spontaneous information requirement, protocolling and printing or plotting of meeting results. This functions are not possible in conventional conference rooms without additional equipment. In conclusion the new conference room is an elemtary functional element of he InnoPlan method.

Moderation:
Usually a Metaplan-session is moderated by two persons: one is moderating while the second one is visualizing and pinning cards. This roles can be changed. In exceptions (spontaneous moderations, short moderations) a single person can moderate. Also at InnoPlan meetings a moderation by two persons is recommended, because additional tasks like managing the technology and the moderation system. The co-moderator is responsible for arranging the projections to the different working and reminder screens, the handling of the peripherique equipment an the taking of the protocol, that can be distributed to the team members at any time. As an exception the InnoPlan session also can be done by a single trained moderator.

Handling and functionalities of the InnoPlan-system:

The individual in/output device (InnoPlan pad). During the session, every team member has an own InnoPlan pad as a multifunctional device (figure 1), that is personalized to him and logged in to the InnoPlan system. This enables or supports the automatical protocolling. Similar to a web-pad, the InnoPlan-pad consist of a touchsensitive display (apx. DIN A5) with an integrated computer, which can be operated by a pen. This allows handwriting, scetching, automatical recognition of handwriting as well as an input via a n onscreen keyboard. The pads communicate with each other an the server by wireless LAN or BlueTooth technology. The intuitive GUI of these pads, allow to generate cards, to edit and delete them. Within a next step these ideas can be classified and grouped or clustered. Also cards can be send from pad to pad or from the collective work area to a pad (e.g. when working in subgroups or when using the 6-3-5 method). The pads allow to publish cards in order to appear and to be progressed on the collective working area.

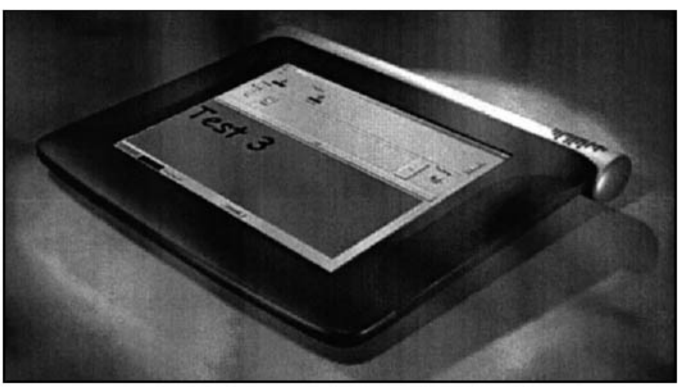


Figure 1. InnoPlan-Pad, the individual interaction unit of the InnoPlan-system

The collective work area corresponds to the Metaplan poster. Here, the published cards can be explained, discussed, reviewed or clustered .... Like on the InnoPlan pads the interaction on the digital Whiteboards can be done with a pen. In addition to moving, clustering and arranging, the cards can be supplemented with additional comments, headlines or labels. The GUI on he collective work area correspond to the pad GUI but is enhanced with additional collective functions. This allows for example to save and restore posters as well as to move them to other working areas or reminders. Posters or definable sections can be plotted or printed. From the collective work area the protocolling function can be controlled too. In particular the „history function“ has to be mentioned, that allows to scroll by use of a time-slider, to any desired chronological position of a poster or even to play the evolution of a poster as a film.

The reminders are passive projection screens, and do not allow any interaction. They are used to visualize ellaborated posters or other important information (also multimedia) to the team. The contents of the reminders can be controlled by the collective work area or by the moderator system, e.g. arranged, exchanged or restored on an active work area.

The moderator system is the central element of the InnoPlan-system. Here all data from the wireless LAN are merged and prepared for visualization. The moderator system consists of a graphics workstation and a screen, that is controlled via mouse and keyboard. These possibilities are used to open a new session, to personalize the InnoPlan-pads and to aquire additional information for protocolling and to control the peripheral.

Peripheral functions. Acquiring additional information from slides or printed matters can be done by a document camera. With this camera it is also possible to visualize parts or details on a reminder area. As already mentioned, all intermediate results or finished posters ore definable sections can be printed. The printers and plotters can be accessed by the moderater system. Also additional information, which are important for the result of the InnoPlan session can be acquired from the intranet or internet by the moderator system. As default the pads don't have direct web-access in order to minimize any disturbance by private surfing. However this can be changed by the moderator system. In case of distributed team sessions, where one subgroup is physically distant but is working on the same theme or in case of involving external specialists into a session, a video-conferencing-system can be used for communication.

Differences InnoPlan /Metaplan:

Similiar to the Metaplan technique also the InnoPlan-technique presumes that all spontaneous ideas are noted individually on cards. This cards are published after a preselection, they are discussed and structured on the collective working area. Like in the Metaplan-technique elaborated Posters can be projectedon a separate screen as a reminder.

Unlike the metaplan technique within the InnoPlan technique cards can be grouped and the resulting clusters can be moved or simplyfied by the use of the level-of-detail-function (LOD). This functionality improves the clarity and the possibilities of organization. Cards can be edited at any time (size, color, shape, font type and size...). Each card is automatically appended with protocol data that can be added with individual annotations. These additional information can be viewed and edited at any time. The collective work area can be zoomed and scrolled in order to simplify the structuring of the cards.

Procedure of a team session:
Preparation of a session. The moderator starts the InnoPlan-system and enters all relevant data (topic, place, timetable, members...) to open a new session. The InnoPlan pads are personified for the following workshop, specially information about the pad user, further participants, subject... are entered, so that the pad can welcome the arriving participants with a friendly "Good morning Martin and a lot of fun at the workshop "new technologies in teamwork".
Opening of a session. After all participants found their pad and thus their seat, the subject is introduced by the moderator. For this task, the InnoPlan-system is used as a presentation-system and the moderator can inform about the topics, the goal of the meeting and the timetable by showing videos, powerpoint presentations, prepared graphics or also by writing on the touchsensitive boards. For firsttime-user of the InnoPlan-system a short instruction in the usage of this system (max. 5 minutes) can follow.
Writing on cards. The workshop can start now and all participants note their ideas and propositions on the virtual cards using the InnoPlan-pads .
Publishing cards. These cards can be clustered and grouped on the pad and be sent to the collective work area.
Clustering. On the collective work area the incoming cards appear as a symbol near the icon of the corresponding team member. From there, they can be moved, annotated and placed on the collective working area by the author or the moderator. In a next step, all inputs are clustered, sorted in logical structures and dependencies and grouped by the plenum.
Complementing and criticism. During the collective work, cards can be edited by the author on the collective work area or by returning them to the author's individual pad. Declarations and remarks can be added to every card in the drop-down-annotation-menu.
Evaluation. Titles, marks, references... can be added on the collective work area by the participants. The possibility to edit color, shape, size, font and design features of all elements allows a lucid visualization.
Posters. Processed work areas are stored as posters. By the use of icons on the collective work area all elaborated posters are shown and can be simply restored. Due to their digital form, the posters can be displaced onto reminder areas and processed with the computer in any known manner (plotting, printing, sending...).
Protocol. The system automatically generates a protocol of a session and stores any action in a log file with a time stamp and a person assignment. Thus, every card has an edition-history, which shows the evolution in the drop-down-annotation-menu. All publications and actions that are done on the collective work area, are recorded with the "history function". With a slide control (Timeslider) any chronological position of the evolution of a poster can be recalled. Also parts or whole posters can be viewed as memomotion study.

Specials of the InnoPlan-system:
The proved procedure of the Metaplan session is complemented with the efficient possibilities of the information technology. For presentation purposes the multimedia functionality can be used and the access to the Intranet/Internet allows an easy acquisition and distribution of information. The simple displacement and exchange of the contents of the collective work area and the reminder increases the clarity and simplify the presentation of earlier elaborated posters. The digital protocolling function facilitates the taking of a protocol and the history function gives the possibility to make a "time trip" to any desired evolution position of a poster by moving the time slider.

The direct communication between the individual InnoPlan-pads supports the work in small groups as well as the usage of other methods and techniques (e.g. 6-3-5-method). Therefore the function of the collective work area or of the moderator system can be given to one InnoPlan-pad of this small group. In order to collect the results, the data of this pad will be sent back to the master moderator system to be worked out in plenum.

5 First Results

First series of experiments with a prototype of the InnoPlan system showed that the handling of the installed information technology is very simple and intuitive. A positive result was the good acceptance and the spontaneous and natural use of the novel system also by persons which are not used to work with computers. The usage of novel technologies, the fun factor and the combination with a well known and easy to understand method increases the motivation even with users that very often work with the Metaplan technique.

The equipment of conference rooms should include at least 4 projection areas from which in minimum one has to be interactive. The tests with the prototype showed, that the displacement and the new arrangement of the contents of working and reminder areas has to be redesigned for a simpler handling. A further result was, that the extension of the InnoPlan-system with interactive small teamwork places can increase the possible applications from information technology in the early stages of the product development process.

6 Conclusions

The experiments with the prototype showed that the usage of information technology is not only possible in early stages of the product development process, but an efficient and interesting alternative to the traditional techniques. The direct projection of the Metaplan technique in a technology supported environment as a first application for interactive teamwork in creative groups was successful. The multimedia possibilities, the fast availability of information via Intranet/Internet as well as the automatic protocolling and the fast distribution of the digital results and information are essential improvements of the traditional Metaplan-system. Nevertheless, the potential of the used technology allows the development of new and different methods and applications for interactive teamwork..

7 Future work

Because teamwork often requires flexible and spontaneous work in different constellations, the possibilities of an IT-supported environment also in these fields has to be expanded. In a first step a workplace for smaller groups is supposed to be developed. This interactive place will allow a subgroup of 4 to 6 persons to elaborate ideas and solutions that can be processed in plenum later on at the collective work areas of the InnoPlan system. A network of such small teamwork places enables a locally distributed work as well as the usage of new creativity and moderation techniques.
In a next step it is planned to network multiple InnoPlan systems, so that CSCW (ComputerSupportedCooperativeWork) and video conferencing will be enriched with new possibilities and qualities of interaction and communication.

In parallel with the technical development new methods and methodologies should be developed, tested and implemented, which use in an optimal form the expanded possibilities of the technology supported interactive teamwork.

8 Acknowledgements

The above project was funded by the Swiss research fund KTI 4475.1PMS. We thank in particular the IfAP (Institute for Work and Organization Psychology) and the HGKZ (University of Art and Design Zurich).

References

[1] Ernesto Arias, Hal Eden, Gerhard Fischer, Andrew Gorman, Eric Scharff; „Creating Shared Understanding through Collaborative Deisgn"; University of Colorado, Boulder; CHI-Proceedings

[2] Thomas P. Moran, Patrick Chiu, William van Melle, Gordon Kurtenbach; „Implicit Structures for Pen-Based Systems Within a Freeform Interaction Paradigm"; Proceedings og CHI 1995

[3] Jason A. Brotherton, Janak R. Bhalodia, Gregory D. Abowd; „Automated Capture, Integration, and Visualization of Multiple Media Streams"; Proceedings IEEE 1998

[4] Orfeo Niedermann; „Nutzenanalyse des Einsatzes der Virtual Reality Technologie in der Failure Mode and Effect Analysis"; ETH Internal Research Report 1999

[5] Klebert, K.; Schrader, E.; Strub, W.G.; „KurzModeration: Anwendung der ModerationsMethode in Betrieb, Schule und Hochschule, Kirche und Politik, Sozialbereich und Familie bei Besprechungen und Präsentationen“; 1987, Hamburg

[6] Die Fibel zur Metaplantechnik, Metaplan GmbH, Goethestraße 16, D - 25451 Quickborn http://www.moderationstechnik.de

[7] Peter Troxler, Kristina Lauche, Kyeni Mbiti; „The Use of Interactive Boards for Collective Design Processes“; International Design Conference, May 23.-26, 2000

Mr Thomas Kennel
Swiss Federal Institute of Technology Zurich, Centre of Product Development, Tannenstrasse 3, CH-8092 Zurich, Switzerland, Tel: +41 (0) 1 632 23 13, Fax: +41 (0) 1 632 11 81, e-mail: kennel@imes.mavt.ethz.ch

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

UNDERSTANDING AND SUPPORTING INNOVATION IN TEAMS

K Lauche, R Ehbets Müller, and K Mbiti

Keywords: Innovation methods, design teams, cooperative design, protocol analysis, descriptive models of design

1 Introduction

It is during the early phases of product innovation processes that the main direction of a project is determined. This period is also important for the development of a shared understanding of the general problem so that each discipline involved can start from a common basis. Marketing people, designers, engineers and managers should understand each other's interest as far as this might influence their own work. Despite their impact, early phases have not been researched very much. Most empirical research relies on protocol analysis of individual work on small, artificial tasks or on ad hoc groups in laboratory situations. Observation of team interaction has so far only been conducted on later phases [1]. To understand and support the early processes, we observe early phases of innovation projects in different industrial settings and different design traditions. We explain our theoretical approach and the our research method for protocol analysis. Options for support are suggested in terms of coaching techniques and an innovative setting.

2 Theoretical approach to innovation processes

The challenge for early phases of design is to "open the mouth" and to "narrow the neck" of the innovation funnel [2], i.e. to include a broad range of ideas and to focus on the right ones. Our theoretical framework combines problem solving methods as used in systems engineering with a model of collective action regulation [3]. Both approaches include elements of reasonable strategies for solving complex problems such as goal setting, structuring the problem and evaluating the suggested ideas. Laboratory experiments have shown that successful individuals and teams show complete action cycles with all these steps. It has also been shown that creative processes may be stimulated by the presence of others. However, collective conceptualisation needs professional support. Otherwise groups tend to develop too much conformity whereby good ideas may be disregarded, a phenomenon known as group think. A technique for facilitating problem solving in teams is the moderation method [4]. The major instrument is explication and visualisation of questions and answers. A coach, who does not take part in the problem solving itself, helps the team to focus on certain relevant aspects, stimulates individual or collective brainstorming and keeps a structured record on a public protocol. Our hypothesis for the empirical investigation was that this technique could be used to develop a shared understanding, to generate more ideas, and to integrate them into a successful solution. Also, reflection of the process was expected to foster learning and improve strategies.

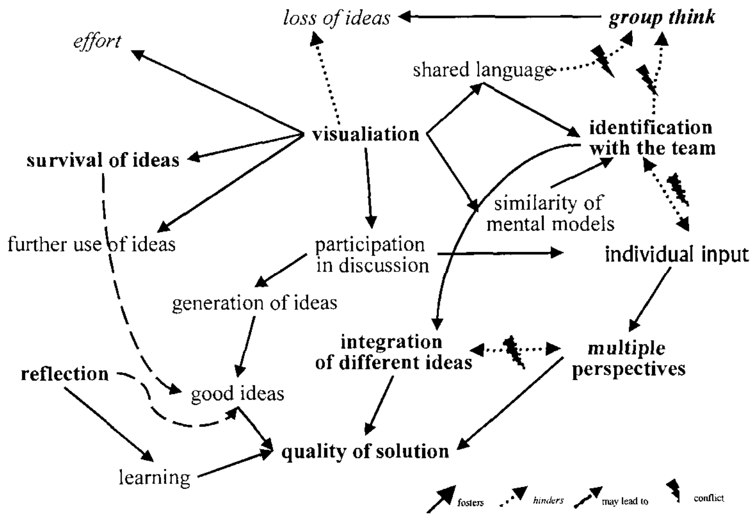


Figure 1. Assumed influences of visualisation on shared language, integration, and quality of solution

Figure 1 shows the assumed influences. Visualisations are seen as important to develop a shared understanding and may contribute to survival and further use of good ideas. Identification with the team may or may not be a positive factor towards the quality of solution: It can lead into group think which hinders creative individual input and multiple perspectives. However, it is essential for the integration of different ideas. Reflection may influence the quality of solution via the evaluation of ideas and learning, as the survival of ideas may contribute to the survival of the good ones.

3 Research Methods for observing interaction

As a research method, we used online observations of real teamwork in the mechanical and consumer goods industry. Meetings were video taped and analysed according to the theoretical framework of action theory and design methodology. Different areas and types of meetings were compared to understand what helps a team be innovative under what circumstances. We developed a category system, three types of data recording and criteria for the quality of the meeting. It differentiates between the primary innovation task to produce solutions for the needs, the secondary social task to organise the allocation of work and resources, and the meta-task of monitoring and coaching the team. Also, the use of media is classified.

primary task	secondary task	structuring task	use of media
goal setting	positive emotions	stimulating questions	flipchart
identification of needs	negative emotion	procedural suggestions	video beamer
generation of ideas	internal organisation	summaries	cards
integration of ideas	external organisation	reflection	private notes
evaluation			
decision			

Figure 1. Category system

3.1 Three types of recording data

We have developed three types of data recording: The first is dedicated to providing immediate feedback to the participants after the meeting. On a protocol sheet, the observer notes the time, the person and keywords of oral speech and artefacts or visualisations used. These protocol sheets give an overview of the whole meeting and help identifying which parts deserve further investigation but are not used for scientific purposes. For a detailed analysis of the content, the videotapes are transcribed and classified according to the categories, which is a consuming procedure and requires intensive training of observers. Besides being the best basis for personal feedback for team leaders, the practical value is in most cases restricted to retrieve lost ideas. Therefore, we also developed a faster way of analysing team processes. One part of the solution was a reduced category system. The second part was providing technical support for recording: on a pen-based, handheld computer, the person and their action are marked – time is automatically registered [5, for the palm application see www.smiledesign.ch].

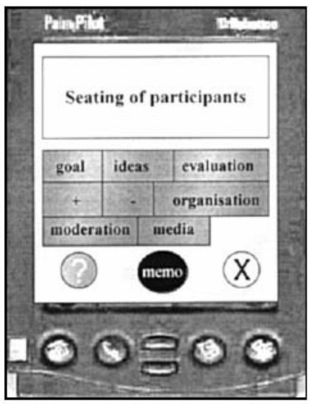


Figure 2. Fit-system with reduced category system

Due to good usability, observers were able to keep recording actions for up to two hours without fatigue. The data can be transferred into an excel sheet to analyse the activities of the team and the different persons. They are analysed for comparison of the interaction patterns, the overall content of the meeting and for the use of media and artefacts.

3.2 Criteria for the quality of meetings

In order to evaluate meetings and to differentiate between good and poor practice, we formulated criteria for how the meetings were conducted. The final criterion for any innovation process is that the product can be successfully launched on the market and creates new potential for the business to perform future projects [2]. This final criterion, however, cannot be determined in the early phases and is dependent on many other factors. Therefore, we defined criteria closer to the actual behaviour in the observed sessions.

Criteria for co-ordination meetings

For a co-ordination and decision making meeting the ideal was defined as follows. The goal setting should include an agenda visualised on a flipchart or board which remains visual throughout the meeting. If a list of topics is prepared, input from the participants can be added, and the topics are then prioritised. The team should know relevant key figures and milestones and should have an established procedure of decision making. As results and proposals are presented, someone is in charge to keep track of the comments and decisions on a public protocol visual to all. This protocol may be formatted as matrix of what was decided, what remains to be done by whom and until what date. The person writing the protocol may initiate a summary of the meeting and a reflection of the adequacy of the collective planning.

Criteria for idea generation in subtasks

For a meeting focussed on the solution of one subtask of the innovation, goal setting has usually done beforehand. The team should have transparency of the criteria and their rationale and should recall important facts and decisions on the subjects. They can employ different methods of idea generation altering between individual and collective work as long as the results are documented. Visualisations of individual results can serve as a basis for discussion and should therefore be accessible by everyone (e.g. big size, not restricted to one professional code of sketching). When integrating and evaluating ideas, they should be compared to the needs and the technological options. As even the most technical type of meeting has social and co-ordination functions, there should be time devoted to the internal organisation of work, what is required from others and a reflection on how the meeting went.

Criteria for strategic meetings in early phases

For a kick-off meeting or a convention of the whole team in the early phases to determine what direction the innovation process should take, goal setting an important step which has to be done in the team. A helpful start is an orientation by the management about the overall strategy, that also serves to signalise the relevance of the project and support from the organisation. The challenge for these meetings is to "open the mouth" of the innovation funnel, which means to proceed beyond the known. Therefore, an instruction to envision the future or methods for encouraging fantasy may help to stimulate creative, new, even wired images of future needs and inventions. If a set of requirements has been abstracted, the TRIZ methods (Russian for theory of technical invention) for generating ideas on the basis of known principles have been shown to be effective [6]. The record of ideas, comments and scenarios should be recorded so that it can be published and re-used easily, that can be an argument for computer based infrastructure as long as it does not hinder creativity. On evaluation and prognosis, several variants should be maintained at this stage of the innovation process. The collective project planning serves as a road map for the team and the individual subtasks. As a kick-off meeting or a convention determines very much of the emotional and motivational basis for the whole project, the reflection should not only focus on the task but also on the team process and way the meeting was structured and supported.

4 Empirical results

Data were collected from different types of meetings during innovation processes in mechanical engineering and design of consumer goods. The one closest to traditional research is a presentation and discussion among six engineers about a new technical concept for cleaning conveyor belts (133 min., 1147 interactions). Two teams focussed on co-ordination and review of possible new products with four to five people from engineering, marketing and management involved (82 to100 min. recorded, with 325 to 571 interactions). Also three long-term meetings on early concepts of two days up to a whole week and 12 to 16 participants from different disciplines (mechanical and software engineering, marketing, design, management) were observed. One meeting focussed on strategic decisions of future markets and products of two recently merged manufacturing companies. In the second, a R&D team spent three days in closure to conceptualise a completely new product generation of precision instruments with new business impacts for the company. The third meeting was a project week organised by one of our industrial partners with two groups of students from different design areas rethinking furniture for young people. Critical situations from these meetings were included in the detailed analysis (between 30 and 106 min, 94 to157 interacts). All teams were male only except for the student group and in one of the co-ordination meetings.

4.1 Distribution of actions

Two thirds of the interactions were devoted to the primary tasks. The importance of internal organisation and monitoring the team process was neglected even in the cases where an external coach was present.

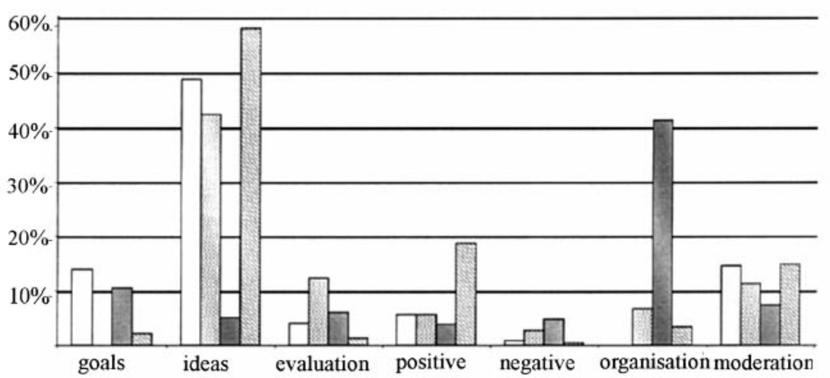


Figure 3. Distribution of categories over percentage of time in four different meetings

As Figure 4 shows, most of the time was spend on idea generation except for one meeting that focussed on organisation. The groups tended to focus on the primary task only. Positive comments outweighed the negative. The conceptualisation was in most cases prepared individually and then shared in the team. The social organisation was generally neglected even when external support was available. Big size visualisation helped to gather a common understanding. Yet in most cases, the structuring and coaching of early phases could have been substantially improved.

4.2 Analysis of patterns

The patterns of activity varied according to the purpose and the quality of the meeting. In the meeting shown in Figure 4, goal setting and assessment of the needs have been determined before.

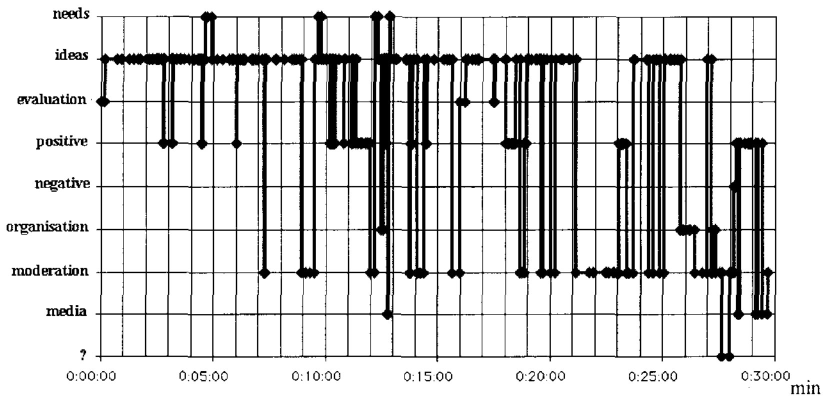


Figure 4. Pattern of activity over time of engineers and designers searching for a technical solution

Most of the time was devoted to idea generation with positive comments from the team. One person took the role of the coach who structures and summarises the discussion.

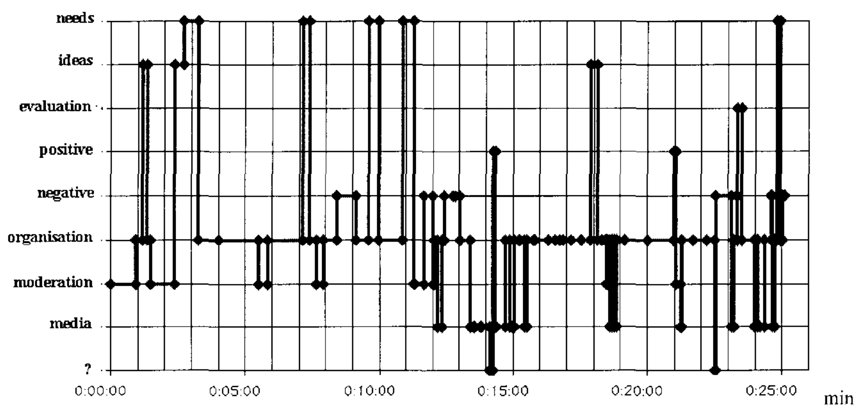


Figure 5. Pattern of activity over time in a co-ordination meeting

The meeting in Figure 5 focussed on co-ordination and decision-making on design suggestions worked out before. Very few ideas were generated, only some needs were clarified. This meeting is also an example that even a few negative comments are perceive as indicating a conflicting atmosphere whereas a lot more positive interaction is needed to give the impression of harmony and fun. Given our theoretical interest in cyclical patterns of action, we searched the data but were not able to reproduce straightforward cycles as the theoretical model assumes, and the laboratory experiments had shown. The main activity was idea generation: goal setting and decision making were hardly ever observed.

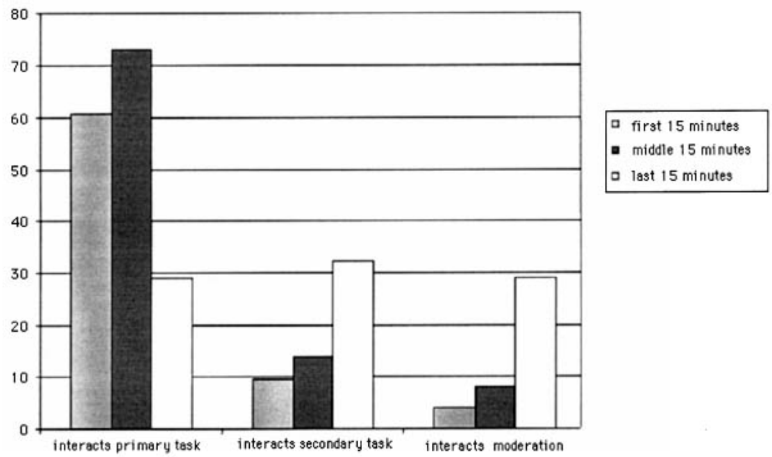


Figure 6. Differences in primary and secondary task, co-ordination and use if media in the first, medium and last 15 minutes in the technical solution meeting

This can be interpreted as a sign a poor performance since in the laboratory experiment only the successful teams showed the complete action cycle. However, another reason could be that in long-term projects goal setting and decision making may well be done at other times and even by other people than the idea generation. To discover complete cycles, one probably needs to do longitudinal observations across a whole project.

5 Suggestions for support

The optimal support for early phases is a combination of team coaching and visual stimuli that allow for individual and collective input. In the best practices, media were used a lot to visualise ideas, and empty spaces for new ideas were always present. Rough sketches on flipcharts or notepads accompany the cognitive process of fabricating ideas and remain open for modifications by other team members. The Moderation method [4] proved useful for abstract question to be answered in words since it allows for parallel input and helps to focus a discussion. Product development, however, very quickly turns to drawings, which are difficult to place on the cards. In most cases that we observed there was very little use of computer support. If early phases are to be supported by technological environment, the input devices should allow for sketching as well as writing. It should be easy to use and intuitive so that is does not disturb the creative process. Another important requirement is flexible seating to allow for plenary session as well as for smaller groups and not impose a frontal, lecture-style arrangement. The environment should foster focussing and concentration which means that it is neutral but can be furnished with stimulating material. Very helpful was a public protocol. Elements like the agenda and the goal of the meeting should remain visible throughout the meeting. A prototype for a technological environment for innovation is described in [7].

From the meetings that did not work substantially well, it can be concluded that lack of organisational support or unresolved conflicts in the organisation hinder innovation even if an optimal infrastructure is present. In these cases, intervention on the use of media was counter-productive and facilitating focussed on irrelevant aspects of the primary task instead of resolving problems with the secondary task. If this is the case, the best approach for coaching would be to address the difficulties and let the group decide what can be dealt with internally and how strategic support can be gained. Attempts to ignore the organisational context and just proceed with the primary task of generating new products are according to our evidence bound to fail as the minds of the participants will be occupied with the conflict and will not be free to be creative.

Acknowledgements

This research was founded by the Swiss Federal Commission for Technology & Innovation in Bern under KTI 4457.1 PMS. We thank all participants of the observed meetings for the opportunity to analyse and videotape their collective innovation process.

References

[1] Frankenberger, E.& Badke-Schaub, P., Integration of Group, Individual and External Influences in the Design Process, In E. Frankenberger, P. Badke-Schaub & H. Birkhofer (eds.), Designers. The Key to Successful Product Development (p. 149 - 164). London: Springer, 1998.

[2] Clark, K.B. & Wheelwright, S.C. Managing new product and process development. Text and cases. New York: Free Press. 1993.

[3] Lauche, K., Ehbets Müller, R. & Grote, G., Collective Conceptualisation in Early Phases of Innovation Processes. Proc.of Design 2000, Dubrovnik, 119-124, 2000.

[4] Klebert, K., Schrader, E. & Straub, W.G. Winning group results. Techniques for guiding gruop thought and decision-making processes with the Moderation Method. Hamburg: Windmühle, 2000.

[5] Held, J.: Fitting the mental model – A new technique for computerised event recording. In D. Harris: Engineering Psychology and Cognitive Ergonomics. London: Ashgate, pp.293-300, 1999.

[6] Amft, M., Demers, M. & Wulf, J. TRIZ in Group Design. Proc. of ICED '99, p. 1695-1698, 1999.

[7] Kunz, A., Müller, S., Kennel, T., Lauche, K., and Mybiti, K. Supporting teamwork by new means of the information technology. Proc. of ICED 01. 2001.

Dr. Kristina Lauche
Department of Psychology
University of Aberdeen
Aberdeen AB 24 2UB
Scotland UK
Tel. +44 (0) 1224 272280
Fax +44 (0) 1224 273211
k.lauche@abdn.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

COLLABORATIVE DESIGN – EXPLORING NEW INTERFACE TECHNOLOGIES

K Lauche and K Höger

Keywords: Co-operative design, computer supported co-operative work, new interface for design, design education

1 Introduction

Even though complex design and planning tasks involve many different participants who need to co-operate at certain stages, most of the current design tools focus on the individual. Members of a team are separated between the screens of computers. Shared databases for CAD systems allow for effective division of labour. However, a CAD system on a normal screen is not very useful for group decision making or discussions with lay people. Not everyone can see the screen, and only one person at a time can interact by keyboard or mouse.

Recent developments in Human Computer Interaction and Virtual Reality offer new options for supporting collaborative design. Ideas can be visualised and discussed more easily, even simultaneously, with the help of physical multi-user interfaces. Improved usability of such new tools and new HCI concepts make computer based work more intuitive. For design science, these new tools can be of practical interest to enhance teamwork in industrial projects or to train humans in social skills for co-operative work. They can also be used to generate a scenario to study collective design processes.

In this paper we describe how one of these tools, the Build-It system developed at ETH Zurich [1], was used for a design competition among architecture students and in a tutorial on Virtual Reality in mechanical engineering. The philosophy and features of the system are characterised in comparison to other developments. From the description of the two tasks it can be seen how a scenario can be set up to introduce a new technology in design education. The results are based on the observed collective design and planning processes and the usability of the system.

2 The Build-it System: A Co-located Multi-User Environment

A wide range of CSCW tools have been developed to use information technology for productive activities that are carried out by groups or ensembles of people. At first, the mere option of bridging the gap in time and space to have distributed, asynchronous communication was stunning. Databases and applications could be shared either by extending individual applications or by specific multi-user designs. In design education, CSCW tools have mainly been used to relate a group of students to the same object, e.g. a database for a project, and to allow for remote access for lectures. However, it was realised that many of the options were not used as remote participants felt left out [2] or it took too much effort to start a shared session. Greenberg & Roseman [3] also pointed out that many groupware technologies were designed to handle only some limited collaborative activity in the time/space continuum. Since then, research has turned to the nature of co-operative work forms and current work practices and on the way information technology can change, augment, and support co-operative work. Suggestions have been made how to generate working spaces for groups by shared whiteboards or chat rooms and how to balance presence on video screens with privacy.

The Build-it system used in this study is different from most CSCW applications as it started as a tool for collocated, physical interaction and manipulation of virtual objects. Though it can be combined with videoconferencing and a distributed version is planned, the focus is on natural communication and collaboration on the spot. The two main goals were to allow for several users dealing with the same objects and to implement an intuitive human-computer interaction open to all sorts of user groups. People of different computer literacy can now jointly work on one task. The basic means of computer interface are physical bricks with a retro-reflective surface that is recognised by an infrared-sensible camera. These bricks replace mouse and keyboard. The interaction takes place on a large scale projection on a table thereby unifying action space and perception space. When a brick is placed on an object, it will be activated and a yellow frame appears.

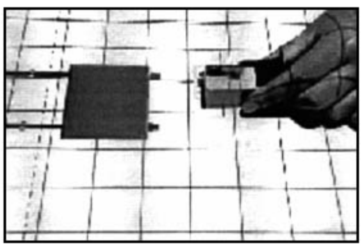


Figure 1. Activating objects through tangible bricks

It can then be modified or moved about - the virtual object will follow the movements of the physical one by a continuous update. This is called augmented reality as it enriches the physical environment with virtual elements. De-selection is done by covering the retro-reflective surface so that the bricks becomes undetectable to the camera, and removing it. This basic principle is highly intuitive, almost play-like, and has been grasped by anyone of different user groups within a minute. For more advanced use, different functions have been integrated, and the selection of objects can be adjusted to the task in question. Adding new objects by generating VRML files is usually a distributed, asynchronous task. The output may also be used at other times and places as any of the common CAD formats. Until several group sites have been realised, the main interaction is face to face.

The benefit for collaboration is the interaction via several devices on one big playground. Instead of everybody facing an individual monitor, groups members can see what the others do and comment on it. Computing is done in the background.

Figure 2. Interaction at the Build-It system in a group seated around a table

As shown in fig. 2, the users have two up-to-date simultaneous views of their scene: the plan view projected onto the table and the perspective view projected onto a vertical screen near the table. A virtual camera allows the users to choose the position from which the views are to be displayed. Next to the common workspace, two virtual libraries are projected: one for the composition elements and one for the navigation and manipulation tools.

The setting is similar to natural communication except that the result of the discussion is a digital representation, not just the sketch. Alterations can be done while relevant people are presented, and the results are immediately visualised for discussion. This was seen as the main benefit by industrial partners involved in the design of the system as it would speed up the process of inquiring client's needs, generating possible concepts, proposing and changing offers. Task analysis was carried out and requirements were gathered for factory planning, urban planning, interior design, architecture and machine configuration [5].

The possibility of interacting in a shared action and perception space is an important pre-requisite for collective objectification [6]. Artefacts or objectifications serve as an exteriori-sation of mental design processes. Individual ideas and experiences are made explicit by transferring them into material (or virtual) form and can thereby stimulate and support collaboration and understanding in a team. If a group works at the Build-It system, they share and mutually exchange their understanding of the task and their ideas for solution. A common artefact is created to which they can refer to in the later design process.

Similar systems, also involving augmented reality, were developed by other research groups: Ishii and Underkoffler from MIT Media Lab developed an Urban Planning Workbench (URP) [3], which is based on an I/O bulb with a camera pointing down on an ordinary table. The camera tracks the changing positions of physical architectural models placed on the table. Different features such as shadow, reflection, distance measurements and wind effects are computed and projected on the table. At Cornell University, Greenberg & Piccolotto develop an Interaction Table for architectural design that works with a transparent digitising surface that records pen movements by one or multiple designers. In comparison to the bricks in the Build-It system, the pen allows the users to perform more precise actions, such as required for sketching and drawing. The Envisionment and Discovery Collaboratory (EDC) by Aries, Eden, Fischer, Gorman and Scharff at the University of Colorado at Boulder uses com-putational whiteboards and specialised interaction handles with process-driven functionality [4].

3 Application in design education

This study was part of an interdisciplinary teaching network at ETH Zurich to support lecturers in applying new technologies for specified teaching purposes. Build-It serves as an example. Lecturers can exchange their experience. The benefit for both teachers and students is formally evaluated. We assumed that the Build-It system would stimulate interaction by its intuitive handling. It should encourage groups to explore a scenario and try out different solutions. The possibility of simultaneous use of several interaction bricks gives equal chances to all members to interact and should thereby increase the rate of participation. The Build-It system was therefore employed to train students in social skills for teamwork while working on a design task. We were also interested in a formative evaluation of the system to improve its usability. Two scenarios have been tested so far: an abstract design competition with architecture students and a VR tutorial in mechanical engineering.

3.1 Design Competition – Creating Spatial Compositions

The first scenario was a design competition directed at architecture students [7]. In the first phase, the participants constructed three-dimensional building blocks for Build-It in the VRML (Virtual Reality Modelling Language) format. For modelling, they could use any conventional CAD programme. The task was to design objects that could become components of an abstract composition. In the second phase, the participants formed nine teams of three to five people and together created the composition during a two-hour Build-It session. The teams could use all elements submitted by any participant as they were taken into the virtual library of the Build-It system. The competition was set up as a pure design and composition task. The teams were asked to work out their own theme and to elaborate it within their composition.

Figure 3. Architecture students designing an abstract composition

At the beginning of the sessions, each team was introduced to the system. All participants learned to handle the interface within minutes. The interaction was videotaped and recorded on protocol sheet for the type of action (exploration, planning, building, evaluation), the type of decision making (consensual versus dominant) and the degree of equal participation. After the Build-It session, the teams titled their composition and answered a usability questionnaire. A jury with experts from architecture, industrial design, multimedia, and interactive visual computing evaluated the compositions interactively on the system. The results were compared on the level of abstract composition. The implementation or development of design principles, their readability and interplay in the overall composition as well as their adequacy in regard to the requirements of the Build-It system were decisive in the ranking.

3.2 VR Tutorial – Planning of Production Systems

The second study was a tutorial on virtual reality (VR) in mechanical engineering. The main goal was to enable students to differentiate how various VR-technologies can be applied in the planning of production systems. They were taught the major principles about generation of data, transfer to visualisation tools and interaction with complex data in a lecture. The scenario for the tutorial was to work out a layout for a packing production system in a given factory. The instruction specified details of capacity or safety regulations. Compared to the design competition, the second task contained much more external requirements that were known before and explicit in the instruction.

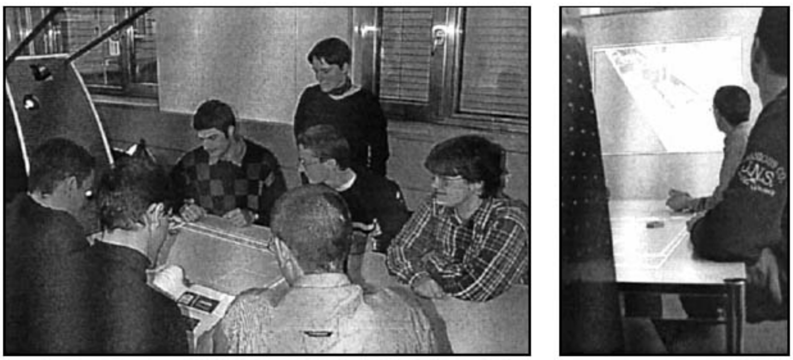


Figure 4. Engineering students planning a virtual factory layout

Four groups of six students each spent 30 minutes working with the Build-It system. They answered a test on VR knowledge before and after the tutorial and a usability questionnaire afterwards. The interaction was also videotaped.

4 Results

To evaluate the use of a common design platform, the actions and interactions of the participants were observed and recorded during the Build-It sessions and classified for the design approach and teamwork. Learning outcomes as well as usability were assessed by a questionnaire. The test results of the VR tutorial showed that most of the students acquired an understanding about the option and use of VR in the product development process. They could describe the use of a system like Build-It, name the difference between solids and surface models and list limitations of VR. However, the inter-individual variance was greater than the one between pre- and post-test. Those who answered right on the knowledge test did so after the lecture and before the tutorial. Nonetheless, the lectures saw the benefit that the students had actually experienced the different kind of data and user interfaces. For the architecture students, the benefit was seen in the experience of teamwork and in the combination of goal-oriented and explorative elements.

4.1 Observations on Design Approach

In the design competition, most groups started by exploring the system and the objects submitted by other teams. After some time of mainly playing, the participants would bring up the question what they intended to create. Ideas were discussed and tried out. Some groups fixed a goal and strategy in the beginning and spent the rest of the session carrying out their plan. For example, one group modified and placed objects to form a subway entrance. Another group constructed a field of multiple reproductions of one abstract object. These groups sometimes encountered difficulties with the system's functionality and had to readjust their goals. Others left more room for arbitrary configurations and saw what happened if everybody interacted at one end. The virtual camera was used to check the design result by changing the view and the focus.

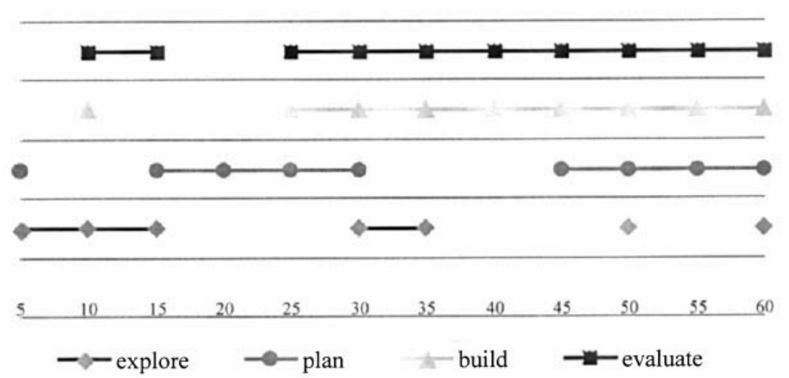


Figure 5. Action pattern over time in one of the observed groups

Given the complexity of the task, the students in the VR tutorial could only work out parts of a solution in the given time. As their goal was defined in the instruction, they only had to recall and understand it. Exploring the system coincided with exploring the factory set up. Most of the groups went immediately into the task and used Build-It as a ready-made tool. On evaluation, the planning result was checked against the requirements in the instruction.

4.2 Results on Teamwork

The hypothesis that Build-It would encourage participation was proofed: 80 % of the time all participants were active. The evidence that the system would foster a far more democratic way of decision making is not as clear. In the observation, 62% of all decision were classified as consensus oriented and only 32% as dominated by one person. If one member articulated critical questions, this was considered in 80% of the cases.

The team who won the first price worked in a consensus-driven co-operative manner. They had discussed their design ideas prior to the session and integrated diverse design ideas of all team members into a coherent composition. However, the winners of the second price, who convinced the jury through their clear, congruent design, worked on a dominant strategy of the most experienced person. A few teams also quickly defined a division of labour in handling certain tools or carrying out reproductive steps of the design. Some participants suggested separate screens for subtasks. In these cases, the artistic quality of the design was given priority to the unified action space for co-operation. Following expert decisions may be sensible in some cases but will also miss out important contributions from people with less or different expertise such as production worker in factory planning. In these cases, the Build-It system does not level the differences in experience or status but it offers the technological option for equal status.

4.3 Usability

The usability scores were satisfying for the scales on control, conformity to expectations, ease of learning and suitability for teams. The scores for task appropriateness, for self-description and for options for individual adaptation received only medium satisfaction. Very low scores were given for tolerance to errors, which may relate to the lack of stability of the system. The students enjoyed the parallel interaction and the intuitive handling. The benefit was seen for discussing prototypes and recognising errors in an early stages. Some liked the element of chance, others complained that they couldn't exactly carry out what they had intended. Even though the handling of the third dimension was a bit awkward, the students liked the simultaneous use of 2D and 3D and thought it helped to get a good spatial understanding. The general tone of critique was that the Build-it system was a good concept, but not fully developed. Participants complained about the speed and some malfunctions of the system. The more experienced the participants were with CAD the more they lacked certain options of conventional programmes. On comparing the two types of system, they missed the precision of CAD and the opportunity to create new objects. Limitations were seen in the computing capacity and in the collision with other members on parallel interaction.

5 Conclusions on New Technologies in Design Education

The Build-It competition and tutorial provided an opportunity to employ new ways of collaboration on computers in design education. The students and lecturers were introduced to new mediations to augment reality, which both enjoyed to explore. For design research, it offered a task oriented setting for observation of collective design. For design education, the study can serve as an example how recent IT development can be applied for instructive purposes. Since the system is easy to learn, the students could focus on their task and develop their domain skills as well as practising social skills for teamwork and getting to known a new tool. In order to be a real skills training, however, the sessions should be extended. The comparison of the two scenarios shows that an extended exposure to the system is favourable if the students shall be trained in social skills and not only get a rough idea about technological options.

This pilot study also showed that careful preparation is required for the use of new technology and tools in design education. A close co-operation between lecturers and technical support was required to establish the scenarios. A tool which is still being further developed may require unforeseen changes for a particular scenario. As is this case, where only one system was available, flexibility in the timetable was required or alternative lectures and tutorials had to be set up for the rest of the class. These have to be carefully design as well so that they do not spoil the benefits of the tool seen as most attractive. The set-up of this study required the lecturers to specify their educational goals for the session in much more detail than they would have done themselves. It helped to ensure that no only the system worked but that it was also beneficial to the course. Support for the conscious use and evaluation of new technology in design education should involve expertise in higher education methods as well as acquaintance with the technological options available.

Acknowledgements

This research has been founded by a grant from the ETH Swiss Federal Institute of Technology for improving university teaching. The photos were taken by Peter Troxler (1+2), Kerstin Höger (3) and Andreas Kunz (4).

References

[1] Rauterberg, M., Fjeld, M., Krueger, H., Bichsel, M., Leonhard, U., Meier, M., BUILD-IT: A computer Vision based Interaction Technique for a planning Tool. Proc. HCI '97, Berlin: Springer, 1997, p. 303-314.

[2] Greenberg, S., Roseman, M., Using a Room Metaphor to Ease Transitions in Groupware. Research Report 98/611/02, Department of Computer Science, University of Calgary, 1998.

[3] Underkoffler, J. and Ishii, H. URP: A Luminious-Tangible Workbench for Urban Planning and Design, CHI, 1999.

[4] Arias, E., Eden, H., Fischer, G., Gorman, A. and Scharff, E. Transcending the Individual Human Mind. Creating Shared Understanding through Collaborative Design. ACM Transaction on Computer-Human Interaction, 7, 2000,p. 84-113.

[5] ISO 13407, Human centred design processes for interactive systems, 1999.

[6] Fjeld, F., Lauche, K., Bichsel, M., Voorhorst, F., Krueger, H. and Rauterberg, M. Physical and Virtual Tools: Activity Theory Applied to the Design of Groupware, CSCW, 2001.

[7] Build-It Teaching Network: http://caad.arch.-ethz.ch/research/buildit_tn

[8] Höger, K. Build-It. In Engeli, M., Bits and spaces. Architecture and Computing for Physical, Virtual, Hybrid Realms, Basel: Birkhäuser, 2001, p. 26-31.

Dr. Kristina Lauche
Department of Psychology
University of Aberdeen
Aberdeen AB 24 2UB
Scotland UK
Tel. +44 (0) 1224 272280
Fax +44 (0) 1224 273211
k.lauche@abdn.ac.uk

Kerstin Höger
Chair for Architecture and CAAD
ETH Swiss Federal Institute of Technology
CH-8093 Zürich-Hönggerberg
Switzerland
Tel. +41 (0) 1 6332997
Fax +41 (0) 1 6331050
hoeger@arch.ethz.ch

Product and System Modelling

Including sub-sections:

Product Data Management
Sketch/Vague Models
Functional Modelling
CAD
Features
Virtual Reality
Integration

This page intentionally left blank

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

VISUALIZATION OF PRODUCT STRUCTURE AND PRODUCT ARCHITECTURE FOR A COMPLEX PRODUCT IN A MASS CUSTOMIZATION COMPANY

T K P Holmqvist

Keywords: product structuring, product architecture, design representation

1 Introduction

A fundamental challenge in developing a new product or process is to combine engineering detail into a coherent whole [17]. Besides a well-designed product development process and clear goals for the product development, knowledge about the product structure, both in detail and for the whole, is needed. This issue is of particular importance for mass customisation companies, as they are to make products with an almost infinite variety while still achieving component economy of scale and rapid response to market demands [11]. To be able to do so efficiently, the participants in the product development process must be able to handle the relations within the product. Several researchers and practitioners have studied ways to describe the relations within the product (see e.g. [4], [12], [9]). Different models and tools for visualisation and management of product architecture and product structure have been proposed (see e.g. [15], [2], [5], [4], [9]). However, for complex products the lack of tools for design support remains a problem. Product complexity refers, in this paper, to products with a large number of variants, large physical differences between the variants, several technologies and several different functions, for example cars or trucks.

The overall research question in this paper is how to visually represent the product structure and the product architecture for a complex product during the product development process. The question has two constraints: firstly, that the visualisation of the product should also visualise the interfaces between the product modules; secondly, that it must be possible to compare different interface selections by allocating costs to the different alternatives. The paper presents a model for visualisation of the product architecture for complex products, and concludes with managerial implications for how this model should be used in order to increase the product development efficiency and to enhance the product modularity. The empirical data have been collected during a research project carried out in collaboration with Volvo Truck Corporation (VTC) to explore product visualisation and modularity.

2 Frame of reference

Two of the main issues in mass customisation are the variety and variability of the product. Variety refers, in this paper, to when two chunks differ from each other either physically or functionally and variability to when a product with the same content can be configured in at least two different ways (colours are excluded). At the same time variants and variability are necessary for profitability, as well as incurring a lot of costs for the companies, both direct and indirect costs [see e.g. 13].

Visualisation of the product in a general manner has interested several researchers over the years. It makes an overview possible and facilitates the product development work. One of the common ways to describe the product is to decompose it into smaller parts and arrange them in a hierarchical order, in this paper referred to as product structure. Hubka & Eder [4], Suh [12] and Pahl & Beitz [9] have carried out fundamental research and presented models in this area. They use functional requirements for decomposition of the product. The hierarchy is mainly represented with a tree, where the end product is placed at the top and the functions are placed on the different levels and connected to elements above and below. Function is commonly used for decomposition, but other parameters can be used for arranging the product hierarchy. Hubka & Eder [4] for example use other parameters, e.g. complexity or type of operand, to classify and structure the product in order to enhance the overview.

The increased variety has also created an interest in product architecture (PA), which Ulrich [14] discusses and defines as: (1) the arrangement of functional elements, (2) the mapping from functional elements to physical components, and (3) the specification of the interfaces among interacting physical components. The same author stresses that the possibility of creating variety resides not with the manufacturing system, but with the PA. Other authors (see e.g. [10] or [2]) argue that both product performance and productivity are determined when the product architecture is designed. Modular is a type of PA, defined by [14] as a one-to-one relation between function and physical part. It has attracted much interest, as it offers the possibility to create product variety in a short time with a small number of parts. Since the design of PA is a crucial point in the development process, there exist several methods for designing the PA (see e.g. [14], [5]). However the PA can be used for other purposes too. For example Pimmler & Eppinger [10] suggest a method for reducing the design complexity and for organising the product development work by combining the decomposition and the product architecture.

The PA is to a large extent discussed for single products, and the variety problems are thereby neglected. This issue is addressed by a concept that is called the platform concept (see e.g. [6], [8]), which makes it possible to increase both the variety and the commonalities. The concept is powerful and has been adopted by several companies globally. The platform concept is, however, mostly discussed on a strategic level, which relates more to utilisation of investments and technology than to design methods. The concept is often discussed in connection with star cases of successful companies. What tools have been used to support the product development process and the designers is not studied further.

Product family is another conception that is relevant for product structuring in mass customisation companies. The products that share a common platform but that have specific features and functionality, required by different sets of customers, are a product family [7]. Moreover, the product family has been used in combination with the three previously mentioned concepts: variety, product structure and PA. Erens & Verhulst [2] discuss the PA for product families by decomposing the product. They stress that a product can be visualised by using three independent domains, namely the functional, the technical and the physical. Each domain describes the whole product but from different viewpoints. Hence, the real product is a combination of the tree domains. Even if the domains are independent the decomposition of the product is an iterative zigzag patterned process of decomposition and allocation. Each domain also has its own architecture, which means that the PA is a combination of the architecture from each domain. Fujita et al. [3] and Jaio et al. [5] use the same approach with three different domains, in combination with a tree structure to visualise the structure in each domain. Both authors ([3]&[5]) also propose a methodology for mathematically optimising a product family with modular PA in new product development.

Each of these three methods ([2][3][5]) discusses the relation between the different domains. However, neither of them proposes a model for how both the architecture and the relations can be visualised.

Several authors have discussed relations between different design elements. Pahl & Beitz [9] discuss relations that contribute to the function, such as flow of energy, material and signals between design elements. Ulrich & Tung [16] discusses incidental relations between modules, which are interactions not wanted due to not contributing to the function.

3 Research question and methodology

Mass customisation companies have to manage both variety and variability. These are one aspect of what this paper refers to as a complex product. Both variety and variability have a great impact on the efficiency of the product development and the manufacturing process (see e.g. [13]). Strategies for managing the mass customisation dilemma, such as product platforms and product families, have attracted much attention from both researchers and practitioners. These strategies are powerful but have no tool for visualising the product structure and PA. Models for visualisation of how different elements relate to each other in the product have been constructed in the two areas of PA and product structure. The latter discusses visualisation of the product but only the hierarchical dependence and does not consider all the dependences between the different domains. PA models, on the other hand, address this question but no model for visualisation of the PA is developed. Since the design of PA is important for mass customisation companies that strive towards a modular product, the visualisation of the PA can be a tool to facilitate this transition and to increase the product development efficiency. Thus, the aim here is to answer the research question of how to visually represent the product structure and the product architecture for a complex product during the product development process.

The empirical data underlying this paper derive from a case study that was performed from mid-1999 to mid-2000 at and together with the Volvo Truck Corporation (VTC). The researcher spent one or two working days per week, on average, in the VTC organisation during this nine-month period. The choice of the case study method was natural since the case study is preferable if the boundaries between the phenomenon and the context are not clearly evident, as is true in this case [18]. Here it was necessary to study the product itself, the dependence within the product and the reasons for the dependence, which to a great extent are found in the organisation at VTC.

The case study was performed in three steps. The results of each step as well as the whole case study are the outcome of an iteration process between theory and empiric that is described in Dubois & Gadde [1]. Firstly, the present product structure was compared to the theory about product-structuring methods, in order to learn whether the lack of support from the system was due to unsatisfactory decomposition of the product. Secondly, one chunk of the VTC product was structured by a method of Hubka & Eder [6, p. 98]. This method was chosen among several others since it structures the product by using complexity, which is one major issue for a mass customisation company such as VTC. Third, the results and knowledge from the previous steps, together with theory from the frame of reference, were used for improving the visualisation of the product. The data were collected in two ways; by using the present design support system at VTC and by interviewing designers and managers who are involved in the product development process.

4 The case of Volvo Truck Corporation

The present design support system at VTC uses design function as decomposer of the product and is structured in hierarchical levels. The system handles information about what parts the product contains and how the different design functions can be configured. However, it is not possible to see how they depend on each other. The present product structure was compared to several product structuring models ([4], [12]) in order to discover whether the dissatisfaction was due to improper decomposition. No such relation was found, but the structure was similar to the description of the technical domain by Erens & Verhulst [2].

Since the present product structure at VTC was proper but did not visualise the product satisfactorily, another product-structuring model by Hubka & Eder [6, p. 97] was applied to the VTC product structure. The model [6, p. 98] uses function as decomposer and structures the product into hierarchical levels by using complexity. As shown in Table 1, the hierarchical levels matched the present product structure well.

Table 1. Hierarchical levels in the VTC product structure

Level of complexity	Characteristics compared to [4]	Designation at VTC
I	Elementary system	Part
II	(Realisation of the simple system)	Variant
III	Simple system	Function group
IV		Sub-module
V	Systems that perform a closed function	Module
VI	System that fulfils a number of functions	Vehicle Module

However, the designers lacked information on the relations between the different elements, so the next step was to connect the levels to one another. A top-down approach was used. Each element on each level was connected to at least one element on a lower level. The relations were easy to determine for levels III to VI, but when the function groups were connected to the variants of function groups, certain problems arose. Sometimes the relations could not be determined because information from selections in other branches of the tree was needed. For example, if the function group D is possible to connect to variant H or I, the choice may depend on the choice between variant N or O, and perhaps on several other choices too (see Figure 1, left). This means, that if the purpose was to assign the different function groups to variants, the result was satisfying, but information about how the variants relate to each other remains invisible.

The chunk of the vehicle that is focused on in this study contained three function groups as well as relations on 22 other function groups, each function group containing several variants. Consequently, before making a change in a variant, a designer firstly has to investigate which function groups are affected by the change; secondly, which variants within the function groups can be affected by the change; and finally, what changes in variants of the other functions will affect this variant. As only one chunk in the vehicle was studied in the case described above, we decided to investigate whether the same phenomenon could be found within other chunks. All function groups in the vehicle, a total of a couple of hundreds were examined. The function group that had the fewest relation to other function groups was found to have relations to 1% of the other function groups, and the group with the most relations had relations with 52% of the function groups. The implication is that visualising the product structure with a hierarchical tree structure of the product was not satisfactory because the information needed for the selection was not found in the vertical hierarchical levels above, but depended on several other selections.

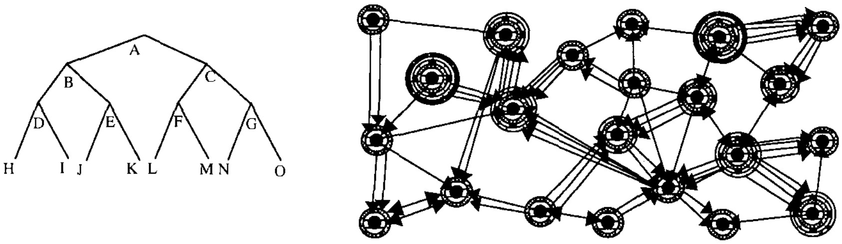


Figure1 A hierarchical tree structure (left) and a web (right).

The largest number of relations was found within the level of function groups and variants, see Table 1. Therefore, instead of using the whole product structure, we focused on the relation within these two levels. These two levels are good in a level of abstraction point of view, since number of elements is manageable. All the 22 function groups, in the studied chunk, were plotted on a piece of paper and, those function groups that had a physical relation to each other, according to the design support system, were connected with a line. This visualisation was called web instead of a tree, due to its appearance. Now it was possible to see which function groups that had a relation. However, the purpose of the study was to visualise the PA, jet visualisation was technical solutions with their physical relations and neither the variety nor the variability was visualised. Each variant are different specification of the technical solution that name the function group. Hence, there is a strong vertical connection between the function groups and the variants. But which specification that should be chosen depends on selections of other function groups. For example which variant that should be selected in the shock absorber function group are dependent on the selection of suspension type. Since each variant is unique is each variant specified in a bill of material. All variants in each of the 22 studied function groups were drawn as circles around the function groups, as the in drawing previously described. The relations between the function groups could now be specified with the relations between the different variants instead of each function group. This was necessary since not all variants in a function group have the same relation to each other.

However, even if the relations between the different variants were visualised, the designers wanted information about the types of relations. Three types were found evident: functional, physical, and both functional and physical. The functional relation occurs when at least two function groups are used, or are using the same function, but do not have a physical interface to each other. For example, a truck's generator and battery support each other but do not have a direct physical interface. The physical relation occurs when two function groups have a physical relation but the functions are independent. For example, two function groups may share a bracket but they need not be functionally related. Both the physical and the functional relation occur when the two preceding ones are combined, for example in the engine and the gearbox. Since not the incidental interactions Ulrich & Tung [16] were not considered, we found it more proper to use dependences to denote the relations between the variants, because the relations that are visualised are necessary for the product functionality. These dependences were drawn between the function groups and variants, whereby the visualisation became more satisfactory than before. But then a further obstacle arose: when the product has large physical variety and one chunk's physical volume varies greatly, other chunks or components will be affected even if they do not have interfaces to the varying chunk. For instance, the cab on a truck has no interface to the chassis packages, but these are affected by the cab length. This phenomenon was called artificial dependence. It is artificial since the dependence is not needed for the product's overall function, nor is there an interface between the two chunks. An example of a visualisation can be seen in Figure 1 to the right.

This visualisation was introduced to the designers and they found it were useful. The drawing was then compared to the definition of PA by Ulrich [14]. The function groups and the variants are technical solutions that contribute to the overall function. This in combination with dependences, which visualises which function groups that are collaborating to perform a function and how they relate to each other. Hence, part one of the definition is fulfilled. The part two of the definition is the mapping of functional elements to the physical components. This part is fulfilled by the fact that the variants are unique and that each has a bill of material assigned to it, whereby it is specified physically. The physical dependences also contribute to fulfilment of the definition, namely the third part. Since they specify which elements are connected physically to each other they can also be seen as the physical interfaces between the interacting physical components. Hereby, all three parts of the definition are fulfilled, and the result can be seen in Figure 1. The variety of the product is also visualised since the variants define the variety range and the variability. This is visualised in the web, as there will not be any dependence between them, if two variants can not be used together.

The two restrictions with regard to the purpose are also fulfilled. The interfaces between different product modules can be visualised by drawing a line that surrounds those function groups that form a module. The dependences that are crossed by this line are also a visualisation of the interface characteristics. The second restriction is then easy to fulfil since the variants are physical parts, and all the costs for development, purchasing and production are possible to calculate.

The type of dependence is not the only factor determining how two function groups will affect each other. The dependence can have a direction as well. For example, the engine and the water pump have a mutual dependence, and a redesign of the engine may change the design of the fuel pump, but the converse is not true. This was incorporated in the web by using an arrowhead to symbolise the direction of change in each dependence, regardless of type. The web is thus an aid to be employed by mass customisation companies to employ during the product development work. It visualises PA and shows what characteristics the dependences within the product have. It also contributes to the visualisation of the product structure.

5 Creation and management of the web

The previous section has described the development of the web and what parameters it visualises. This section will explain how the web should be created and how it can best be used as an aid during the product development process.

First, decompose the product in the technical solution domain. Second, create a hierarchy of the different functions. For both the first and second stages, product structuring models (see e.g. [4], [12], [8]) can be very useful either for reference or as aids to structuring. Third, find where in the hierarchy either a lot of dependences between the levels can be found or the level of abstraction is such that the visualisation is supportive and not too inane or too comprehensive. Fourth, create the visualisation by drawing simple systems (in the VTC case called function groups) as dots and their variants as circles around them. Then connect the different variants with each other, according to what type of dependence exists between them. The types of dependence are physical, functional, both functional and physical, or artificial (see the last section for more information about the dependences). Next, draw arrowheads in the directions of possible changes that may affect other simple systems. Then compare to the definition of PA by Ulrich [14].

The web can be used for visualising the interfaces between modules in the product since the web visualises the PA and modules are one type of PA. An interface can then be visualised by drawing a line, between the dots in the web that crosses the dependence lines. The lines that symbolise the interfaces will help the designers to see the type of interfaces that the module has. It can also be used when selecting the interfaces for a new variant or function group. Since both the function and the physical parts are visualised, different alternative interfaces can be evaluated both functionally and economically. In addition, the interface line visualisation can be used when striving for a more modular PA. The fewer dependences an interface line crosses, the more modular is the architecture. Since the variants can be translated to physical parts the cost for the module can be calculated and different alternatives can be evaluated.

The web can also be used to foresee the effect of changes in the product. By using the web to employ the dependences between the variants, the arrowheads on the dependence line show whether a change in one variant may affect any of the other variants in the function group or in any other function group. Hence, the chain reactions due to coupled interfaces can be predicted before the product change is initiated.

6 Conclusions

To be successful, mass customisation companies have to manage both variety and variability. The key to both of these is the PA. This paper describes a tool for visualisation of the PA. The empirical data derive from a case study performed together with the Volvo Truck Corporation during a nine-month period. In the case study one chunk of a truck was structured into a hierarchical tree structure using complexity [4]. When the product structure was examined in a PA perspective, a visualisation of the PA was found that was called the web. The web is not a substitute for a hierarchical tree structure, but a complement.

The web displays both the functions and their physical realisation, using dots for the simple systems and circles around the dots for variants of physical realisation to capture the PA. Further, these elements are interconnected according to how the functions and variants depend on each other in the product. Four types of dependences were considered: functional, physical, both functional and physical, and artificial, the last occurring when two chunks affect each other without having a mutual interface. Arrowhead/Arrowheads on each dependence can be used to visualise the direction of change between two variants. Thus, the web can be used to anticipate the consequences for all variants if a design change is decided upon. This is important for the product development efficiency in a mass customisation company since all variety causing complexity is difficult for the designers to handle.

Besides enabling visualisation of the product architecture, the web can exhibit the interfaces between modules within the product. Since all elements in the model can be found in reality it is possible to calculate the cost for different positions of interfaces within the product.

References

[1] Dubois Anna and Gadde Lars-Erik, 2001, "Systematic Combining- an abductive approach to case research", Forth coming in Journal of Business Research

[2] Erens F. and Verhulst K., "Architectures for product families", Computers in industry, No. 33, 1997, pp. 165-178.

[3] Fujita K., Skaguchi H., and Akagi S., 1999, "Product variety deployment and its optimisation under modular architecture and module communalisation", Proc. ASME Design Engineering Technical Conferences, Sept. 12-15, 1999, Las Vegas, Nevada.

[4] Hubka V. and Eder W. E., "Theory of Technical Systems: a Total Concept Theory for Engineering Design", Springer-Verlag, Heidelberg, 1988.

[5] Jiao J. and Tseng M. M., "A methodology of developing product family architecture for mass customisation", Journal of Intelligent Manufacturing, No. 10, pp. 3-20., 1999.

[6] Meyer M. H. and Lenherd A. P., "The power of Product Platforms: building value and cost leadership", Free Press, New York, 1997.

[7] Meyer M. and Utterback J., "The Product Family and the Dynamics of Core Capability", Sloan Management Review, Spring, 1993, pp. 29-47.

[8] Muffatto M., "Introducing a platform strategy in product development", International Journal of Production Economics, Vol. 60-61, 1999, pp. 145-153.

[9] Pahl G. & Beitz W., "Engineering design: a systematic approach", Springer, Berlin, 1996.

[10] Pimmler, Thomas U. & Eppinger, Steven D., Integration analysis of product decompositions, Design Theory and Methodology, Vol. 68, 1994, pp. 343-351.

[11] Pine B. J., "Mass Customization: the New frontier in Business Competition", Harvard Business School Press, Boston, Massachusetts, 1993.

[12] Suh, "The principles of design", Oxford University Press, New York, 1990.

[13] Suzue and Kohdate, Variety Reduction Program, Productivity Press, 1990.

[14] Ulrich K. T., "The role of product architecture in the manufacturing firm", Research Policy, 1993, Vol. 25, pp. 419-440.

[15] Ulrich, K. T. and Eppinger S. D., "Product Design and Development," McGraw-Hill, New York, 1995.

[16] Ulrich, K. and Tung, K., 1991, Fundamentals of Product Modularity, Proceedings of the 1991 ASME Winter Annual Meeting Symposium on Issues in Design/Manufacturing Integration, Atlanta, GA.

[17] Wheelwright and Clark, "Revolutionizing Product Development: quantum leaps in speed, efficiency, and quality", Free Press, New York, 1992.

[18] Yin R. K., "Case study research: design and methods", Sage Publications, California, 1994.

Tobias K. P. Holmqvist
Department of Operations Management and Work Organization, Chalmers University of Technology, S-412 96 Gothenburg, Sweden, Tel: +46 31 772 51 24, Fax: +46 31 772 11 94, E-mail: tobias@mot.chalmers.se, URL: www.mot.chalmers.se/omw

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

MANAGING MECHATRONIC SIMULATION MODELS OF TECHNICAL PRODUCTS WITH PDM-SYSTEMS

M Krastel and R Anderl

Keywords: *Configuration modelling, systems modelling, product data management, mechatronics, information modelling.*

1 Introduction

Short term changes of customer's requests and the increasing globalisation of markets require a shortening of product and innovation cycles, as well as a comprehensive product configuration ability in order to react fast to customer requirements. The enterprises' effort to fulfil the individual customer demands results in an increasing amount of product variants. On the one hand, the striving for digitalisation within a virtual product development helps enterprises to shorten development times. At the same time, product complexity increases as a result of the closer cooperation of different engineering disciplines, such as mechanics, electrics, hydraulics, control and software engineering within one mechatronic product. The challenge for enterprises is to manage the increasing number of product variants and the resulting larger number of parts and functions during all phases of a multidisciplinary virtual product development.

The virtual product development, i.e. the shifting of knowledge-gathering to the computer, leads to a strongly increasing importance of simulation and computation to forecast a product's behaviour before prototype production. Figures from a Japanese automotive manufacturer show, that the consequent usage of simulation leads to a considerable time an cost advantage (see table 1, left side).

Table 1. Time and effort reduction caused by the application of simulation [1] and PDM [2]

Simulation	Reduction
Collision Number of vehicles for tests	40 %
Noise and vibration Work load for idling noise	70 %
Front end air flow Number of tests	80 %
Engine performance Engine prototype cost	15 %

PDM	Reduction
Time-to-market Time	40 %
Change management Effort reduction	90 %
Revisions Number of revisions	80 %
Throughput time Time	50 %

Particularly in the context of simulation, the supply of all sub processes with valid and up-to-date product data is indispensable. In order to reach this goal, enterprises have introduced product data management systems (PDM-systems), which allow an integration of data. A German automotive supplier reports a significant effort reduction by introducing a PDM-system in the area of aggregate development (see table 1, right side).

With the facts presented in table 1, it is obvious that the biggest amount of cost and time reduction can be achieved with the synergetic use of simulation and PDM-systems. The integration of simulation data in PDM-systems is a future challenge. First research results will be presented in this contribution.

2 Product Data Management – Principles

PDM-systems act as an integration platform for different application systems used within the product lifecycle, such as design-, 3D-CAD-, and simulation systems (FEA, MBS). The provision of consistent and up-to-date information at the right place, at the right time, for the right person in suitable quality and quantity has to be guaranteed by the PDM-system.

The last years' developments in information technology also influence the methods and tools in the area of product data management [3]. In the mid seventies, the manual, 'paper based' management and exchange of product related documents was beginning to be replaced by a digital document management. A file server based architecture was used to store and exchange digital documents. Any classification was based on the directory structure of the file system. The first product data management system approaches, called engineering data management systems (EDM-systems), were developed in the early eighties for special purposes such as drawing-, document- or bill-of-material management. With EDM, the information are stored in a database and additional functionality, such as user management, workflow management, and the management of versions and variants was implemented [4]. Today's approaches introduce a product data management based on an integrated product model that is stored in a data base in order to allow a data sharing of product data.

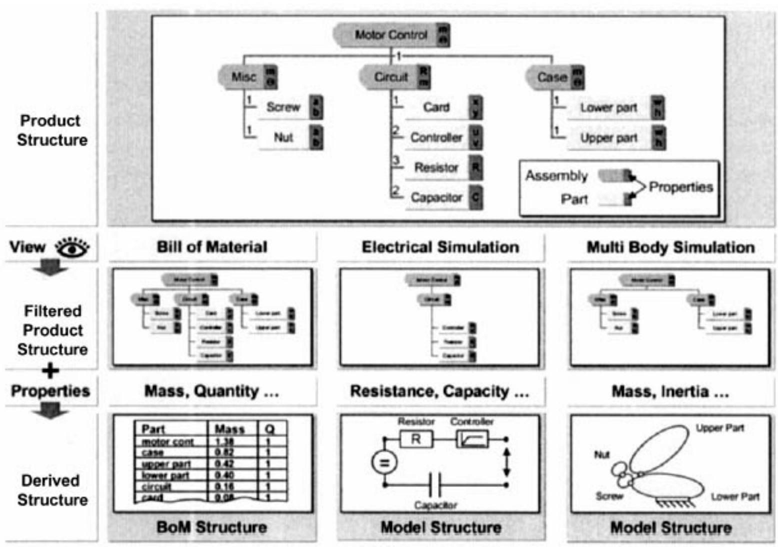


Figure 1. Product structure with views and derived structures

From a software-architectural point of view, current PDM-system developments show a future trend towards integrated system solutions with a modular architecture. These integrated systems realize the different PDM functions by the interconnection of different modules or with links to external software tools, e.g. in the areas of data-visualization (connection to different viewers) and message-functions (connection to external mail-systems). The basic PDM-system architecture is separated into two different data layers. The PDM-systems use a macro layer to store the semantics and relationships of information. Within the macro layer, the internal data model (meta data model) of the PDM-system is defined. Meta data means "data over data" and defines the relationships of information objects. Examples for meta data objects are the creator of a document, the part number of a product, or the file name of a CAD model. The physical files are managed within the second layer, the micro layer. The information stored in the meta data are then linked with the physical files.

The core of the meta data model is the product structure (see figure 1). The product structure should be the central point for the interconnection of application systems. DIN 199 defines the product structure to be 'a product presentation which represents all relationships between modules, assemblies and parts of a product'. The product structure establishes a logical connection between the product and its constituents. In order to support the design process, a special type of PDM-system was developed to integrate the CAD model structure and the PDM product structure. The so called team data management systems (TDM-systems) support the communication between the CAD- and the PDM-system. Typical functionalities of TDM-systems are: management of dependencies between CAD model constituents (assembly structure), management of CAD parametrics, management of associativity between 2D drawing and 3D model, family table management, and feature management [5].

One deficit of current product development processes lies in their lack to integrate simulation data. No comparable approach to the CAD/TDM-systems integration exists. In order to establish this integration, information such as simulation model structures, simulation model configurations, simulation experiments, and simulation results, need to be stored in the meta data model of PDM-systems. In some enterprises this problem has already been identified and first steps towards setting up a product data management supporting the virtual product development of mechatronic products have been taken [6].

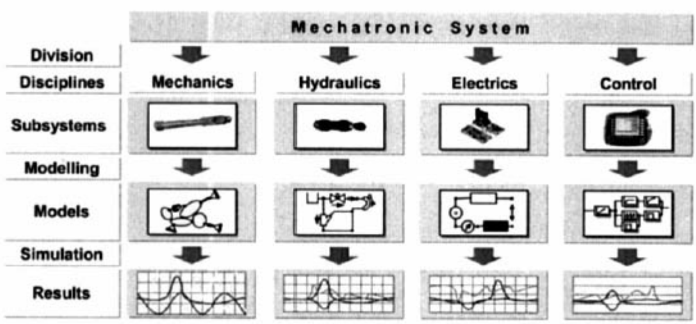


Figure 2. Mechatronic simulation

3 Mechatronic Modelling and Simulation

The essence of mechatronics is the integration of different technical disciplines into one product. In a mechatronic product development, the simulation is different from simulation of products dominated by only one engineering discipline [7]. The simulation of product functionality and behaviour has to be divided into several disciplines with specialised and optimised computation and simulation tools [8], see figure 2. This is a consequence of the fact that the functionalities of common simulation tools are usually limited performing a simulation within an individual discipline's context [9].

Basis for the interpretation, analysis and optimisation of dynamic product behaviour is the modelling, i.e. the mapping of real technical systems to computer-interpretable simulation models. The multidisciplinary character of products results in a multiplicity of specialised and optimised CAx tools for the simulation of the different disciplines. The modelling is done in the syntax of a software tool specific modelling language. Usually, each software tool has its own modelling language and is specialised in one engineering discipline [10]. Due to the multidisciplinary character, a lot of different software tools and, as a consequence, modelling languages are in use in enterprises. Examples are ADAMS, SIMPACK and SD/FAST for multibody systems, SPICE, VHDL and MAST for electronic circuits, FLOW MASTERS, AMESim and VHDL AMS for hydraulic systems, SIMULINK and SYSTEMBUILD for control systems. Since the end of the seventies, approaches for a discipline-spanning modelling of mechatronic systems were developed. In order to be able to control the complexity of such a model, the specification languages feature object-oriented characteristics. Examples are DYMOLA as well as its further development MODELICA.

A closer look at the different modelling languages shows some common characteristics. Each modelling language contains information regarding the model structure (structure model), the model behaviour (behaviour model) and the model parameterisation (parameter model), see figure 3. Despite these analogies, they differ regarding the syntax and the separation of structure and behaviour model. Some languages explicitly separate the structure from the behaviour model, e. g. VHDL [11], some languages implicitly connect the two models, e. g. SD/FAST [12].

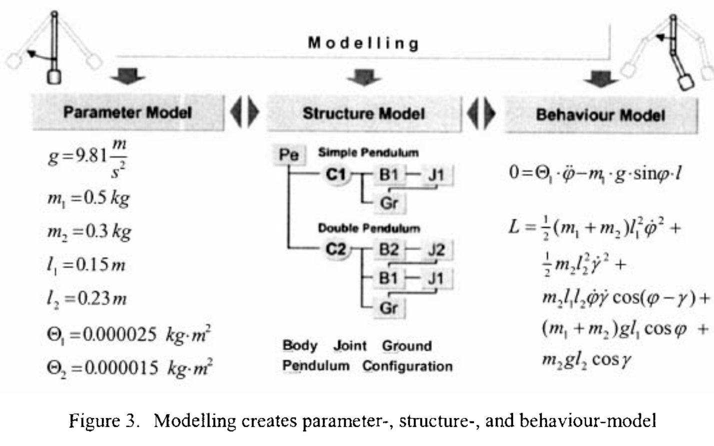


Figure 3. Modelling creates parameter-, structure-, and behaviour-model

Lets recall the above mentioned TDM-systems, which establish a link between the CAD-system model structure and the PDM-system product structure. From the product data management point of view, similarities exist between the CAD model structure and the structure represented within the simulation modelling languages. The product structure serves in both cases as the lowest common denominator. In last consequence, the model structure within the different modelling languages represents a derived product structure from different engineering discipline viewpoints (see figure 1). The model structure represents only the discipline relevant properties of thc product structure..

Precondition for the application of PDM-systems in the computation and simulation departments is a mapping of the respective simulation model information to the PDM-systems' meta data model (see section 2). Thereby, an explicit transfer of all information from the available modelling languages is not feasible due to the presented variety. As a solution a metamodel is introduced, which could serve as a basis for the application of PDM-systems in the simulation processes. This metamodel will be presented in the following section.

4 Object Oriented Simulation Metamodel

In order to integrate the variety of simulation modelling languages, an object oriented metamodel was developed. Please notice the difference between the PDM-systems meta data model as explained in section 2 and the simulation metamodel presented in this section.

The developed metamodel allows to only extract the product data management relevant information of the modelling languages and put these information on a uniform basis. This concept makes it possible to represent the relevant information (e.g. structure and parameters) of the modelling languages in such a way that it can be processed in PDM-systems. Approaches from Systems Engineering were used to build the initial metamodel structure. Systems Engineering defines a system as a collection of parts with relationships between these parts. Characteristics of a system are the system border and system environment, the elements and their relationships, the system structure with sub- and super-systems, the system hierarchy, and the different views on a system [14]. In a first step, these basic characteristics were defined in the metamodel to then make a further detailed specification according to the requirements of the modelling languages. A second source for the definition of the metamodel was the currently developed 'PDM schema' within the ISO 10303-STEP activities [14].

In order to represent the different aspects of the modelling languages an ontology is provided to ensure an unambiguous mapping between the objects. An excerpt of the ontology is shown in table 2.

Table 2. Excerpt of the ontology

	METAMODELL	VHDL	DYMOLA	SD/FAST	SPICE
Observed unit	system	entity	model	model	circuit
Structure definition	sytem_ configuration	configura-tion	submodel	implicit	implicit
Parameters of the system	system_parameter	generic	parameter, constant	gravity	---
Elements of the system	system_element	entity, component	model_ instance	body, ground	1. column
Parameters of the elements	element_parameter	generic	parameter, constant	mass, inertia	last column
Element connection	connection_point	port	node	joint	2.+3. column

The Metamodel contains for example an object *system* which corresponds to the VHDL language element *entity*, the Dymola language element *model*, the SD/FAST language element *model*, and the SPICE language element *circuit*. These common characteristics could be used to uniformly manage the model structures defined within the different modelling languages. A PDM-system only needs to implement the metamodel object *system* in its meta data model. With this implementation of only one object, the PDM-system is able to interpret the language elements of four modelling languages based on the presented ontology. The same methodology was used to represent other areas of the modelling languages such as model hierarchy, modularisation, or model parameters. The resulting metamodel structure is depicted in figure 4.

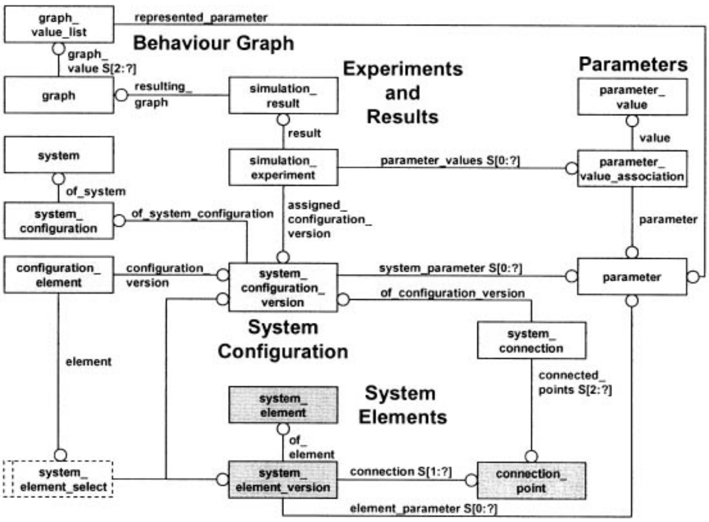


Figure 4. Basic structure of the metamodel (simplified)

From the product data management point of view, the metamodel enhances the functionality of the modelling languages. These are currently in general not able to manage versions or configurations of elements. For this purpose the metamodel provides a versioning concept with the objects *system_version, system_configuration_version* and *system_element_version* which allows the versioning of system configurations and system elements. The configuration of models is possible with the metamodel objects *configuration_element* and *system_configuration*. Based on the metamodel, it is possible to implement a configuration management of simulation models within PDM-systems. Furthermore the metamodel objects *simulation_experiment*, and *simulation_result* allow the storage and unambiguous assignment between configurations of simulation models, performed simulation experiments and achieved simulation results.

5 Application of the Metamodel within PDM-Systems - Benefits

The presented metamodel provides a basis for the management of simulation models within PDM systems. What will be the benefits of such an application? An important functionality of PDM-systems is the configuration management. A configuration is the status of the product structure and its linked documents at a defined point of time. The configuration management defines the corresponding technical and process rules and is usually an integrated constituent of PDM-systems. As presented in section 2, TDM-systems allow the configuration of 3D-CAD model structures. Similar to the 3D-CAD model structure, each simulation system stores the model structure of its simulation models. With the application of the metamodel it becomes possible to configure the structure of simulation systems.

The benefit of this procedure is that it leads to a single source for product configuration. It is only necessary to define a product configuration once in a PDM-system. Then it can be used subsequently in design, simulation, and computation tools.

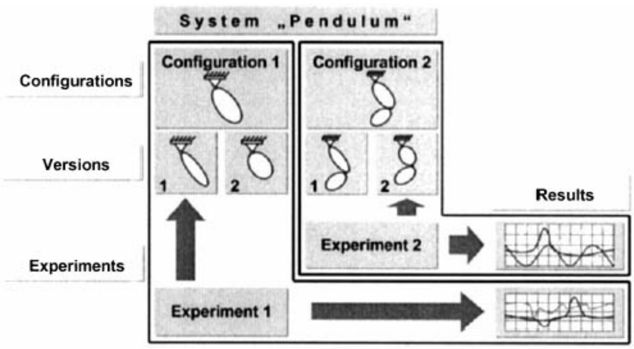


Figure 5. Dependencies between configurations, versions, experiments and results

Against the background of an ISO 9000 certification it is necessary to ensure the safety of the product. The geometrical safety of the product is guaranteed because of the close integration of CAD- and PDM/TDM-systems. With a metamodel enabled PDM-system also the functional safety of product behaviour could be achieved. The approach chosen within the metamodel is shown in figure 5. It allows to realise an unambiguous assignment between the performed simulation experiments, the achieved results and the versioned simulation model configuration. Because of a consistent version and configuration management during the entire process chain, errors like simulating a wrong version or outdated configuration of a product cannot occur any longer.

An application scenario starts with the configuration of simulation models based on the product structure in the PDM-system. With the presented ontology, the PDM-system automatically generates the model structure in an arbitrary modelling language. The simulation tool calculates the results with the correct configuration and version and submits them back to the PDM-system. The PDM-system ensures the storage of results linked to the correct model configuration and version.

6 Conclusion

In this contribution a current deficit of product data management arising from the insufficient processing of mechatronic simulation data was pointed out. A metamodel approach was presented in order to handle the variety of modelling languages. The metamodel approach enables PDM-systems to manage and integrate information of the computation and simulation departments. The results presented in this contribution show that it is possible to perform a uniform management of system models which are defined in different modelling languages.

In summary, the computation and simulation processes take a crucial role in advanced virtual product development. Only by a consequent application of engineering analysis tools in all design phases and a data integration with PDM-systems, the number of time-consuming and cost-intensive prototypes can be reduced.

References

[1] Yoshiharu Hori: 3D Data Based Vehicle Development at Nissan, 4^{th} Workshop of Current CAX problems, Ito, Japan, Nov. 2000

[2] Vajna, S.: Die neue Richtlinie VDI 2219: Praxiserprobte Hinweise zu Einführungsstrategien und Wirtschaftlichkeit von EDM/PDM-Systemen. In: VDI Bericht 1497, München, 19.10.1999. Düsseldorf, VDI Verlag, 1999, S. 25-42.

[3] Anderl, R.; Krastel, M.: Multidisciplinary Product Data Management. In: Proceedings of the 2^{nd} Symposium of Concurrent Multidisciplinary Engineering, Bremen, 1999.

[4] McIntosh, K. G.: Engineering Data Management – A Guide to Successful Implementation. Berkshire: McGraw-Hill Book Company Europe, 1995.

[5] Machner, B.; Trautheim, A.: PDM – TDM: PDMI2 position paper for the system integration scenario. Project report, ProSTEP GmbH, Darmstadt, 1998.

[6] Herden, W.; Jahn, R.: Produktdatenmanagement für Entwicklung und Fertigung von mechatronischen Systemen. In: VDI Bericht Nr. 1289, Düsseldorf: VDI-Verlag, 1996.

[7] Isermann, R.: Mechatronische Systeme - Grundlagen. Heidelberg: Springer, 1999.

[8] Cellier, François E.: Continuous System Modeling. New York: Springer-Verlag, 1991.

[9] Anderl, R.; Krastel, M.: Multidisciplinary Product Data Management (MPDM) for the development and simulation of mechatronic products. In: Proceedings: Product Data Technology Europe 1999. Stavanger, Norway, 1999, S. 65-73.

[10] Otter, M.: Objektorientierte Modellierung Physikalischer Systeme, Teil 1. In: Automatisierungstechnik 47 (1999), R. Oldenbourg Verlag, 1999.

[11] Ten Hagen, K.: Abstrakte Modellierung digitaler Schaltungen: VHDL vom Funktionalen Modell bis zur Gatterebene. Heidelberg, Springer Verlag, 1995.

[12] Hollars, M. G.; Rosenthal, D. E.; Sherman, M. A.: SD/FAST User's Manual. Version B.2, Mountain View, USA. Symbolic Dynamics, 1994.

[13] Haberfellner, R.: Systems Engineering: Methodik und Praxis. Zürich: Verlag Industrielle Organisation, 1997.

[14] Kindrick, J.; Hauser, M.; Barra, R.: Usage Guide for the STEP PDM Schema – Release 4.1. http://www.pdm-if.org/pdm_schema/usageguide/pdmug_release4_1.pdf..

Dipl.-Ing. Marcus Krastel
Department of Computer Integrated Design (DiK),
Darmstadt University of Technology,
Petersenstr. 30, D-64287 Darmstadt, Germany,
Phone: ++49-6151-16-6583,
E-mail: krastel@dik.tu-darmstadt.de,
www.dik.maschinenbau.tu-darmstadt.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

REDUCING DESIGN DEVELOPMENT CYCLE BY DATA MANAGEMENT WITHIN THE DESIGN OFFICE

M Storga, D Pavlic, and D Marjanovic

Keywords: Product data management, design information management, information representation, web-based systems.

1 Introduction

Product development is exposed to increasing demands with respect to quality and costs [1]. Striving to improve and optimise the production process, companies have been increasingly computerizing certain areas of production process. Particularly in the product development process, introduction of the CAD technology has increased efficiency. In the same time, product development process within design office has inherited the traditional methods of product data management. Those traditional methods cause many problems for the companies: effort and money is spent unproductively for data recapturing; development time is extended needlessly; data correctness can not be guaranteed due to the redundant data storage and multiple data conversions between different systems. Such problems have a strong negative effect on company's competitiveness, market share and revenues. Driven by such issues, the need for a software design tools to support representation and engineering information management becomes more critical.

This paper is concerned with data management system for design office of small to medium size enterprise with standardized single unit production line. Although presented research is focused on the particular company, the identification of data management problems within design office and implementation methodology are guide general. The support offered to design office from the PDM technology viewpoint is discussed. An efficient usage of PDM technology demands knowledge from a number of disciplines such as management, product development, database systems, and information technology. The aim of this research performed in cooperation with the electrical power transformer factory is to develop a framework to integrate those aspects.

Product data can no longer be locked into one application or another, but must be reusable down stream, as well as up stream, and by many applications as possible [2]. Reuse introduces a different perspective on the ownership of product data. Product data may not be treated as the responsibility of any particular department within the company. Instead, all product data must be treated as a corporate asset, where different departments should have different view ports on data, and different privileges for its use. The need to compress lead-time requires the introduction of concurrent engineering techniques. All the activities need to use and share the different subsets of the overall product data. This in turn brings the need for an effective Product Data Management system that controls the creation, reuse and retrieval of product data. Such a system can serve as the "backbone" for managing the product lifecycle as shown on Figure 1. PDM manages product data through the enterprise, ensuring that the right information is available for the right person at the right time and in the right form [3].

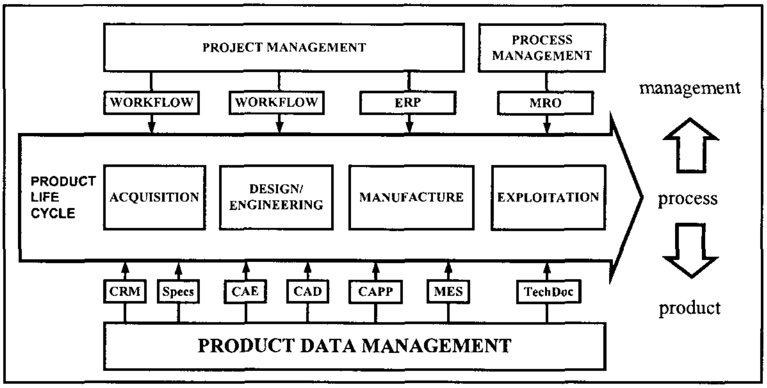


Figure 1. PDM – the backbone for managing the product lifecycle [2]

The goal of the project is to reduce the design development time. After preliminary problem analysis the following stages of the project were determined:

- Data collection/analysis about existing design process, methods and tools;
- Problem identifications and solution proposal;
- Solution implementation;
- Validation.

The data about existing design process, design methods and tools used within design office have been collected within following methods:

- Task time analysis in hours per day for all employees in the design office;
- Unstructured interviews with designers guided with a questionnaire proposed after the first series of interviews.

The structure of this paper is as follows; in chapter 2 requirements on PDM systems are defined and key functions are described; in chapter 3 the methodology of the PDM systems introduction to the product development process is discussed; in chapter 4, test example of the developed PDM system in design office is introduced. Finally, in chapter 5, the conclusions are drawn.

2 Basic requirements and key functions of a PDM system

Today, there is no generally accepted definition of what a PDM system is, but is possible to define what a PDM system should do. In order to be able to define expectations of the PDM system, the interviews were done with members of a design department office. According to founded problems and expectations, which had to be solved during the introduction of PDM system, and using the literature [4], [5], [6], the basic requirements of a PDM system were determined:

- *Secure storage of documents and product information in a central store with controlled access and without data redundancy*
  The first motivation for considering PDM comes from designers frustrated from not being able to find the document they are looking for. The people may only knew that the needed documents exist in a hard copy or e-document, but the time needed to get it is not acceptable.

- *Management of the previews designs*
  The designers spend more time modifying existing designs for new projects, because design tasks belong in high percentages to the adaptive and variant design tasks. The traditional design document system was inappropriate to CAD tools leading.

- *Management of alternative variants of design object*
  Many design objects (products and documents) have alternative variants. For example, the main assemblies of power transformer can be made in several standard variants, or a technical documentation for a particular product can be available in different languages.

- *Management of the inspection and approval procedures*
  Design tasks and design objects should be checked and approved with prescribed procedures enabling results independent on a his or her experience.

- *Management of the recursive division of design objects into smaller components*
  Almost all design objects of electric power transformer have a hierarchical breakdown structure, which divides the design object into assemblies, which are further divided into subassemblies, components, etc.

- *Management of a multiple structural views of a design object.*
  A PDM system should make it possible to have different views of a design object. For example, document structure and content level of detail, varies with the purpose of the document (contract, manufacturing, test, maintains).

- *Software tool integration*
  From the designers' viewpoint, the usability of a PDM system depends very much on how well the system is integrated with other tools needed in daily work (of special importance is the exchange of product information with the CAD system).

- *Component and supplier management*
  The management of standard components from external suppliers is significantly time consuming, because inconsistency in design data between different producers.

Based on this requirements the following key functions of PDM system were determined: product data modelling, design process information management and design workflow management, as shown on Figure 2.

Product Data Modelling

Product data model includes data definition and classification, describing products properties and related document data, as well as relationships between them. To ensure efficient storage and data retrieval, a methodology for data classification has been established according to the specific needs of the company. One of the fundamental properties of any product is a breakdown structure, that describes how the product is divided into assemblies, which are in turn divided into sub assemblies, etc. Documents relating to assemblies and subassemblies are similarly structured, maintaining relations between documents and the products structure elements. Generally, companies are increasingly interested in configurable products, which may be adapted individually for customer requirements. In case, all standard variants of the electric power transformer have been described with generic product structure, which is the basis for modelling the electric power transformers data.

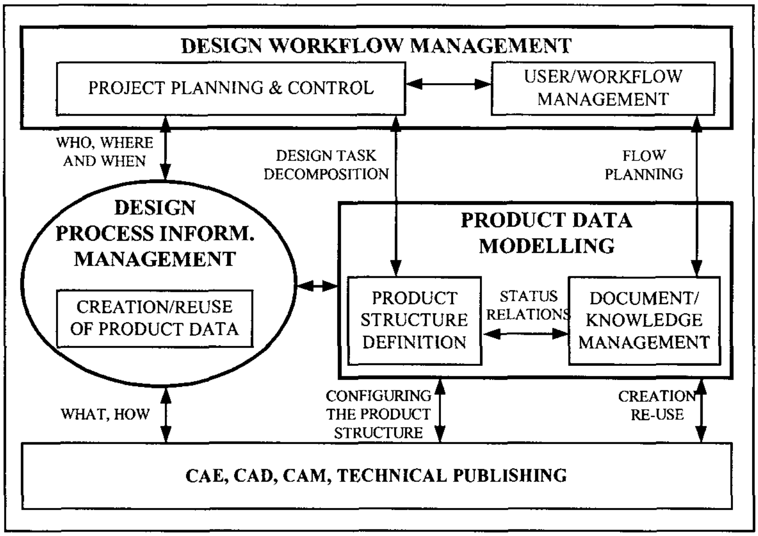


Figure 2. The key functions of PDM system

Design process information management

Design process information management includes activities that create or use product data. Among the required functionality of data processing support are: easy data capture, indexing and storage; fast data browsing and retrieval; multi-view data representation; data transformation between models/views and consistency management. The key property is that user needs to be able to 'get at' assemblies and subassemblies data by a variety of routes. User could move up and down through the product tree structure in different ways; pick the path through a product structure; simply call-up data user want by searching for it by name or number, or search for groups of data by specifying an attribute or combination of attributes. Every modification must be separate captured, because different versions can represent the evolution of a product through development stages.

Design workflow management

Design workflow management covers user rights, design task assignments, project planning and control. The term 'design workflow' refers to the flow of work through design activities in which design data need to be created, modified, viewed, checked and approved by many different people, perhaps several times over. Design workflow is not a linear process; it is a complex process in which some activities run in series, and some run in parallel. The traditional design workflow is inefficient. Unnecessary steps, time-consuming changes make it expensive, slow, spending more than expected. It suffers from poor communications and a lack of management understanding. Therefore, it is very important for project leaders to control the progress of the project, influencing on the product development time and expenses.

3 Methodology of the developed PDM system introduction

Considering the key functions of Product Data Management system from the theoretical viewpoint, the next step is the practical introduction of the system into the design office. Introduction methodology is a separate problem and will be discussed in this chapter.

Successful introduction of a Product Data Management system can be a time consuming and complex process, but it is well worth the extra effort required to do it right the first time [7]. They are many issues to consider in the process of evaluating and introduction a PDM system. Each step in this process is vital to overall process success. The outline of proposed methodology is given in the following five steps:

1. *Identifying specific company's needs*
   Involves identification of specific needs and requirements from both a management and end-user perspective. A CAD operator needs a design-oriented approach to locate files and revisions while an engineering manager would be more concerned with procedures to shorter design cycle and increase the overall productivity of the department. Ad hoc solutions for data archiving are the main cause of problems in exploitation.

2. *Analysing company's existing information flow*
   The data about the workflow and processes in the design office, gathered through the series of unstructured interviews has been used as baseline for information flow analyses. Overall workflow is analysed with detailed look at: data flow, the data format (hard copy, digital notes...); the processes over the data; the time periods for each process and a time it takes to transfer data between processes. After this analyse, bottlenecks were identified. A bottleneck is anything that causes delays or creates extra work such a paper based transfer of information, duplicate entry of data, inefficient work methods or unnecessary procedures.

3. *Designing the system's architecture*
   Includes the database structure designing, the security scheme, the user interfaces and users' communication protocols, data presentation forms and system developing tools. The storage and memory requirements need to be determined. The number of concurrent users and corresponding network bandwidth requirements need to be determined to insure adequate performance. Additional work may be needed if the system needs to be integrated with other applications.

4. *Planning the implementation procedure and implementation*
   The key to smooth implementation is planning. The more time spent on planning, the less time will be required for the implementation. There are many decisions and questions. It is necessary to know what resources (people and time) are required, what is the implementation schedule and which documents should be considered. Designers should receive a proper training before the implementation. Administrative issues of the new system should be prescribed and familiar to each member of the design team; special care should be taken for backups and recovery procedures.

5. *Maintaining the system*
   Once the system is functional and users are satisfied, maintain become crucial. The system will meet expectations of designers only by delivering constant functionality. Therefore, maintains of such systems requires the new kind of specialist for the design office capable to solve periodical updates, solving unexpected problems and implementation of the additional users requirement, problems from the computational viewpoint and from designers viewpoint.

4 Project implementation

As mentioned before, accordingly to the proposed methodology, the prototype of the PDM system has been introduced into design office of electric power transformers factory. The production process of the electric power transformer could be described as a low volume production of valuable products composed of in function highly standardised main assemblies. The range of products has been standardised by variations in structure form and performance. The most of the hierarchical structure of the electric power transformer is well known. That enables the concept of "design templates" which may be used as a support framework in the design process for appropriate new designs or redesign of the existing solutions. Design template, represents the generic product data structure, and includes: a hierarchical relations between assemblies; material information, component identification, quantities, norms, instruction documents for designing procedures and possible variants of standardized assemblies.

All the activities along the design workflow create and/or use product data. The workflow exists to provide the product data necessary to produce and the support the product. Without product data there would be no need for the design workflow. Product data and design workflow are very closely linked. Based on experience from the existing design process, design data flow has been coordinated. Such coordinated data flow manages: project identification; tasks schedule planning for each project phase; current status of the pending projects; information about the previous projects; key parameters for catalogue, storing, searching and retrieving of design solutions; users' identification and status. Design activity procedures for each user's group are also included and described in [8].

Digital model of design templates and design workflow are mapped within relational database system [8]. Database structure creates an information infrastructure for virtual design environment, which is used as a core for the product data and document exchange between engineers in design development cycle [Figure 3].

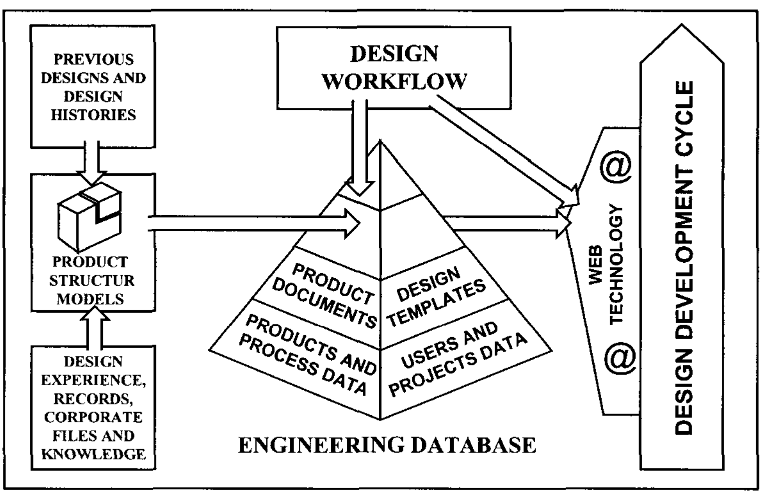


Figure 3. Implementation architecture of PDM system

Mechanisms included into the database system insure data consistency and support the activities of design information management through security storing, searching and retrieving data. Project implementation includes a realization of adaptive user interfaces that allow distributed users to fill, edit and browse engineering database [9]. User interfaces were developed using the 'Web' technology. The new dynamic features add a wealth of power and interactivity to the World Wide Web applications through dynamic data manipulation. Through the dynamic web pages the members of the design team can follow the product structure to start their own work and than to collaborate with each other on project. E-documents of the product are being exchanged using 'Web' technology [Figure 4]. Those documents include product specification, design notes, component variants specifications and on-line created Bill of Materials. Communication between the users is realised through the automatic email protocol, through which the user gets all the information necessary for his design duties.

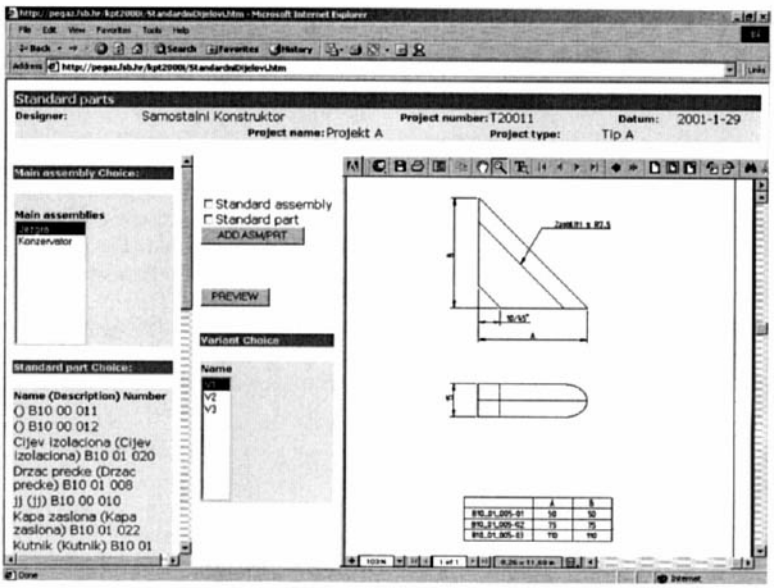


Figure 4. Example of users form with integrated e-document

5 Conclusion

Generally the validation of research projects is a very subtle area. With a given goal the validation is quite clear because the measure of success is the requested reduction of design development cycle time. Described system is one part of solution implemented in real situation [10] that has demonstrate significant possibilities of reducing design time cycle by the introduction of the developed PDM system in design office. An improved collaboration and management within a product development team is a key factor to an efficient product development process. The proposed system supports the product development process on two levels: engineering data management and design data flow coordination. Data modelling is a key issue in order to achieve an effective structure that will support the product data and document management. By coordination of the engineering workflow, bottlenecks are reduced, and better control of the projects is established. As a final result of proposed methodology design productivity was improved, data integrity was protected and design time was reduced.

References

[1] Andreasen, M.M., Hein, L.: "Integrated Product Development"; IFS Ltd; Springer-Verlag; Berlin NewYork London Paris Tokyo; 1985.

[2] Timmerman, H.: "Do your homework, save time and money"; CD with selection of 25 Conference Papers presented at "e-business for Industry", London, 2000.

[3] Schnurer, R.: "The Application of Product Data Modelling Techniques to Product Support"; CD with selection of 25 Conference Papers presented at "APLS/CALS Europe", London, 1999.

[4] Peltonen, H.: "Concepts and a Implementation for Product Data Management"; Doctorate Thesis, Helsinki University of Technology, Department of Computer Science and Engineering, Finland, 2000.

[5] van den Hamer, P., Lepoeter, K.: "The Philips 5-dimensional framework for modelling design data and design process"; Proceeding of the 2nd WDK Workshop on Product Structuring; pp 83-108; Delft University of Technology; The Netherlands; 1996.

[6] Piskoz, P, Malmqvist, J.: "Possibilities and Limitations when Introducing PDM Systems to Support the Product Development Process"; Proceedings NordDesign '96; pp 165-175, Espoo, Finland; 1996.

[7] Fries, B.: "Successful Implementation of Product Data Management Systems"; Premier Design Systems, Inc.; http://www.pdmic.com/articles/artpdmim.html; 1995

[8] Pavlic, D., Marjanovic, D., Storga, M.: "The implementation of web-based technologies in engineering data management"; Proceedings of the 6th International Design Conference DESIGN 2000"; CTT, FSB, WDK; ed. Marjanovic, D.; Dubrovnik; Croatia; 2000.

[9] Storga, M., Pavkovic, N., Marjanovic, D.: "Computer Aided Product Structure Design"; Proceedings of the 6th International Design Conference DESIGN 2000"; CTT, FSB, WDK; ed. Marjanovic, D.; Dubrovnik; Croatia; 2000.

[10] Marjanovic Dorian, Bojcetic Nenad, Dekovic Damir, Pavkovic Neven, Pavlic Davor, Storga Mario, Zezelj Dragan: "Design Department Data Flow Integration"; Proceedings 3rd International Workshop IPD 2000; Magdeburg, Germany; University of Magdeburg, 2000.

Mario Storga, B.Sc.
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, Design Department, Ivana Lucica 5, 10000 Zagreb, CROATIA, Tel: + 385 1 6168 117, Fax: + 385 1 6156 940, e-mail: mario.storga@fsb.hr, URL: www.cadlab.fsb.hr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

AN ENGINEERING WORKBENCH FOR COMPLEX PRODUCTS, BASED ON A HYBRID PRODUCT MODEL

M Koch and H Meerkamm

Keywords: product modelling, digital mock-up, product data management.

1 Introduction

Today's state of the art three-dimensional CAD systems are well suited to handle and display geometric information of complex shapes, but there is still a need for providing appropriate tools to support the whole product development process. Especially synthesis and analysis functionality for the early (non-geometric) stages of design are not available. On this account a new approach was developed to handle geometric data as well as non-geometric information (such as organisational, technological and functional data). Using a so-called "hybrid product model" an implementation is possible which provides extensive support for the mechanical engineer during his design. Further more such an approach permits extension to the early stages of design and thereby facilitates a continuous DMU-process (digital mock-up).

2 Product Model Requirements

The demands on the management of product data have been increasing enormous over the last few years. On the one hand the amount of information which is generated during a development process has been multiplied several times. On the other hand these data should be available at every stage of design following the phrase "any place, any time". These new demands are beyond the scope of existing models which are usually governed exclusively by the geometric information of the currently processed product. To resolve this problem, various product models have been built up by several research facilities. The new characteristics and potentials, claimed in brief in the last preceding paragraph, can be summarised as follows [1]:

The product model has to be able to accompany the product through the entire process of design. This means that product-specific information has to be handled from the first steps in planning, over the conceptual, embodiment and detailed design to the production. Finally, even the product recycling uses information from the early stages of design and therefore has to be supported by the product model, too.

To support the idea of a complete digital mock-up, which includes e.g. additional information for decisions in the past or simulation results, all accumulating information has to be stored inside the product model. These data can differ fundamentally from product type to product type. The data structures have to be modular in order to compromise this variety. Therefore it is easily possible to suit them appropriately to an individual product type.

The collected product data has to be accessible in other stages of design and even other projects with different designers. Meeting this high availability makes great demands on hard- and software which have not to be underestimated.

Today's known features of shape-oriented CAD systems like unlimited nesting levels and multiple fitting of parts and assemblies have to be supported and transferred to this new approach.

Last but not least a state of the art implementation has to be open to interpret and implement new ideas of tomorrow. This "option for further development" can be maintained through a standard interface.

3 The Engineering Workbench *mfk* and existing restrictions

The Engineering Workbench *mfk* (KS*mfk*), which has been developed at the Institute of Engineering Design (University of Erlangen-Nuremberg, Germany), is an approach for bridging the described gap from the designers task to CAD systems by use of design elements [2,3]. Inside the semantic and relation based product model design elements are provided for covering all the design knowledge. Although the KS*mfk* is set on efficient CAD systems, their functionality is used only for dialog control and visualisation. In several research activities couplings to different CAD systems have been implemented whereby the current developments base on Pro/ENGINEER™ (Parametric Technology Corporation, MA, USA). For fulfilling the specific requirements of variable product families and design stages, the Engineering Workbench is subdivided in distinct modules, e.g. for sheet metal design or design of cast parts. The concept of the Engineering Workbench consists of a synthesis part for the description of the product and an analysis part for evaluating different DfX (design for X) criteria. A connection between these parts is guaranteed through the product model and the pattern editor.

Due to the research and considerable improvements in the above described Workbench *mfk* a huge step was made towards a designer-friendly tool supporting the whole process of design. However, developing this tool for years revealed problems as well in usage as in implementation:

Being challenged with complex design problems the Engineering Workbench shows some restrictions in *usage:* On the one hand modern CAD systems build perfect geometry like free form surfaces, but these features can not be reproduced by the KS*mfk*. Modelling all geometrical shape with design elements fails in case of free form surfaces. The reason for this consists in the nature of design elements, providing geometrical patterns with the ability to vary some parameters. For complex shapes it is impossible to provide appropriate patterns. On the other hand today's products mostly consist of many parts and assemblies. These objects can be arranged in any desired structure. Objects, which occur repeatedly, have to be stored every time they are used. This wastes capacity and makes advanced analyses (e.g. an automatic exchange of standard parts) impossible.

Another problem consists in the *implementation* of data structures for containing the product model. This static structures embedded in the Engineering Workbench *mfk* entail a complete new build-up or at least a revision of the interfaces from the additional software to the product model for every new analysing tool or software update. Beyond this a complex source code edited by different programmers is extremely error-prone by itself.

Meeting the unresolved problems mentioned above led to two separate steps: first the handling of elements and objects inside the product model had to be reconsidered and adapted, then the inefficient and unmanageable implementation of all data structures had to be restructured and tightened.

4 The New Approach

4.1 The Hybrid Product Model

The goal to overcome the geometrical restrictions of design characteristics leads to the concept of a "hybrid product model". Beneath the former components of the product model, also pure geometry shall be provided with semantics.

To connect the unachievable handling of complex geometry in CAD systems to the idea of the Engineering Workbench *mfk* to support the entire process of design, the following consideration was made: Until this point of time all shapes used in the CAD system were totally described in the product model of the KS*mfk*. On the one hand this was very useful, because no data from the CAD system was needed to build up the shape. On the other hand this redundant information led to the described lack in modelling complex surfaces. If it is not possible to hold all necessary data for the free form surfaces inside the product model, consequently the data generated by the CAD system has to be used to manage more complex elements. But how can such a data-structure, which only uses selected information of the CAD system, be realised?

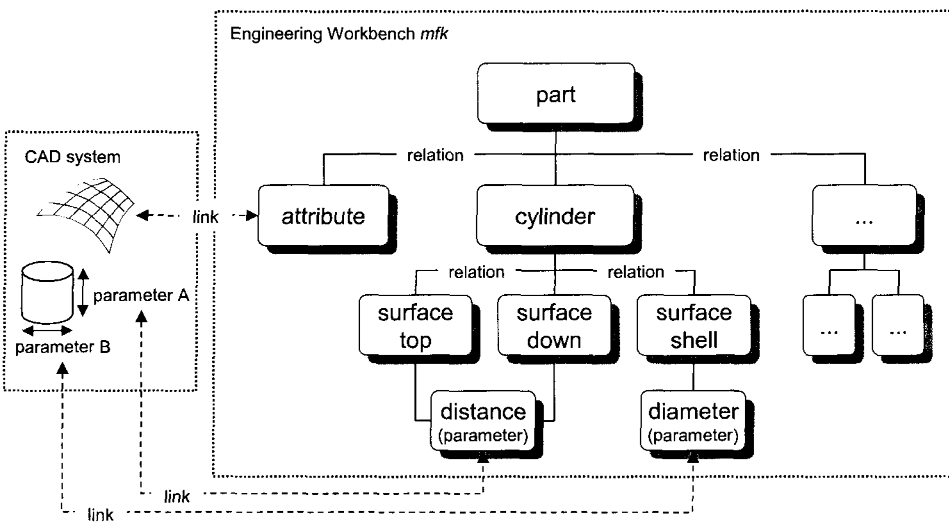


Figure 1. An entity-relationship-model with a link to external CAD reference

The Entity-Relationship-Structure, which represented the base of the former product model as well, is able to handle this "exception" very smart. To make the CAD-generated geometry part of the Engineering Workbench *mfk*, an attribute is set to the otherwise unknown shape (figure 1). For the very most cases of synthesis and analyses the information given by this attribute-"flag" is absolutely sufficient. For further geometrical characteristics, e.g. dimensions, a link (stored on the "flag") to the CAD data can be used. To access these values, both, the CAD system and the specific product data generated by this system are necessary. This avoids redundant geometric information in product model and CAD data. As a consequence of binding information to pure geometry, also the CAD model has to be stored with or within the product model. Further a connection from product model to CAD model has to be established.

For instance the rim in figure 2 itself has a complex shape which can not be described with predefined geometrical patterns, but a CAD system can shape it without problems (figure 2, right). In contrast a design element for the drilling pattern inside the rim can easily be provided by the KS*mfk* (figure 2, left). To connect the both elements in the Engineering Workbench *mfk* an attribute is set to the complex shape of the rim. This makes the external object "rim" known to the product model and thereby accessible to any desired relations inside the product model. To put the pattern on the rim, the simple attribute can be accessed instead of the complex geometry. For any additional information which is needed for analyses etc. the attribute contains the link to the geometry inside the CAD system. Following this link, any information can be output by the CAD system's programming interface (Pro/TOOLKIT™ in case of Pro/ENGINEER™). The requested values are then passed back to the hybrid product model and are there stored temporarily during the analysis.

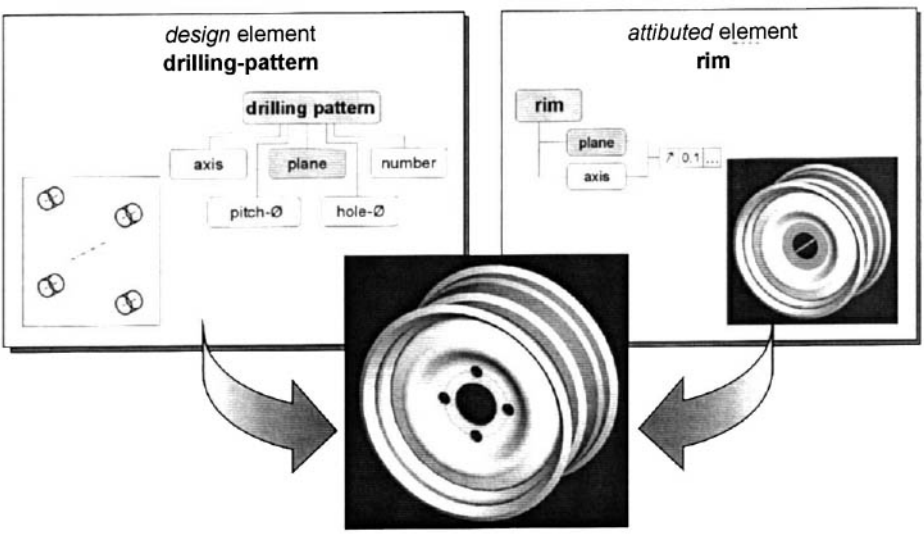


Figure 2. Complex products with design- and attributed elements

Resulting from the use of attributes a further advance stores pre-defined templates for reuse. The structure of these attributes is described through appropriate classes inside the product model. In a class the structure of an attribute is determined and can be created or changed by an user or the administrator. This concept allows to set-up an attribute structure manually once and to offer it to various designers for reuse in several parts more often.

An example for handling these templates is a material-characteristics-class and -attribute. The rudimentary structure could look like the attribute class in figure 3 (left). To create an attribute for any desired element, e.g. a cylinder (right), the workbench builds up a datasheet and connects it to the element. This element either can be a known workbench-element or an element created by the CAD system (that means another attribute).

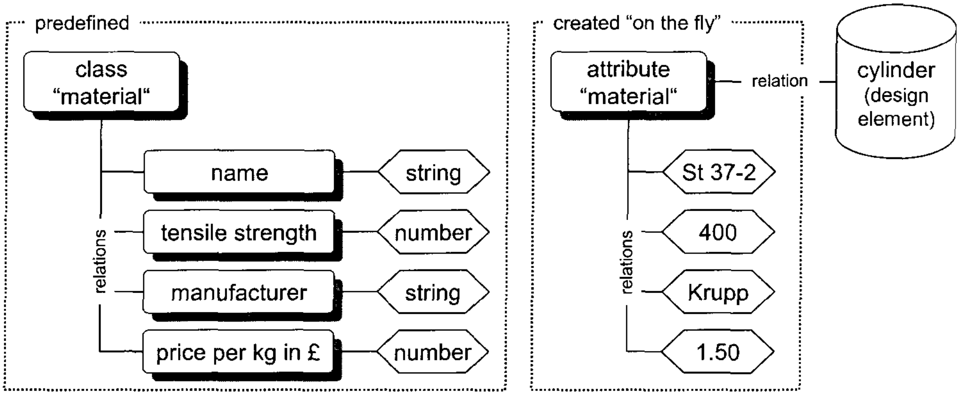


Figure 3. Classes and deduced attributes for elements (e.g. a cylinder)

A further restriction was the handling of parts and assemblies inside the Engineering Workbench. The limitation on one nesting level was not any longer topical and had to be updated to the standard of modern CAD systems. Therefore the entity-relationship-model was used as well. The different objects (parts and assemblies) are connected by relations. Fixing more relations to one part automatically means a multiple fitting of this specific – only once managed – part. To embrace the fact of various build-in situations, additional positioning-information can be added to every connectivity-relation (figure 4). Using this method, managing multiple fitting of objects in any desired nesting levels is possible without problems [4].

4.2 New structures for data management

In contrast to the growing problems in implementation of data structures with expanding functionality of the system KS*mfk*, the fundamental idea of an entity-relationship-model, on which the whole system is based, was not upset during all the time. Therefore this approach entered the new product model as the fundamental theory. All parts of the system are now structured in entities and relations [5]. The entity-relationship data model views all elements of a part or assembly as a set of basic objects (entities) and relationships among these objects. The outstanding advantage of this system is its scalability. While object-oriented methods have to be initialised at the beginning of the design process, it is possible to upgrade the data structure on-line during the whole process. This flexibility is very important, because at the beginning of a more complex design task it is never possible to survey all needed data structures.

As explained in the preceding sections, the enormous amount of product-information can not be stored in data structures within the hybrid product model of the Engineering Workbench *mfk*. To manage larger data capacities securely and quickly, a commercial data base software is used [4]. Because entities and relations represent the product model, it is obvious to choose a relational data base. Outsourcing the data structures towards an external data base has various positive side effects: First of all the users never have direct access to information offered by in or written to the data base. A special part of a commercial software called database management system controls every read – and even more important – every write access. That means the integrity of data base information is granted at every time. Furthermore the Engineering Workbench gets rid of all problems (regarding performance, disk space etc.) by handing over data to the data base. Whether efficiency or data throughput will not be sufficient for further tasks, an other more powerful data base solution can be connected to the product model replacing the actual one. To make such a change-over possible, a standard interface between the data base and the KS*mfk* is necessary. During the last years the interface-language "SQL" (structured query language) has reached a wide spread among relational data bases and has established itself to a world wide standard. The Engineering Workbench is a further application supporting SQL.

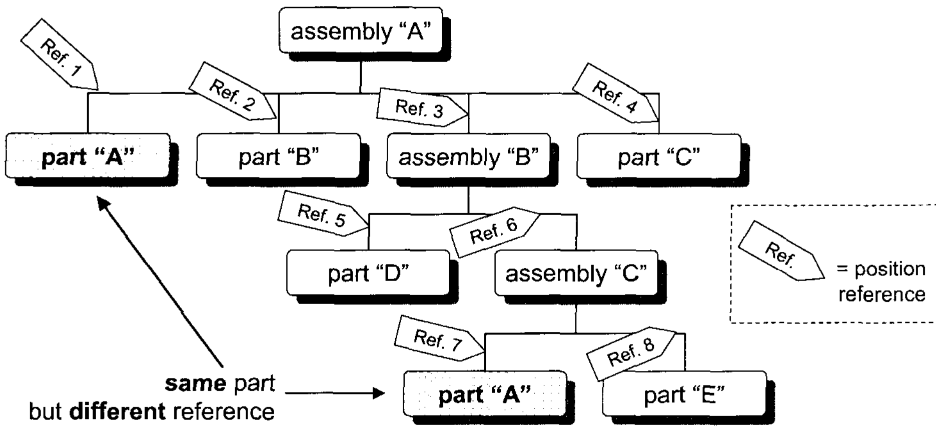


Figure 4. Multiple fitting of a part with different position references

The herein mentioned advantages according SQL have already been verified by replacing a very simple software ("mSQL", Hughes Technologies) by a professional data base system ("Oracle", Oracle Corporation).

5 Example

The new possibilities of the hybrid product model are now shown at a power train and a differential gear of a car. Figure 5 shows the assembly "power train" which consists of four wheels and the differential gear.

During early stages in design these five elements are connected merely by lines, which represent functions. With the Engineering Workbench *mfk* functions are handled by attributes inside the product model. Another attribute makes the displayed design space known to the Engineering Workbench (figure 5, left). Using attributes makes it possible to turn the advantages of the system KS*mfk* in account without abdication of the CAD software. The designer is enabled to work in a known environment whereby the acceptance and productivity of the early stages of design will be improved significantly.

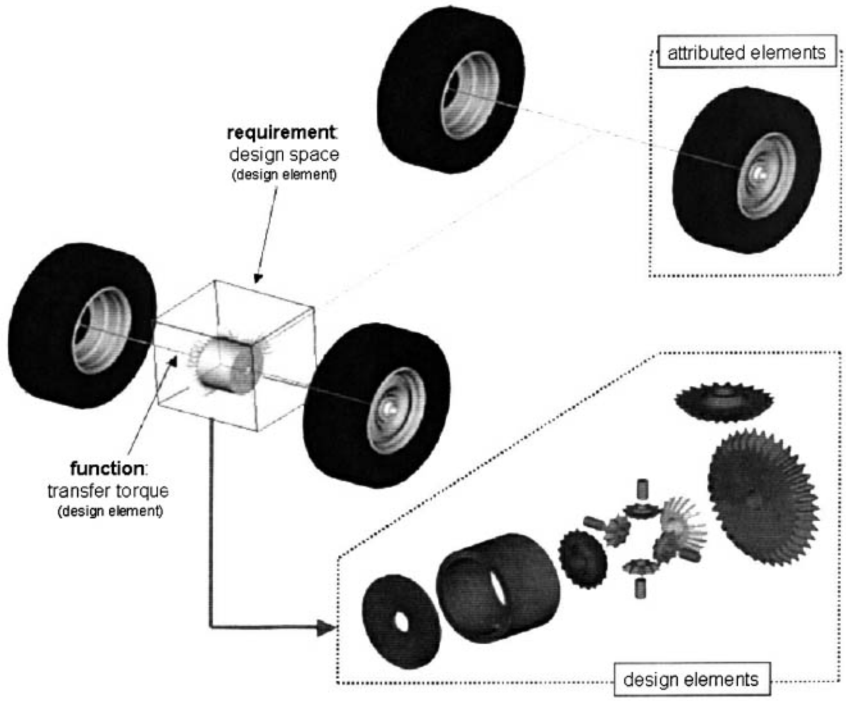


Figure 5. Assembly "power train" with design elements and attributed elements

The power train in one concrete solution includes different sub-assemblies: on the one hand there are design elements, which consist of geometry advanced by additional information. On the other hand attributed elements with complex free-form surfaces like the wheels are built in

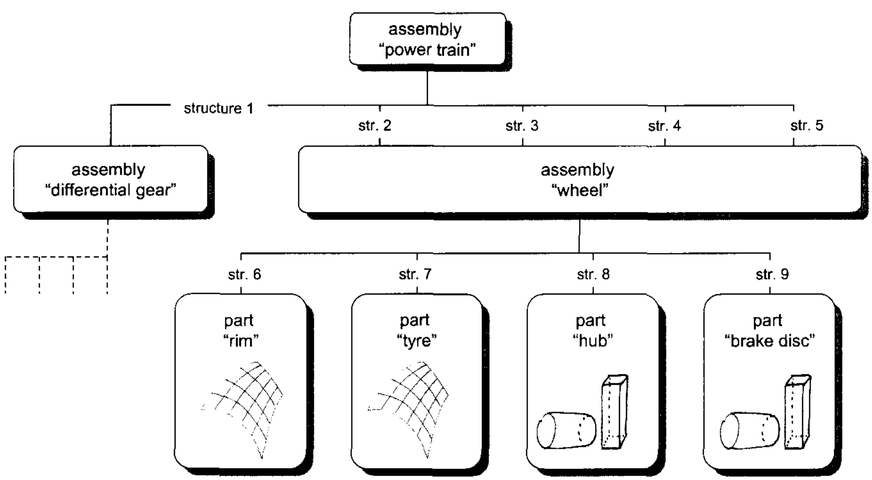


Figure 6. Structure of the example "power train"

the train (figure 6). Attributes are used in CAD-generated geometry as well as in the basic elements of the gear. First case attributes fulfil the general support for complex geometry while second case attributes append additional, non-geometric information (e.g. material data or vendor) to the parts.

6 Conclusion

There is still a tremendous lack between the designer's work and the capability of CAD systems. With the presented "hybrid product model" it is possible to connect the productivity of a present CAD system to the widespread possibilities of the Engineering Workbench *mfk*. This system is an appropriate tool supporting the designer also in the earlier stages of his work using definable functions and requirements while the extension of the applied product model permits to describe also complex products. Use of a database management system is a key technology for reducing implementation problems. Therefor the goal of a platform strategy, allowing to set up various tools for supporting designer's work, seems to be reachable. This should be a basis for future research in the field of digital mock-up.

References

[1] Pahl, G. and Beitz, W., "Engineering Design", 2nd Edition, Springer Verlag, 1996.

[2] Bachschuster, S. and Meerkamm, H., "Ein Ansatz zur produktspezifischen Erweiterbarkeit eines objektorientierten, wissensbasierten Konstruktionssystems", International Conference on Engineering Design ICED 95, Praha 22.-24.8., 1995.

[3] Weber, A., "Ein relationsbasiertes Datenmodell als Grundlage für die Bauteiltolerierung", dissertation, university of Erlangen-Nuremberg, Erlangen, 1992.

[4] Koch, M., "Konzept und Realisierung eines hybriden Produktmodells", Studienarbeit Universität Erlangen-Nuremberg, Erlangen 1999.

[5] Zwicker, E., "Unterstützung der unternehmensübergreifenden Produktentwicklung durch den Einsatz moderner Informationstechnologien", Dissertation ETH Zürich, VDI Reihe 20, Nr. 288, VDI Verlag, 1999.

Dipl.-Ing. Michael Koch,
Friedrich-Alexander-University Erlangen-Nuremberg, Institute of Engineering Design, Martensstrasse 9, D - 91058 Erlangen, Germany, Phone: +49-9131-852-7984, Fax:+49-9131-852-7988, e-mail: mlkoch@mfk.uni-erlangen.de

Prof. Dr.-Ing. Harald Meerkamm,
Friedrich-Alexander-University Erlangen-Nuremberg, Institute of Engineering Design, Martensstrasse 9, D - 91058 Erlangen, Germany, Phone: +49-9131-852-7985, Fax:+49-9131-852-7988, e-mail: meerkamm@mfk.uni-erlangen.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

TOWARDS 'SKETCH THREE-DIMENSIONAL PROTOTYPING' FOR AIDING CONCEPTUAL FORM DESIGN

J C Borg, K P Camilleri, P Farrugia, F Giannini, and J Muscat

Keywords: Design representation, sketching language, rapid prototyping, modelling

1 Introduction

Despite the progress and sophistication of commercially available CAD tools, designers in industry continue to resort to traditional 'paper-based' sketching for the rapid exploration of component form solution concepts. The research reported in this paper concerns the *on-going* development of an approach framework through which a physical 3-D prototype can be rapidly generated from a 2-D paper-based, plan sketch, of a prismatic component. The goal of this approach is to retain the important characteristics of freehand sketching i.e. pencil and paper, whilst at the same time exploit the benefits of rapid prototyping technology for swiftly generating physical 3D-prototypes of 'component sketches'. The aim of this paper is to disclose the concept and preliminary evaluation results of the *Framework for Early 'Form Design' Rapid Prototyping* upon which this physical prototyping approach is based.

2 Problem background

Component form is one of the five basic characteristics that are used by designers to describe a mechanical artefact [11]. Form is known to have an influence on various performance measures of different product life-phases [2], therefore requiring careful consideration as from 'early design'. Solution modelling assists in elaborating, synthesizing, evaluating and communicating design ideas [1]. Freehand sketching, in which the use of drawing instruments such as a ruler is omitted, is a fast and powerful approach used for rapidly communicating the internal ideas of a designer about an evolving artefact solution during the conceptual design stage. Due to its efficiency in capturing and communicating design intent and visual aspects, freehand sketching is still popular with practising designers. As argued in [12], mechanical engineers are notorious for not being able to think without making 'back-of-the-envelope' paper sketches of rough ideas. At the same time, it is common for designers to alternate between *paper-based sketching* and *physical modelling* when generating solution concepts [1] since geometric prototypes are known to be useful for evaluating fit and form [5]. Although research in computer-based sketching technology [13] is maturing, the use of *pen-and-paper sketching* is still a popular and practical way by which designers freely externalize their thoughts. Traditional CAD tools divert the designers' attention from 'form conceptualising' as they force them to make specific and hence more time-consuming commitments, such as precise dimensions. Industrial designers thus consider CAD tools as too rigid, lacking the fluidity of the pencil sketch. Thus CAD tools are currently more suitable for detailing solutions in the later stages of design. Also, whilst that *rapid prototyping* had made a tremendous impact on the way physical prototypes of solutions are generated in industry, it is essentially employed *late* during design, *after* a detailed component solution has been generated as a 3-D geometric model in a CAD system. The above background indicates that designers would benefit from tools allowing them to rapidly explore alternative, physical *component form* design solutions.

3 'Form concept' rapid prototyping framework

To retain the important characteristics of freehand sketching whilst at the same time exploit the benefits of rapid prototyping technology for generating 3D physical models of 'component plan sketches', a *Framework for early 'form design' Rapid Prototyping* conceived at the Faculty of Engineering, University of Malta, is being developed. As outlined in Figure 1, this prototyping approach framework consists of a number of frames that collectively allow a paper-based sketch to be rapidly reproduced as a physical 3D model.

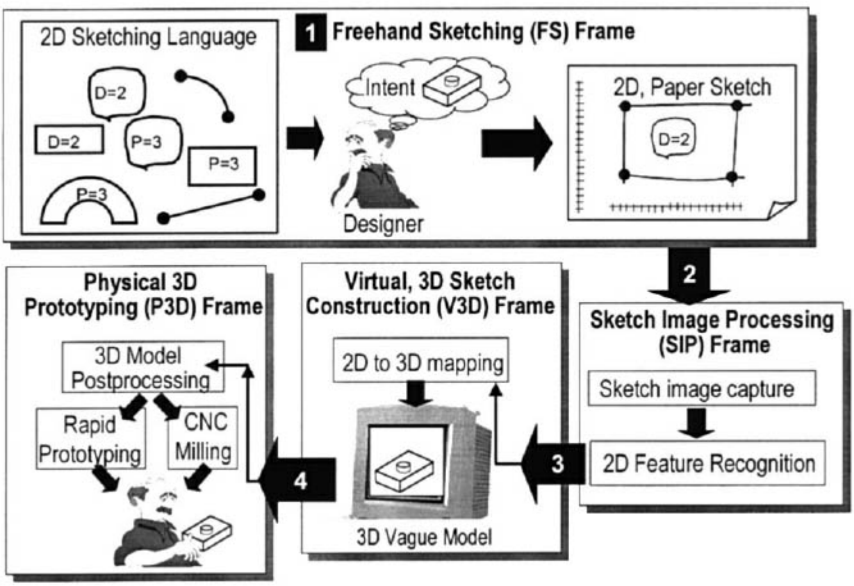


Figure 1. Framework for Early 'Form Design' Rapid Prototyping

Basically, in this approach, by utilizing a specially developed sketching language (Figure 1), designers can externalize their component form design intent onto paper, as a 2-dimensional plan. The *sketch image processing frame* would capture the paper based sketch and convert it into a suitable digital format in which the different form features making up the sketched component are identified. The *Virtual 3D Sketch Construction Frame* will map the recognized features list in the digital format into appropriate 3D features described in a CAD geometric format and compose it into a virtual 3D geometric model that can be manipulated in a CAD system. The *Physical 3D Prototyping Frame* converts the virtual 3D CAD model into a real physical model. The frames collectively making up this approach are detailed next.

3.1 'Freehand sketching' frame

Essentially a *sketch* is a set of vague marks on paper [9] representing various elements. These *sketching elements* may have symbolic meaning (e.g. number 7), geometric meaning (e.g. line) or both. Projection views can be either in 2D or in isometric. In component design practice, sketches lack details such as *hidden lines, material type, material thickness, tolerances* and *dimensions* since their scope is to rapidly capture and model temporary solution ideas that are later worked out in more detail as design progresses. Although sketches convey relevant design information, they can therefore due to such missing information, be a source of misinterpretation as reflected in the example of Figure 2. As in the *Framework* (Figure 1), a paper is a 2D medium used to capture the sketch representing the plan of a 3D prismatic part, a *sketching language* is thus necessary to avoid any misinterpretation resulting with the mapping of a 2D-plan representation into 3D-drawing representation.

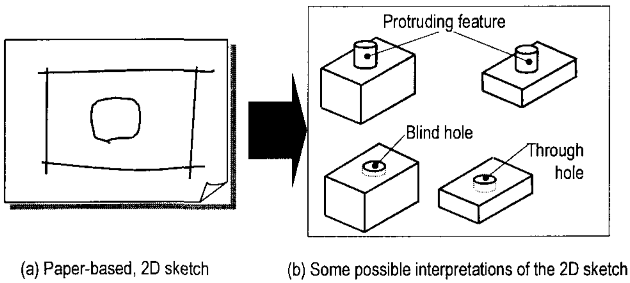


Figure 2. Typical Misinterpretation of The Designers Intent from a 2D-plan based sketch

The 'Freehand Sketching (FS) Frame' consists of a pre-defined *sketching language* that can be employed by *designers* to generate paper-based 2D plan sketches of prismatic components. The major research work involved in this frame is the development of a suitable language. This language is currently in its infancy, limited to describing very simple components such as those shown in Figure 4. Generally speaking, a design language is based on a grammar [8] that has *alphabet, syntax, semantics* and *phonology* as its constituents. In the case of this research, phonology is ignored as the language being developed is intended for written communication. The alphabet in the sketching language consists of a set of *reusable elements (marks)* that represent either geometric entities (e.g. arc) or alphanumeric characters (e.g. a +). Various options (see Figure 3, c-d) were considered for the actual syntax. For instance *alphanumeric characters* are required to associate the sketching entities with certain characteristics (e.g. approximate height), since on paper, only 2D plans of the prismatic parts will be sketched.

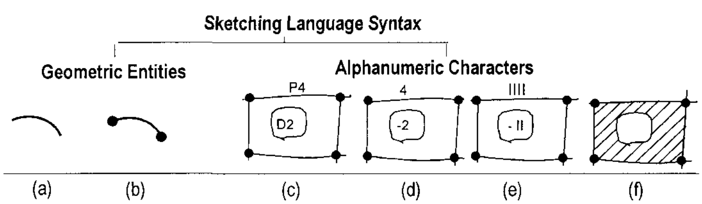


Figure 3. Alternative Language Syntax

One syntax is to utilize letters and numbers to denote the *attribute values* of a feature with respect to a datum. For example *D2* stands for 2 units deep while P4 represents a protrusion of 4 units. Alternatively a negative sign (-) for depth can be employed (e.g. Figure 3d) or dots or vertical lines utilised to replace numbers, similar to bar codes (Figure 3e). Since space is a constraint, this presents practical limitations for situations where for example a designer has to mark a number of vertical lines in a small diameter circle sketch. Alternatively, in order to avoid the use of alphanumeric characters and hence simply image processing, *hatching* was considered to denote protruding features (Figure 3f). The latter method however presents difficulties for components, which are constituted from two or more protruding features such as a stepped boss. It has therefore been established that a suitable representation is to employ a pair of numbers separated by a comma above the geometric entities (Figure 4). The first number represents the absolute, starting Z value of a particular entity above the Z datum, whilst the second number represents the absolute ending Z value of the same entity. The negative (-) character after the second number is employed for *error correction* purposes (e.g. Figure 4 a). Further, for *geometric entities* (Figure 3a), a syntactical rule has been established which confines the designer to make bold dots (see Figure 3b) at every end-point, large enough (about 1mm diameter) for the *SIP frame* to robustly detect vertices.

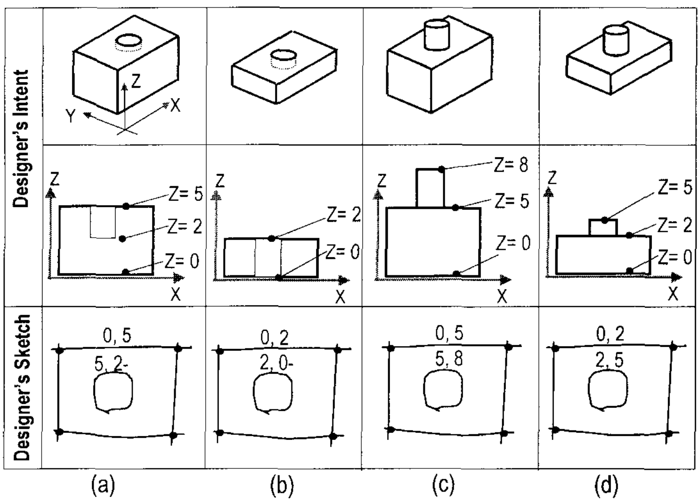


Figure 4. Example of How A Language Reduces Designer's Sketching Intent Misinterpretation

3.2 Sketch 'image processing' frame

The *Sketch Image Processing (SIP) frame* is concerned with *capturing* the paper based sketch into digital form and applying appropriately developed pattern analysis algorithms as in [4; 14]. Its scope is to robustly identify the different form features (e.g. blind hole versus through hole) described via the sketching entities drawn through the sketching language. As outlined in Figure 1, the *SIP frame* produces a description of the 2D-plan sketch in a pre-defined formal description language. This formal language describes each *sketching feature* such that the subsequent frame can process it further.

A major requirement of the sketching language is that it should be robust when used in conjunction with the *SIP frame*. This arises from the fact that individual designers have their own geometric / alphanumeric sketching *style*. Therefore no matter how good a particular image processing algorithm is to recognize say an alphanumeric character, there is still the possibility that certain sketched characters are left unidentified or mixed up.

As a step towards the development of robust *SIP algorithms*, an analysis of freehand circles sketched by a group of females and males having different educational levels and ranging from 18 to 60 years of age was carried out. Figure 5 discloses the example of an *'intended circle'* as sketched by different members of the evaluation team. As reflected, the intended circle differs in *smoothness*, size, *termination* and *ratio of major to minor radii,* this highlighting the necessity of having a robust SIP frame.

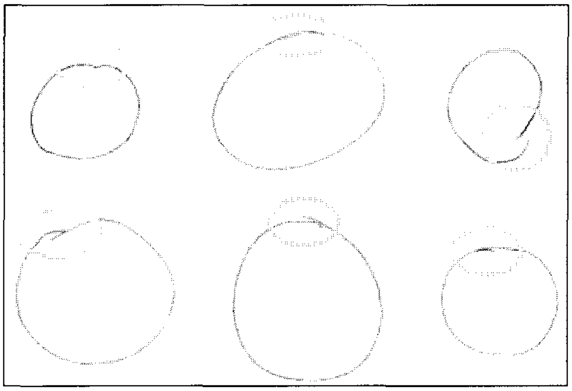


Figure 5. Typical differences in circles sketched by different evaluators

Thus the research work in developing the SIP frame is in itself contributing to the development of the *sketching language*. For instance for robust pattern recognition, the sketching language predefines the style in which a particular alphanumeric character should be manually sketched. In this way the different types of, say, the number '7' associated with alternative designers, when marked on paper will be based on a common *predefined* style (see Figure 6). Although this sacrifices some freedom associated with 'freehand' sketching, it contributes towards robust pattern recognition. In terms of practical acceptance, it is relevant to mention that this concept is widely employed by palm-size computers commercially available on the market.

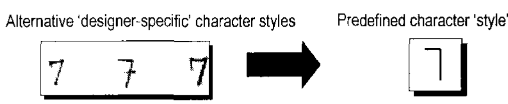


Figure 6. Reducing SIP problems associated with recognizing individual designer styles

For preliminary implementation of the *SIP frame*, paper-based sketches were scanned by a flat-bed scanner at a resolution of 300 dots per inch and saved in *Portable Grey Map (PGM)* format. These were then automatically thresholded to obtain a binary line drawing image. Using the *C* programming language, initial tests were performed by employing the *Hough Transform* [6] algorithm to detect the lines and circles in sketches.

3.3 The virtual, '3D sketch construction' frame

The *Virtual 3D Sketch Construction* (V3D) frame is concerned with *mapping* [13] the recognized 2D features (e.g. blind circular hole) listed in the formal description of the 2D plan sketch, into appropriate 3D features described in a CAD geometric format. This makes it possible to compose a 3D geometric model. Problems inherent in this frame are for instance the definition of certain dimensions such as the height of the prismatic part, which for physical prototyping purposes needs to be specified. This frame is therefore also concerned with modelling *vague* 3D geometric models [10]. As a function, the V3D frame produces a digital description of a 3D geometric model that can be manipulated in a CAD system as illustrated in Figure 1. A set or related rules have been formalized.

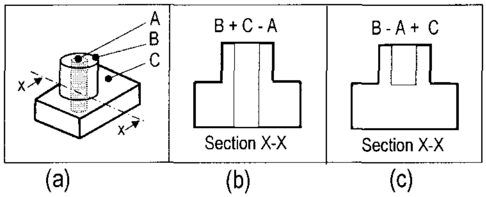


Figure 7. Different Geometric Models Resulting With Different Boolean Operation Sequences

For a rapid evaluation, the *V3D Frame* has been partially implemented using the AutoCAD [3] system. Basically, the 2D features (e.g. a hole) recognized by the *SIP frame* are mapped by the V3D Frame into 3D solid primitives (e.g. a cylinder), the latter formally described in a *command script format* that is automatically processed by AutoCAD to compose the 3D geometric model. Experimentation has revealed that difficulties with the V3D frame relate to automatically determining, the correct sequence of Boolean operations necessary to construct the 3D solid model representing the *intended* sketched component. Consider for instance, the solid primitives A, B, C (see Figure 7a) recognized by the *SIP frame*. If the sequence of Boolean operations employed in the *AutoCAD command script* file is B + C - A, then the resultant component will be that in Figure 7b. If on the other hand, the sequence is B – A + C, then the resultant 3D component will be that in Figure 7c.

3.4 A 'physical 3D prototyping' frame

The *Physical 3D Prototyping (P3D) Frame* is concerned with transforming the 3D geometric CAD model into either *CNC part program* format for processing on a CNC milling machine or into a format that can be processed by a rapid prototyping system, such as *stereolithography (STL)* [5]. In the case of using CNC milling for generating the physical prototype, postprocessing is also concerned with the generation of a suitable process plan for milling the form features making up the prismatic part in a feasible sequence. As an output, this frame produces a 3D physical prototype. For preliminary implementation purposes, the MasterCAM ™ Mill V7 system [7] has been employed to postprocess the sample 3D CAD models into Heidenhain TNC 145 ™ CNC format for machining on a Bridgeport ™ CNC milling machine.

4 Preliminary evaluation of 'form concept' prototyping approach

Before investing further research effort, a preliminary evaluation of the partially implemented framework has been performed. This evaluation was carried out with 6 engineers having a numbers of years (>2) of industrial experience in mechanical component form design. It consisted of feedback about the sketch 3D-prototyping approach, obtained through an explanation of the approach and a structured questionnaire. Key results of the evaluation are presented in tabular form in Table 1.

Table 1. Table – Preliminary Results of 'Early Design' Rapid Prototyping Approach

	Yes	Not Sure	No
1. Do you still consider sketching on paper as an important conceptual form design activity?	✓✓✓✓✓✓		
2. Would you find it useful to be able to generate 'form concept' prototypes from paper-based sketches?	✓✓✓✓✓		✓
3. Would you be prepared to learn & adopt a sketching language?	✓✓✓ ✓	✓✓	

Question 1 discloses that sketching is still an important 'early design' activity. Question 2 indicates that designers consider providing a means of generating physical prototypes from paper-based sketches useful. A reason given is that such a means allows designers to rapidly develop, visualize and share their ideas. Another reason given was that some design concepts are never really prototyped in industry as to do so, time in building a proper CAD model is required. Thus, this approach would reduce such a bottleneck. Finally, question 3 indicates that designers have an inclination towards learning and adopting a sketching language provided it is easy to do so.

5 Conclusions

Traditional *rapid prototyping technology* is known to useful for generating solution models during the later, detailed design stage of components. On the other hand, the framework presented in this paper aims at allowing paper-based plan sketches of prismatic components generated during the early, conceptual design stage, to be rapidly transformed into 3D physical prototypes. The preliminary implementation of this framework has provided an insight into the strengths and limitations of the approach. The problems being exposed with this on-going research indicate that future work is necessary to improve the individual frames employed. For example, there is a need to refine:

- the sketching language by improving its alphabet, semantics and syntactical rules, in order to widen the range of components that can be sketched;

- the routines employed in the *SIP frame* and implement others that cater for an extension in the alphabet;

- The reasoning strategy for combining the separate solid primitives together.

Nevertheless, the partial implementation and preliminary evaluation indicate that the *FS, SIP, V3D* and *P3D* frames collectively contribute an incremental step towards the concept of an 'early design rapid prototyping' approach.

Acknowledgements

The authors are grateful for the contributions to the project made by several students.

References

[1] Andreasen, M. M., "Modelling - The Language of the Designer", Journal of Engineering Design, Vol.5, (2), 1994, p103-115.

[2] Borg, J., X. T. Yan and N. P. Juster, "Guiding Component Form Design Using Decision Consequence Knowledge Support", Artificial Intelligence in Engineering Design and Manufacture, Vol.13, 1999, p387-403.

[3] Burchard, B., D. Pitzer and F. Soen., "Inside AutoCAD 14", New Riders Indiana, 1997.

[4] Durgun, S. B. and B. Ozguc, "Architectural sketch recognition", Architectural Science Review, Vol.33, 1990, p3-16.

[5] Grabowski, H., J. Erb and K. Geiger, "Choosing the Right Rapid Prototyping Technology For Each Phase of Product Development", IFIP WG5.3 Conference on Life-Cycle Modelling for Innovative Products and Processes, Berlin, Germany, 1995, Chapman & Hall, pp.281-292.

[6] Illingworth, J. and J. Kittler, "A Survey of the Hough Transform", Computer Vision, Graphics and Image Processing, Vol.44, 1988, p87-116.

[7] Mastercam., http://www.mastercam.com

[8] Mortensen, N. H., "Design Characteristics as Basis for Design Languages", Proceedings, 11th International Conference on Engineering Design, Tampere, Finland, 1997, Tampere University of Technology, pp.23-30.

[9] Stacey, M., C. Eckert and J. McFadzean, "Sketch Interpretation in Design Communication", Proceedings, 12th International Conference on Engineering Design, Munich, Germany, 1999, Tecnical Univeristy of Munich, pp.923-928.

[10] Stevenson, D. A., X. Guan and A. H. B. Duffy, "Modelling Vague Geometry", Proceedings, 12th International Conference on Engineering Design, Munich, Germany, 1999, Tecnical Univeristy of Munich, pp.1015-1018.

[11] Tjalve, E., "A Short Course in Industrial Design", Newnes-Butterworths London, 1979.

[12] Ullman, D. G., S. Wood and D. Craig, "The Importance of Drawing in the Mechanical Design Process", Computer & Graphics, Vol.14, (2), 1990, p263-274.

[13] Varley, P. A. C. and R. R. Martin, "A System for Constructing Boundary Representation Solid Models from a Two-Dimensional Sketch - Frontal Geometry and Sketch Categorisation", The First Korea-UK Workshop on Geometric Modelling and Computer Graphics 2000, Seoul, Korea, 2000, pp.113-127.

[14] Yu, Y., A. Samal and S. C. Setch, "A System For Recognizing a Large Class of Engineering Drawings", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, (8), 1997, p868-890.

Dr Ing. Jonathan C. Borg
Department of Manufacturing Engineering, Faculty of Engineering,. University of Malta, Msida, MSD06, MALTA. Tel: (+356) 3290-2366 Fax: (+356) 343-577 E-mail: jjborg@eng.um.edu.mt

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

INTELLIGENT COMPUTATIONAL SKETCHING SUPPORT FOR CONCEPTUAL DESIGN

S Lim, A H B Duffy, and B S Lee

Keywords: Computer-aided design, geometric modelling, probabilistic design

1 Introduction

Sketches, with their flexibility and suggestiveness, are in many ways ideal for expressing emerging design concepts. This can be seen from the fact that the process of representing early designs by free-hand drawings was used as far back as in the early 15th century [1]. On the other hand, CAD systems have become widely accepted as an essential design tool in recent years, not least because they provide a base on which design analysis can be carried out. Efficient transfer of sketches into a CAD representation, therefore, is a powerful addition to the designers' armoury.

It has been pointed out by many that a pen-on-paper system is the best tool for sketching. One of the crucial requirements of a computer aided sketching system is its ability to recognise and interpret the elements of sketches. 'Sketch recognition', as it has come to be known, has been widely studied by people working in such fields: as artificial intelligence to human-computer interaction and robotic vision. There are three main issues in adopting sketch recognition for supporting conceptual design activities:

- Sketch capture: Conversion of a bitmap sketch to a computational model through a sketch recognition technique is an important task, currently most relevant to the detailed design stage, and is still a major research issue. During this process all the vagueness contained in the original sketch is removed.

- Interpretation of the sketch structure/content: It is not always easy to interpret the meaning of a sketch correctly, even for example such a basic attribution of 2D or 3D. The confusion between 2D and 3D often occurs from misunderstanding the structure of the sketch.

- Capturing the intended meaning: One sketch stroke can have different intended meanings such as a geometric shape or abstract idea. In the case of a geometric shape, a sketch can only be meaningful in the sense of 2D or 3D geometry. However, in the case of an abstract idea, it could be a symbol, character, functional relationship between entities, concept or idea, i.e., in this case a sketch may have a meaning beyond the simple geometry represented by the sketch strokes.

Despite the continuing efforts to solve the problem of appropriate conceptual design modelling, it is difficult to achieve completely accurate recognition of sketches because usually sketches implicate vague information, and the idiosyncratic expression and understanding differ from each designer.

2 Existing Approaches to Conceptual Design Support

Considerable work has been done to solve various problems encountered in trying to give computational support of design sketching activities (see Table 1). These efforts can be classified into three main areas: geometric modelling, spatial arrangement and design environment support as follows.

Table 1. Summary of various approaches [2] – see references for comments

Area	*Sub-Area*	*Functionality*	*Technology*	*Works*
Geometric Modelling	2-Dimensional Sketching	• Automatic line tidy • Symbol recognition	• Pre-processing / processing • Fuzzification/fuzzy filter	Easel, FFDS, Electronic Cocktail Napkin.
	3-Dimensional Sketching	• Automatic surface creation • 3D sketching environment • Geometric model structure analysis	• Image processing • Sketch interpreter • Direct-manipulation interaction • Sketch interpretation	Akeo, Lipson, HoloSketch, SKETCH, Viking, ISO-Sketcher.
	Sketch recognition	• Symbol/diagram recognition • Hidden line (re)construction • Image retrieval by diagram	• Low level recogniser • Soft constraint • Perceptual analysis	Lamb, IDeS, Electronic Cocktail Napkin, Quick-Sketch.
Spatial Arrangement	Spatial (Re) arrangement and analysis	• Spatial layout analysis	• Goal object(GOB) • Constrained heuristic search • Similarity of spatial pattern	LOOS, SPIDA, ABLOOS, THESYS, WRIGHT.
	Vague spatial relationship modelling	• Location constraints • Uncertain region	• Vague spatial relationship modelling	GEMCON.
Design Environment Support	GUI builder by sketching	• Interface programming without coding	• Storyboard mechanism	SILK.
	Past design retrieval by sketching	• Retrieval by partial design elements • Retrieval by image	• Case based reasoning • Object-oriented programming	Archie-II, A.S.A.
	Front-end system	• Universal modelling interface	• Vague geometric modeller	ARCHPLAN, ISO-Sketcher.

2.1 Geometric Modelling

The work belonging to this category focuses on representing the shape of a design that has been determined during conceptual design. Many researchers in this area have concentrated on shape (re)construction of a rough 2D/3D sketch (be it on a 2D plane), involving a variety of sketch recognition techniques. Although some remarkable techniques have been developed, there are still limitations in applying them to conceptual design work. The basic problem is the requirement of any computational geometric model to be deterministic and precise. It leaves no room for being vague either intentionally or accidentally. When an early idea is 'refined', the original one is simply replaced by the precise geometric model. This can lead to premature loss of vague information and misrepresentation of a designer's intent.

2.2 Spatial Arrangement

Spatial arrangement is concerned with the modelling and analysis of the spatial relationships of geometric models. Guan [3] points out that this approach usually deals with the problem of physical arrangement of objects and spaces to fulfil the needs of various human activities, based on a variety of explicitly or implicitly defined requirements and criteria that usually conflict with one another. This problem is common in many areas, including architectural and mechanical design. The essential issues addressed in this area of work are analysis and modelling based on spatial relationships between geometric models. Most systems attempt to model 2D spatial layout within their specific requirements using techniques such as numerical constraints. Despite the stated aim of supporting the conceptual design activities, most work in the area of spatial arrangement does not serve the purpose particularly well. Most systems feature automatic decision-making techniques in interpreting the geometric information and this alone is insufficient. They tend to concentrate on representing the spatial relationships between 2D rectangular shapes, apart from Kameyama et al. [4] that supports 3D spatial layout. On the other hand, a few systems support vague spatial relationships that can occur in rough geometric sketches. WRIGHT and GEMCON adopted some inequality types (including $=, <, \leq, >, \geq, \approx$, [,]) and linguistic values (including above, below, front, behind, left, right). Thus it is possible to consider relations between design units that do not have fixed locations or fixed dimensions. However, both systems suffer from their inability to be more specific; for example, if $A > B$, by how much.

2.3 Design Environment Support

A number of systems have been developed to support the environment of geometric modelling serving diverse purposes. SILK allows designers to sketch quickly a user interface using an electronic stylus. It recognises 2D sketched shapes and turns these into an active user interface without re-implementation or programming. It also provides a "storyboard" mechanism to test subjects to evaluate the interface in its early, sketchy state. Some serve as a case browser, emphasising presentation of information to users over adaptation or application of past solutions. Yet others were developed to implement a front-end system. For example, ARCHPLAN explores the usefulness of object-oriented programming techniques to support the abstractions of the design process and the resulting design solution. ISO-Sketcher supports an autonomous pseudo-3D sketching environment, linked at run-time to an underlying geometric modeller, GEMCON. ISO-Sketcher is said to offer the early-stage geometry designer an environment supporting a minimum commitment approach, in which the designer is not compelled to make any commitments as to size, location or spatial relationships until desired. However, the system can only deal with the spatial arrangement of vague objects within a limited space and does not provide any effective means to model and manage vague geometric information.

3 Vague Geometric Modelling

The works examined above have made significant contributions in the integration of sketches, either computer-aided or otherwise, in the prevailing CAD environment. One major element, which has received but scant attention so far, is concerned with representing and managing the vague ideas and information contained in sketches. Some researchers have pointed out the importance of dealing with vagueness during conceptual design. For example, Martin [5] argues that a method is required to represent incompletely specified shapes, or indeed, classes of shapes that agree with such an incomplete specification to a greater or lesser extent. Lipson et al. [6] points out the necessity of incorporating analysis tools in conceptual design. Lim et al. [7] proposed "the necessity and methods of the representation and maintenance of vague geometric ideas to support characteristics of conceptual design stage".

In this paper, the word *vagueness* is defined as "the uncertainty about meaning" [8] which "can be represented by a probability distribution over possible meanings". The uncertainty should be modelled in a form which allows the (re)utilisation of the information [8]. To allow a (re)utilisation of the uncertainty, a vague geometric model should include the entire possible range of vague types or values. Based on the definitions of vagueness, a Vague Geometric Model (*VGM*) can be defined as 'a geometric model which implicates ill-defined abstractions, numerical values and/or spatial information'. Furthermore, Vague Geometric Modelling (*VGMing*) can be defined as 'a modelling method that can represent and maintain the vague information contained in a *VGM*'.

Deliberate preservation and handling of vague information is a non-trivial task and represents a departure from current approaches used in sketch recognition. Preserving the vagueness will undoubtedly call for new techniques to represent and model the vague information, which, in our case, is mostly geometrical. It is also in accordance with the principle of minimum commitment, that is, in keeping as many options open for as long as possible. Consequently, a new approach is required for an effective *VGMing* technique that can represent vague information. To represent vague information of a shape, we suggest that the following functions should be satisfied.

- *Alternatives by multiple probabilities of each element*: A sketch stroke can represent multiple alternatives, such as a straight line, curve, or even geometric shape. Hence, the shape type of an object can be changed by a choice of alternatives within each child-element (in this paper, the term 'child-element' is used to explain the hierarchical sub-class of a shape element which could be a rough sketch stroke). This means that the probability of a vague shape can be different depending upon which alternative child-element is chosen. Keeping all the possibilities like this is helpful for discovering hidden features in a representation without being fixated to a single perspective (see [9]). These unexpected discoveries could be more easily found in the original rough sketch when the original ideas are maintained without any refinement (see also [2, 5]).

- *Combination of probabilities of child-elements in a hierarchical structure*: Biederman [10] argued that a set of 'geons' (geometrical ions) can represent a wide range of shape variations. According to his proposal, single geons correspond to elementary shapes (e.g. cylinders, curved cylinders, and bricks) and all shapes can be represented by a combination of geons. As Kavakli [11] argues a significant proportion of drawn parts, whether from memory, imagination or by over-tracing, are produced part by part, which implies that most sketching activities are done by drawing parts of an object. Since an object is drawn by the combination of the parts, a hierarchical structure of these parts could be one way of representation that can be used for *VGMing*. This allows a clearer understanding of the nature of the vague element, and makes it easier to show a combination of vagueness that could be represented as probabilities (see Figure 1). With this structure, therefore, possible alternative interpretations of the element, object and the sketch as a whole can be made more visible [2].

- *Clustering of a vague shape by customised viewpoints*: Classification and clustering is necessary to represent a possible region of vague information. Everitt [12] pointed out that the idea of sorting similar things into categories is clearly a primitive one since classification, in the widest sense, is needed for the development of language. Language consists of some words which help us recognise and communicate the different types of events, objects and people. In order to perform clustering, appropriate criteria are needed. Vague shapes may also be clustered differently depending on different viewpoints [13]. As some researchers [9, 13-15] argued, this kind of vagueness happens frequently when the designer and the user are different. In addition, the criteria could be more abstractly specified if the type of the object (i.e. design concept) is predefined.

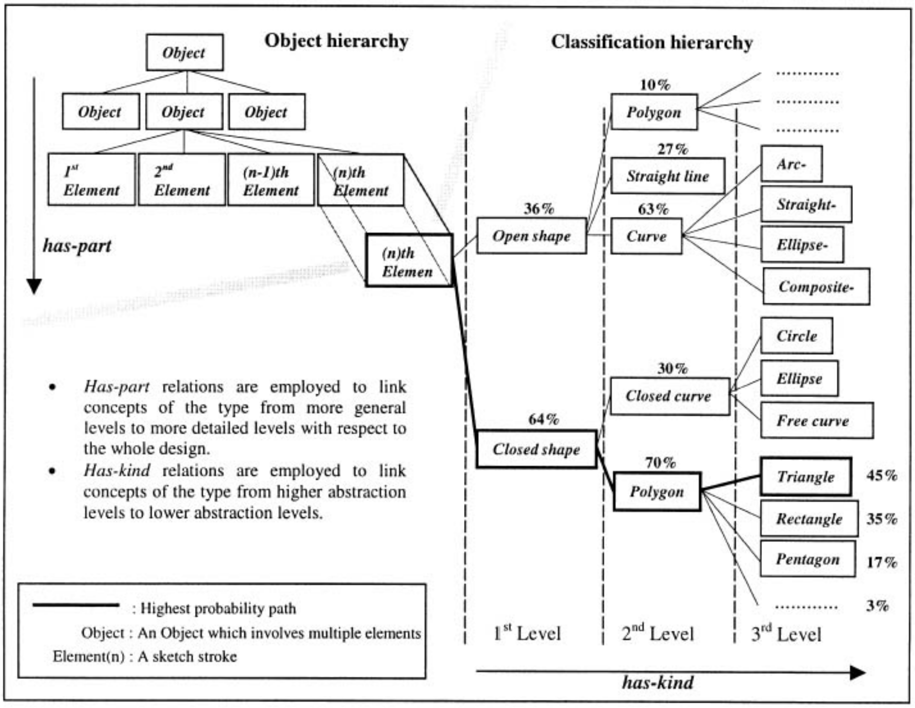


Figure 1. Hierarchical structure of vague information [2]

4 Prototype System

A prototype system, I-MAGI, is currently under development using Macintosh Common Lisp (MCL) 4.0 running under MacOS 8.6. The details of the prototype system process are as follows. Detailed methods are described in [2], thus, only a brief overview is presented here.

- Input: Sketching (draw new stroke) behaviour is similar to a natural sketching behaviour such as pencil-and-paper. Each stroke contains a series of original (x, y) points sampled along a path drawn by the designer. When drawing a stroke, the sample data points of the stroke are stored. If the drawing tool (e.g. mouse button or digital pencil) is released, the combination of line segments is considered as one stroke. During sketching, a designer can group strokes together to form compound strokes, which are then treated as a single element of the object.

- Pre-processor: Each new stroke entered is first sent to the pre-processor, which initially identifies the potential vertices and self-intersection points. These points are given probabilities of being vertices of a polygon according to their spatial relationships with the neighbouring points of the stroke. First, each connected four line segments, which are created by the five original points, are sequentially analysed (we use the term line-unit to denote these bunch of line segments, Figure 2). A line-unit is used to identify the probability of the third point of each line-unit. Basically, the pre-processor identifies all the potential vertices and intersection points apart from unintentional vertices because of a designer's hand vibration (i.e. hand shaking). The pre-processor checks the type of one line-unit, and if the shape type of line-unit is 'VV' or '_/_', then it is not considered as potential vertices. Second, the pre-processor checks each pair of line segments in a stroke respectively to find the intersection points (see Figure 3). Figure 4 shows an example of possible vertices and self-intersection points with associated probabilities. The initial numbers of vertex (*ver-num*) and self-intersection (*sel-num*), (x, y) coordinates, and probabilities are identified.

- Processor: The above vertex probability information is then sent stroke by stroke to the processor, which analyses the probabilities of each stroke representing the primitive elements which are pre-defined by the system (see the primitive elements of third hierarchical level in Figure 1).

- Post-processor: When the designer has defined one object containing a number of strokes, these processed strokes and the probabilities are sent to the post-processor which carries out the three tasks sequentially. Firstly, the size of the input object is now expanded as 3D co-ordinates if applicable. Second, the vague relative spatial relationship is analysed by the combination of distance and direction when there are more than two objects. Third, the *VGM*, which is created through the process, is clustered by customised viewpoints.

- Output: Finally, the designer gets a *VGM* and various alternatives associated with vague information.

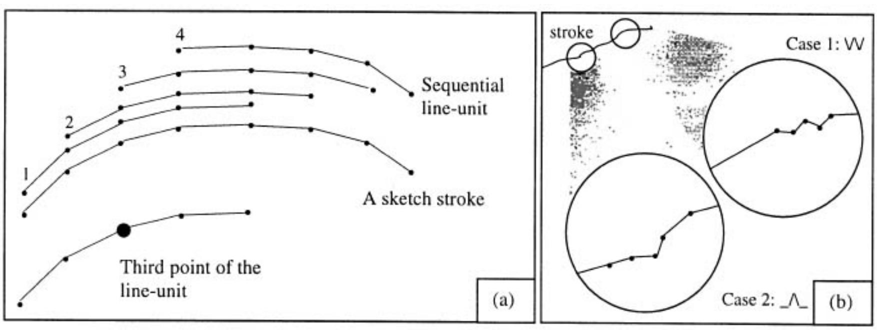
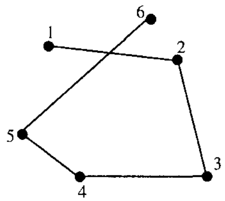


Figure 2. (a) Sequential capturing of the line-unit, and the third point of the line-unit. (b) The exception to identify the potential vertices




```
for i = 1 to n
   for j = (i + 1) to n
      Find-intersection (L_i(i+1), L_j(j+1))
   end
end
```

n: Total number of point in the stroke (i.e., $n = 6$).
L_{ij}: Line segment between i and j.

Figure 3. General algorithm to find intersection point

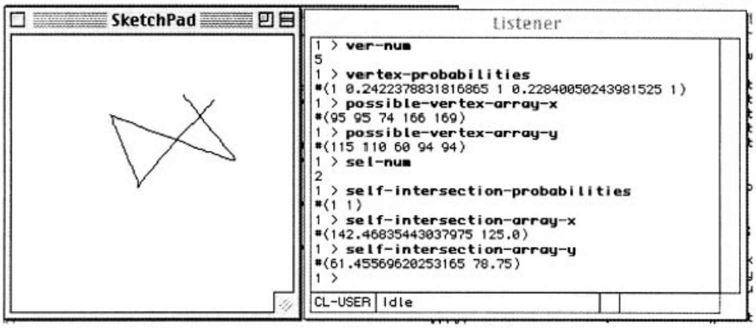


Figure 4. The probabilities of the vertices and intersection points from the sample stroke

5 Conclusion

Visualisation of conceptual models, in the form of sketches, is important for a more flexible and dynamic design. Sketches often play an important role in concept generation in design. Consequently, to capture, model and more fully utilise these concepts we need to integrate the sketching activities into the CAD environment. Much work to date on sketch recognition and computer-aided sketching systems support the need to provide computational based sketching. However, current approaches do not offer any method of representing and managing vague information often found in conceptual ideas as reflected in sketches.

In this paper, the existing approaches and representative works with sketching to conceptual design have been reviewed. A possible approach to vague geometric modelling based on a hierarchical structure and probabilistic method, and their associated issues have been discussed with the prototype system I-MAGI. The new approach will support minimum commitment by: a) modelling of vague shape itself; and b) maintaining the vagueness relating to the interpretation of shape rather than fixing upon a particular shape.

Currently, the prototype system is under development.

References

[1] Jenkins, D.L. and R.R. Martin, "The importance of free-hand sketching in conceptual design: automatic sketch input", American society of mechanical division, 53: 1993, p115-128

[2] Lim, S., B.S. Lee, and A. Duffy, "Incremental modelling of ambiguous geometric ideas (I-MAGI)", Special Issue on Conceptual Modeling in Design Computing of the International Journal of Artificial Intelligence in Engineering, 2001. (to be published)

[3] Guan, X., "Computational support for early geometric design", in CAD centre, Dept. of DMEM, Strathclyde University, Glasgow, 1994, p275

[4] Kameyama, K.-I., K. Kondo, and K. Ohtomi, "An Intelligence layout CAD system for industrial plants", Design Theory and Methodology (DTM'90), 1990

[5] Martin, R.R., "Modelling inexact shapes with fuzzy sets: Fuzzy-Set-theoretic Solid Modelling", in Proceedings CSG'94, 1994

[6] Lipson, H., "An interface for 3D conceptual design based on freehand sketching", IFIP WG5.2 Workshop on Geometric modeling in Computer Aided Design, 1996

[7] Lim, S., A. Duffy, and B. Lee, "Incremental Modelling of Ambiguous Geometric ideas (I-MAGI)", Advances in Conceptual Modeling in Design, AID00, 2000

[8] Cheeseman, P., "Probabilistic versus fuzzy reasoning", Uncertainty in artificial intelligence, 1986, p85-102

[9] Suwa, M., J. Gero, and T. Purcell. "Unexpected discoveries: How designers discover hidden features in sketches", Visual and Spatial Reasoning in Design, 1999

[10] Biederman, I., "Recognition-by-components: A theory of human image understanding", Psychological review, 94: 1987, p115-147

[11] Kavakli, M., S.A.R. Scrivener, and L.J. Ball, "Structure in idea sketching behaviour", Design Studies, 19(4): 1998, p485-518

[12] Everitt, B.S., "Cluster Analysis: Third Edition", 1993

[13] Duffy, A.H.B. and S.M. Kerr, "Customised Perspectives of past designs from automated group rationalisations", International Journal of Artificial Intelligence in Engineering, Special Issue on Machine Learning in Design, 8(3): 1993, p183-200

[14] Fisher, D., L. Xu, and N. Zard. "Ordering effects in clustering", In Proceedings of the Ninth International Conference on Machine Learning, 1992

[15] Howard-Jones, P.A., "The variation of ideational productivity over short timescales and the influence of an instructional strategy to defocus attention", in Proceedings of Twentieth Annual Meeting of the Cognitive Science Society, 1998.

S W Lim
CAD Centre, Dept. of DMEM, University of Strathclyde, James Weir building, Glasgow, Scotland, UK, G1 1XJ, UK. Tel: 44-(0)141-548-2374, Fax: 44-(0)141-552-0557, Email:sungwoo@cad.strath.ac.uk, URL:http://www.cad.strath.ac.uk/~sungwoo/Home.html

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

SKETCHING BEHAVIOUR AND CREATIVITY IN CONCEPTUAL ENGINEERING DESIGN

M Pache, A Römer, U Lindemann, and W Hacker

Keywords: conceptual design, sketching, creativity, CAD

1 Introduction

Product development is – like all processes in the context of trade and industry – subjected to the rules of efficiency and effectiveness. These days, the increasing demands from customers and society result in constant rising complexity of products and processes, ever tighter time constraints and nevertheless, the pressure to develop innovative products to open up and satisfy new markets. Although the probability for a successful introduction of a new product is rather low, prosperous product innovations can result into a market leadership that can last for years. Well known examples are the Sony Walkman and the foldable hardtop of the Mercedes Benz SLK. There are numerous requirements a company has to comply with in order to successfully develop, produce and introduce an innovative product. Still, the basis for such success is the creativity of designers, who assign the characteristics of the following processes and lately of the resulting product from their ideas. The occurrence of creative ideas in design is strongly connected to the medium used by designers to externalise and further develop their ideas as mainly research in other design domains has shown.

The early stages of an engineering design process probably have the most significant impact on the characteristics of a future product and its manufacturing process. Early design stages are characterised by the occurrence of abstract and incomplete solution ideas, which represent a broad variety of potential concrete solutions. Within these stages, many different abstract solution can appear to be promising, but cannot be evaluated yet. Although CAD-systems are widely used by designers even in early design stages, classical sketches are still used to support CAD-work [1, 2]. Therefore, future CAD-systems should combine the functionality of existing systems with benefits provided by classical freehand sketches. An essential prerequisite to develop such systems without restricting designers' creativity is to know the basic mechanisms of creativity during freehand sketching.

In this paper an empirical investigation on conceptual design processes via freehand sketches is presented. From 90 design experiments that were carried out, results regarding the sketching behaviour are derived by an overall analysis. Additionally, a case study is presented, that shows a potential mechanism of creativity when sketches are used in engineering design. Unfortunately, this mechanism could not be verified statistically due to the rareness of occurrence within the case study. Still, the case study is supposed to provide clues for generation of hypotheses in engineering design research.

2 Research Method

90 design experiments were carried out with 30 engineering design students and 15 professional designers, each processing two conceptual design tasks that differed in their complexity and, therefore, their hypothetical demand for using sketches.

2.1 Experimental design

The general conditions provided by the experiment were designed to be as close to industrial reality as possible. Therefore no "thinking aloud" was demanded from the proband, likewise the choice of tools was free, except that no CAD-system was available. Besides the artificial laboratory situation, this was the only (but still important) restriction within the experiment design. Nevertheless, it was necessary since the status of familiarity with the system would have strongly influenced the solution process. From survey results [1, 2] it could be proposed that sketches are widely used for conceptual design. Therefore this restriction was accepted. The design process was recorded by audio and video equipment and the proband's activity was on-line-documented by means of a computer-based protocol. The protocol differentiated between the following categories of action: writing, reading, sketching, erasing, regarding (a sketch), building physical models, analysing physical models, talking, gesturing and "staring into space" respectively action that could not unequivocal be assigned to one of the other categories.

Furthermore the sketches were analysed regarding their level of geometrical abstraction, the occurrence of elements representing functional properties and use of written words and elements with symbolic character. Elements were regarded as geometrically abstract if they were strongly simplified in their shape, such as lines that were meant to represent volumes. Symbols used were for example arrows and other elements known from engineering design methodology (e.g. dots to represent joints). Functional properties were for example curves to represent the run of a part's motion or repeatedly drawn parts to represent several states of a motion. Personal data were gathered, i.e. experience in design, design education, ability for spatial imagination [3] and heuristic competence [4]. After the experiment, the probands were interviewed regarding their design process and the solution they developed.

Additional to the analysis regarding the whole number of experiments, a case study of two corner cases was carried out on the mechanisms of idea generation. Therefore two designers were analysed regarding the conditions under which new solution ideas occurred and were introduced to the solution concept so far. Protocols on the basis of the video were kept for the case study regarding the solution progress, problem solving steps and the partial problem (respectively solution) dealt with.

The first task was to develop a conceptual solution for a barbecue device, that allows to adjust the distance between coal and meat in a stepless manner. Furthermore the design concept had to meet numerous requirements regarding ergonomics, safety, reliability, cost, etc. In the second task, the probands were asked to design the conceptual solution for a device to laser weld sheet metals in automotive industry. As boundary conditions, shape and dimension of the sheet metal were given, as well as the laser and a linear motor for actuation. The proband had to develop a device that makes the laser move with a defined speed and a defined distance along the sheet metal, so the laser beam always hit the metal's surface right angled. Since the sheet metal was shaped rather complex, some of the main problems were to guide the laser along the complex shape of the sheet metal and to transmit the linear actuation from the motor to the laser (see figure 1).

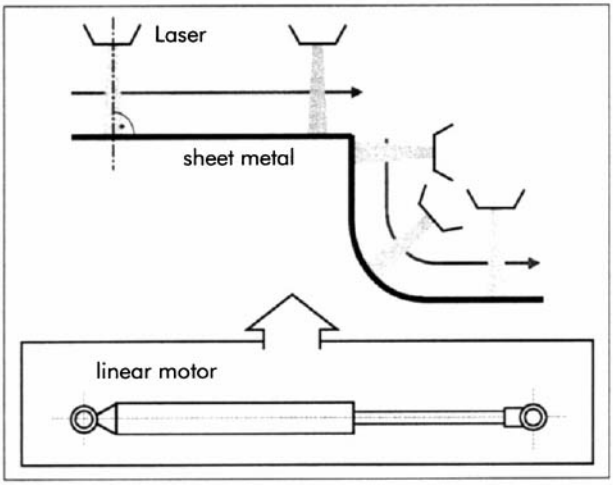


Figure 1. Design task to develop a device for laser welding of sheet metals

2.2 Theoretical background

Due to the limitation of the human "working memory", sketches may support the designer at least in a memory relieving way. Still, it is assumed that sketches are capable of fulfilling functions that are beyond "simple" memory relief [2, 5]. A survey among 106 mechanical designers has shown that especially sketches are still widely used before and even during CAD-work. Since more than 90% of the designers report "developing solutions" (besides "supporting communication") as their intention for using sketches, it may be presumed that CAD-systems lack the functionality in properly supporting the designer in developing design solutions. This belief is corroborated by further results. 75% of the designers reported the occurrence of new solution ideas as one of the main advantages from the use of sketches [2].

Creativity, as it is demanded from designers, needs to be targeted on a certain, desired terminal state, since the resulting product needs to meet very strict requirements. These requirements usually refer to the function of the product, so the designer cannot derive the product's definition directly from those requirements. Therefore, being creative in engineering design is strongly related to problem solving. Problem solving may be processed by conducting a "dialogue with oneself", for example described in the "ARASKAM"-procedure by Doerner [6]. During this dialogue, the designer is switching between different levels of abstraction and complexity.

Similar to the dialogue described by Doerner, Schoen [7] describes professionals conducting dialogues with given situations as a source of creativity. Especially in the case of design, Schoen describes designers having reflective conversations with their own sketches, in which they externalise their vague ideas and in turn find useful clues for further refinement. This dialogue has been specified in several cases, mostly by pointing out that a designer may reinterpret sketches by simultaneously attending previously sketched elements that have not been attended together before. So the designer can discover "hidden" features, configurations and relationships of which he or she was not aware before [8, 9]

Arnheim [10] resumes several mechanisms of visual perception that alter and manipulate shapes and colours depending on the circumstances of presentation to the viewer (such as switching between different views when objects are spatially ambiguous, completion of uncompleted shapes and combining single elements to superordinated shapes). He emphasises that these mechanisms are constantly adapted and he even assigns "intelligence" to those mechanisms due to the cognitive functions they fulfil.

3 Results

Analysing the activity of the probands, it has shown that the rates of sketching activity in relation to the overall process strongly differ from each other for different probands. For sake of clearness, only the results for the professional designers are charted in figure 2. In case of the professional designers, the rate of average sketching activity (referring to both design tasks) ranges from about 17 % to 44 %. The results for the students are similar, in fact the lowest average value is about 7 % in case of the student probands.

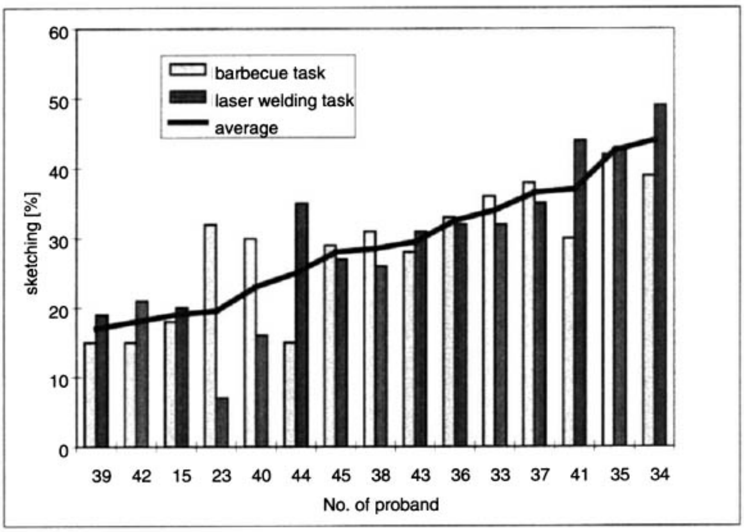


Figure 2. Rate of sketching for professional designers

From this point of view, the probands no. 39 (average sketching activity 17 %) and no. 34 (average sketching activity 44 %) appear to be corner cases with regard to their sketching behaviour. Considering the distribution of all actions on the overall process of processing the laser device task, it shows that the activity profiles for these corner cases strongly differ (see figure 3). Since even for the same task and under the same conditions, sketching processes differ that much, it seems that sketching behaviour is strongly influenced by personal variables. Still both probands work with the sketch for about 75 % of the time. Therefore it should not be supposed that proband no. 39 did not benefit from his sketches during the solution development process.

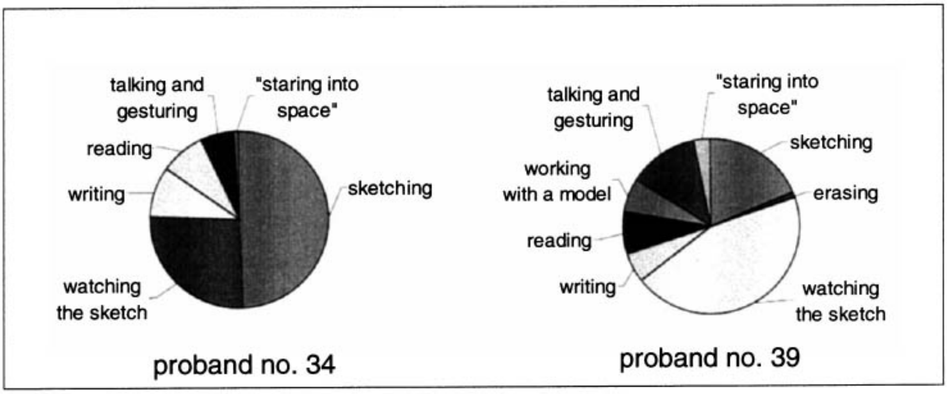


Figure 3. Activities during processing the laser welding task for proband no. 34 and 39 (corner cases)

Keeping in mind the description of problem solving processes by Doerner ("ARASKAM"-process, see above) [6], abstract predefinitions in a sketch might support this process. Therefore, the sketches used by the probands during processing the laser weld task were analysed with regard to the modality of depictions. It has shown, that for the laser weld task, about 73 % of the probands used schematic depictions during solution development. 7 % did even use schematic depictions only to completely develop a conceptual solution for the task. About 93 % of the probands depicted functional properties of the solution concept. Since the task was mainly a kinematic problem, most of those depictions were arrows and curves representing motion of parts. All of the probands used symbolic elements within the sketch and 89 % used written terms.

From these results, it can be derived that sketches are capable of offering elements to the user, that provide a broad variety of different levels of abstraction. How does the modality of elements influence the process of developing ideas in engineering design? In the following, we will describe a sequence that has taken place during an engineering design experiment with a student of mechanical engineering working on an engineering design task. This sequence has taken place early within the design experiment, shortly after a first solution approach was developed. The desired motion of the laser was given by the shape of the sheet metal as described in the formulation of the task. The student decided to achieve the motion of the laser by two tracks that were shaped just like the curve which the laser had to move along. The Laser was fixed to a "crab" that ran in these tracks. Thereby the laser perforce had to move along the shape of the tracks. This solution is comparable with the principle of a monorail or a roller coaster. The student sketched this approach by using simple curves that represented the tracks and a "cart" with wheels, which represented the crab (drawn twice to point out its motion). The Laser, connected to the crab by a line, was represented by a simplified shape (see figure 4, state 1).

Still, this approach did not cover the problem of actuation since the laser had to move with a certain speed along the tracks. To actuate the crab, only a linear drive was available which accordingly just provided a linear motion. Therefore, this linear motion needed to be transmitted into the rather complex motion of the crab. The student then clarified the actuation problem by sketching the speed vector of the crab's motion, represented by an arrow attached to the crab pointing in the direction of the motion. He then added the horizontal and vertical portion of the speed vector, forming a vector triangle (see figure 4, state 2).

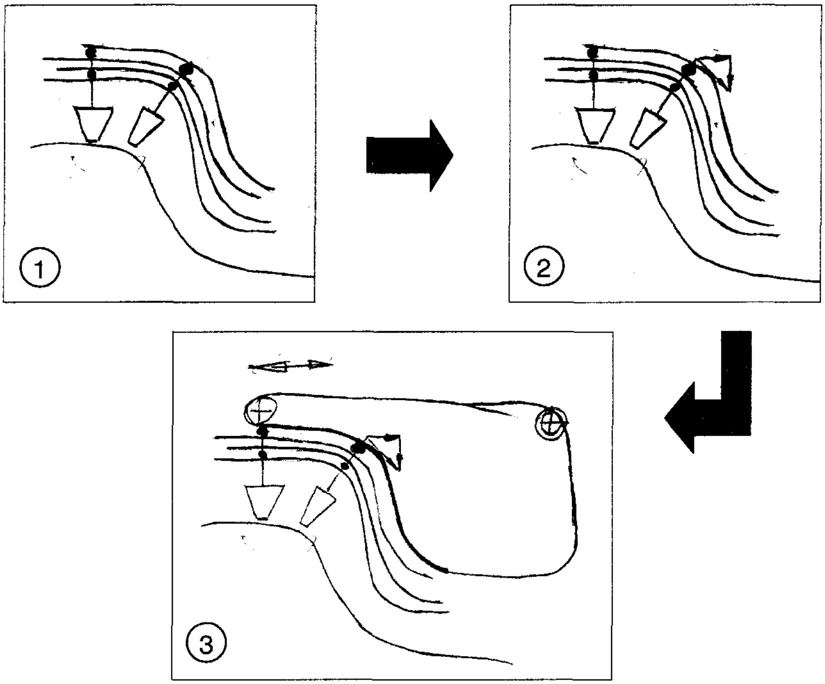


Figure 4. Different states of the sketch when a new idea arose to realise a defined motion of the laser

After regarding the sketch (for a repeated time), he drew one of the track curves (the one that was closest to the vector of the crab) again right over the existing one and thereby intensified it, but did not alter its shape (At this point, it is important to mention that the motion of the student's hand along the track curve matched the direction in which the crab was supposed to move along the tracks). Afterwards, he intensified the track curve again in a slow manner. Now, without removing the pencil from the sheet, he drew beyond the existing shape and completed the curve to a loop. He added pulleys, represented by simple circles, to some corners of the loop and attached the laser directly to the loop (see figure 4, state 3). Therefore, by adding few lines, the student altered the solution principle from a crab running in tracks into a circulating belt with the laser attached to it. Since a belt is not stable in shape, it is forced by the pulleys to move along the curve. Within the described sequence, the design concept was changed significantly. So what might have happened within these minutes?

Premising that the change of concept has been initialised externally by the sketching process, there are two approaches to explain this process which correspond with the findings from other research. One approach is to assume that the speed vector of the crab and the track curve have been attended together and have been correlated. Each element has been drawn independently and without a direct, contextual relation to each other since the vector represented the motion of the crab. The student may have referred this vector to the track and interpreted the combination of both elements as a curved shape that moves "in itself", which may be a belt or a chain. Furthermore, this assumption is corroborated by the fact that from all curves representing the tracks, the one that is closest to the vector has been chosen for modification (although all curves have the same shape). It is probable a viewer can correlate elements more easily when they are spatially close to each other. The discovery of this relation may have been actively supported by the student himself in intensifying the track curve, which highlights this element since its contrast to the surrounding is increased.

This approach to explaining this sketching sequence corresponds with the unexpected discovery of relations between elements described by Suwa et al. [9]. Still, in this case, the discovered relation has altered the meaning of the sketch radically. This might be due to *what* has been represented in this sketch. Even the few elements in this sketch contain enormous information beyond representation of geometry. The track curve, a simple line, has geometric and symbolic character. Its shape roughly determines the geometry of the track. But why is it recognised as a track at all? Only because of the existence of the representation of the crab, the curve gains a contextual meaning and is recognised as a track. In correlation with the vector, the curve gains a completely new contextual meaning. The discovery of this sort of relation between elements, therefore, can result in a new interpretation of the contextual meaning of the elements themselves. The fact that the representation of the belt was developed directly out of the track curve by completing the loop, shows that this process is possible, although it cannot be verified in this experimental setting.

Another approach to explain the events within the sequence of sketching is feasible, which we will only discuss briefly. This approach is offered by the way the track curve has been intensified before its modification to a loop. When the student intensified the track curve, the motion of his hand matched the direction in which the laser was supposed to move. It is possible that the perception of the motion has been correlated to the curve, again resulting in a shape that moves in itself. This perception may not only have been visual, a kinaesthetic perception is possible as well. Since the design task processed by the student is mostly a kinematic problem, this approach should be taken into account. However, such a mechanism of reinterpretation is supposed to occur less frequently since the motion is only perceptible right at the moment it takes place. It may leave a "slight trace" of remembrance, but this probably will fade soon after the event. In contrast to the perception of motion, symbols such as the arrow representing the speed vector are permanently perceptible. Arnheim [10] points out that especially the visual sense tends to reinterpret the perceived picture permanently. The more monotonous the picture is, the more it is reinterpreted by visual perception. This effect probably even will enforce the discovery of relations between depicted elements.

The mechanisms proposed before could unfortunately not be verified in this experimental setting. The student was not told to "think aloud", therefore design events can only be identified by observation of the sketching process. Unequivocal conceptual changes have occurred so rarely that they have eluded from statistical analysis. Still, the proposed mechanisms do not contradict the findings from other design domains. In mechanical Engineering, reinterpretations on a "rather simple" geometrical level may have a major impact on the underlying contextual meaning of the sketch and, thereupon, may result in new conceptual solutions.

4 Conclusions

Freehand sketches provide chances for creative and successful problem solving in conceptual design. A sketch is capable of not only representing technical components, but also functional features of a solution concept, such as motions. An important requirement on the sketching mode of future CAD-systems is to support designers in using geometrical, symbolic and textual predefinitions on different levels of abstraction. The broad variety in modality and levels of abstraction within a sketch enables a designer to externalise an enormous amount of information and again receive from the sketch. When elements that were depicted independently from each other are attended simultaneously, it may appear that these elements are correlated by the designer. Since the contextual meaning of a single element depends on its relation to the other elements, this new interpretation can change an element's meaning completely. The mechanism proposed in this paper could unfortunately not be verified due to the applied experimental setting. Still, these mechanisms are capable of having a major impact on a conceptual design process, therefore engineering design methodology should pay great attention to those mechanisms, even if they turn out to appear rarely. Since sketching behaviour is personally influenced and strongly differs, future CAD-systems should mainly *offer opportunities for different kinds of proceeding, instead of constraining the designer.*

5 Acknowledgements

This project is granted by the funds of the German Federal Science Foundation (Deutsche Forschungsgemeinschaft).

References

[1] Pache, M., Weißhahn, G., Roemer, A., Lindemann, U. and Hacker, W., "Effort-saving modeling in early stages of the design process", Proc. ICED 99, Technische Universität, München, 1999, pp. 911-916

[2] Roemer, A., Pache, M., Weißhahn, G., Lindemann, U., and Hacker, W., "Effort-Saving Product Representations in Design – Results of a Questionnaire Survey", Design Studies, in press, Vol. 22, 2001.

[3] Gittler, B., "3-D-Würfeltest. Verfahren zur Erfassung des räumlichen Vorstellungsvermögens", Beltz, Weilheim, 1990.

[4] Staeudel, T., "Der Kompetenzfragebogen", Diagnostica, 34(2), 1988, p136–148.

[5] Hacker, W., Sachse, P. and Schroda, F., "Design thinking – possible ways to successful solutions in product development", Proc. Designers – the key to successful product development, Technische Universität Darmstadt, 1998, pp. 205-216.

[6] Doerner, D., "Bauplan für eine Seele", Rowohlt, Reinbek, 1999.

[7] Schoen, D. A., "The Reflective Practicioner", Basic Books, New York, 1983.

[8] Goldschmidt, G., "The backtalk of self-generated sketches", Proc. Visual and Spatial Reasoning in Design, Key Centre of Design Computing and Cognition, University of Sydney, Sydney, 1999, pp. 163-184.

[9] Suwa, M., Gero, J. and Purcell, T., "Unexpected discoveries: How designers discover hidden features in sketches", Proc. Visual and Spatial Reasoning in Design, Key Centre of Design Computing and Cognition, University of Sydney, 1999, pp. 145-162.

[10] Arnheim, R, "Anschauliches Denken", Du Mont, Köln, 1977.

Martin Pache
Technische Universität München, Lehrstuhl für Produktentwicklung, Boltzmannstr. 15, 85748 München, Germany, Tel: +49-89-289-15135, Fax: +49-89-289-15144, pache@pe.mw.tum.de, www.pe.mw.tum.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

KNOWLEDGE REPRESENTATION AND PROCESSING IN ABSTRACT PROTOTYPING OF DESIGN SUPPORT TOOLS

E Z Opiyo, I Horváth, and J S M Vergeest

Keywords: Prototyping, quality of design support software tools, knowledge representation and processing, pre-implementation testing.

1 Introduction

Testing of the proposed design support tools to requirements prior to implementation by involving representatives of various stakeholders can contribute significantly toward improving their quality. CMM [1] and usability tests [2] are examples of techniques for software quality and process management that embrace aspects of pre-implementation testing. Apparently, most of the existing techniques do not cover all phases of development. Our work has resulted into the creation of a comprehensive pre-implementation testing methodology that has been named abstract prototyping. This technique ([3], [4]) is stakeholders oriented in the sense that the representatives of various stakeholders of design support software tools are systematically involved in all phases of the development processes as the evaluation subjects. This technique emphasizes on testing of incidental abstract implementations. Tests are carried out at four levels of abstraction i.e. at theories, methods and algorithm levels, and after pilot implementation (Figure 1). We refer to the representations of theories, methods, algorithms and pilot implementations as *abstract prototypes*, *abstract software implementations* or simply as *implementations*. We recognise these implementations as testable implementations of design support software tools and requirements are the evaluation criteria. The form of representation of these implementations may include, e.g., text description, mathematical equations, sketches, reference to articles, system architectures, pseudo codes and executable programs. One of the important issues in abstract prototyping is to make subjects knowledgeable about the abstract software implementations. Thus, there is a challenge of letting the subjects, including rather naïve subjects from outside the development team, be well informed about the abstract software implementations and the concepts behind them. This requires the implementations in question to be represented and availed to the subjects, in any geographical location, in an intuitively precise and expressive way.

Several technologies for handling knowledge and supporting collaboration in geographically separated sites are available. These include, for example, TeamRoom [5], IBM's Lotus Notes [6], Microsoft's NetMeeting [7], and TeamWave [8]. Typical features of these tools include, for instance, user interface with collaboration options; real-time conversations via text; shared whiteboard; shared files, URL links, and images; point-to-point audio and video links; e-mail facilities that enables users to send notices; and status tracking of documents. There are also many research works on the development of methods and tools for supporting knowledge handling (i.e. capturing and accessing information) and collaboration during the development processes. For instance, Tomek and Giles [9] describe a virtual environment to support software development teams in geographically separated locations. The available technologies have been proved to be quite useful, although it is argued that none of them comes close to emulating the conditions of work in one shared physical location [9]. However, there are several useful features that can be adapted and enhanced to support knowledge representation and processing in abstract prototyping of design support tools.

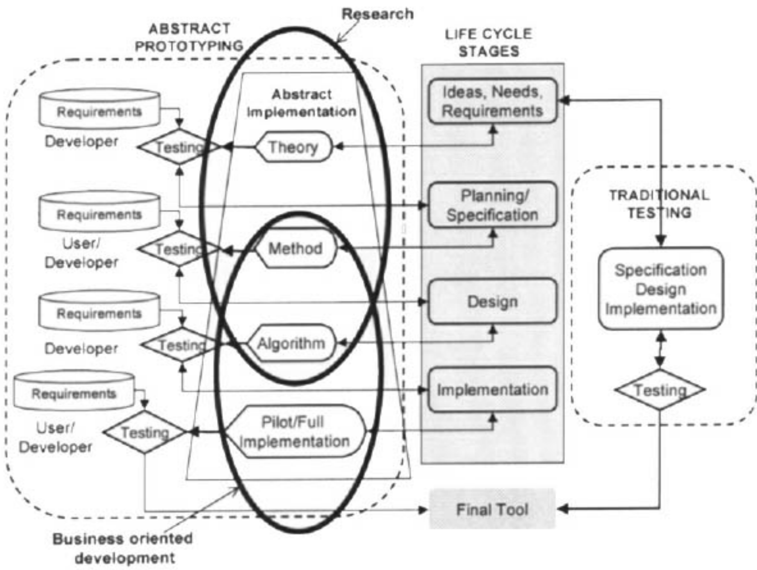


Figure 1. Levels of abstraction of design support tools

As an attempt to deal with the challenge of handling knowledge during the pre-implementation testing of design support tools, we have developed a computer-based knowledge representation and processing utility. Its main function is to support capturing of knowledge and to provide the developers and the stakeholders with access to the required knowledge, regardless of their geographical locations. The philosophy behind this approach is to enrich the stakeholders with knowledge about the abstract software implementations, in particular the underlying theories and methods, thus enabling them to make right judgements. In this paper, we first present the theoretical fundamentals of knowledge representation and processing in abstract prototyping. Then, we focus on knowledge representation and processing. We describe the methodology and the prototype software tools for representation and processing of knowledge. We will also present an example on the use of these software tools in the process of development of a computer aided conceptual design (CACD) system.

2 Theoretical Fundamentals

In this section we outline the basic theoretical concepts of knowledge representation and processing in abstract prototyping. We describe the relationships between requirements and abstract implementation, the complex links that exist among the abstract prototypes, and explain the roles that knowledge of abstract implementation play in pre-implementation testing.

2.1 Requirements, Abstract Software and Expected Functionality

Figure 2 illustrates our interpretation of the transformations that occur between elicitation of requirements and realization of the final design support software product. There are direct relationships among the elements of the abstract software implementations (i.e. theories, methods, algorithms and pilot implementations) and the expected functionality. However, there are indirect correlations among requirements and methods, algorithms or pilot implementations. Similarly indirect correlation exist between theories, methods or algorithms and the expected functionality and the requirements.

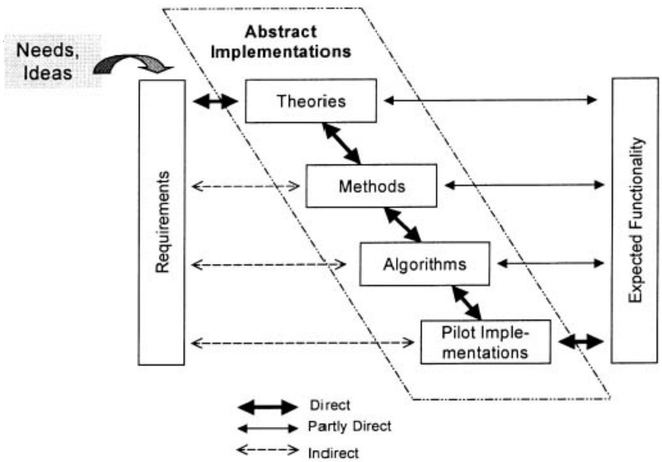


Figure 2. Transformations from requirements to expected functionality

Generally, it is difficult to jump directly into the development of methods, algorithms or pilot implementations based on the understanding of requirements (needs) only, without building the foundation on a theory or a set of theories. Likewise, it is difficult to implement high quality functionality from scratch by not going through the processes of development and testing of theories, methods and algorithms. A purposefully arranged (structured) set of theories is a prototype of the highest abstraction level. Similarly, methods, algorithms and pilot implementations are prototypes of different degrees of abstraction. However, it is important to emphasize that the abstract prototype models (i.e. the representations of theories, methods, algorithms and pilot implementations) are not the replacements of good requirements. Rather, they are dependent on requirements. Conversely, the requirements can be significantly improved through the elaboration processes that involve modelling and evaluation of the elements of abstract prototypes. When gaps in the abstract software implementations are identified questions need to be asked and the answers are fed back into the requirements model (of the level of abstraction in question) for visibility.

2.2 Relationships among abstract prototypes

Figure 3 shows the complex relationships that exist among the abstract software implementations within and beyond the individual levels of abstraction. Direct or indirect connections among abstract prototypes exist, e.g., among abstract software implementations at higher level of abstraction and those in the subsequent levels. It is therefore possible to trace the related abstract software implementations. This can be achieved as follows: Given, say, a theory t_i; $t_i \in T$, where; T is a set of theories for a given functionality, sets of possible methods $M' = \{m_1, m_2, ..., m_x\}$, algorithms $A' = \{a_1, a_2, ..., a_y\}$, and pilot implementations $I' = \{i_1, i_2, ..., i_z\}$ that satisfy specific requirements S_r can be traced provided that the needs (or few key requirements of the anticipated functionality) are known.

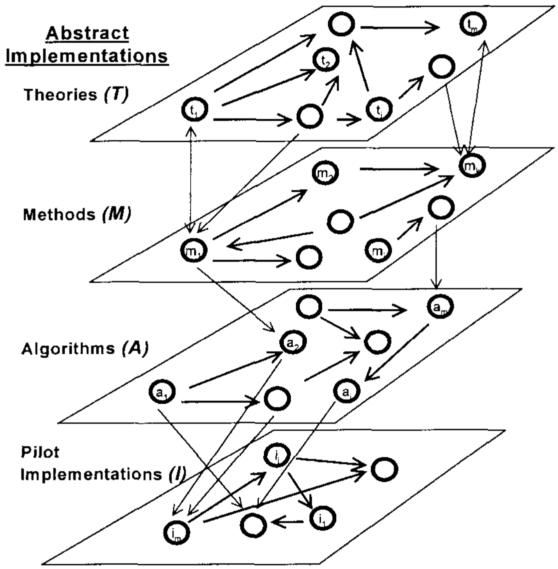


Figure 3. A network of abstract design support software implementations

2.3 Roles of Knowledge in Abstract Prototyping

There are several definitions of the term 'knowledge'. In the context of the abstract prototyping technique, we regard knowledge as clear understanding of abstract software implementations. The notion of the knowledge representation and processing aims at making the persons involved in abstract prototyping to become well informed about the abstract software implementations. Clear understanding of the abstract implementations is quite important in reaching proper judgements during abstract prototyping. Knowledge can be represented and made available in different ways and can serve the following purposes: It can be a fact, a descriptive entity to help humans' judgement, and/or as a test object.

- *Fact*: Facts represent knowledge, which can be discovered by referring to the facts about abstract software implementations, which may be in different forms of representation, e.g. in the form of text, video clip, illustrative diagram, audio description, etc.
- *An informative description to help human judgement*: The knowledge can be an informative description of the abstract prototypes, which provide comprehensive acquaintance that help to improve the understanding. This can be in form of text description, audio presentation, video presentation etc.

- *Test object:* Knowledge can also be derived from an implemented object (e.g. a running pilot implementation). Some sort of hands on tests can be carried out to determine e.g. the usability of the underlying method.

3 Methodology

The knowledge representation and processing scheme provide means for handling chunks of knowledge required in abstract prototyping. These include knowledge about (i) the current abstract software implementations, and (ii) abstract software developed in previous projects.

3.1 Representation and Modelling of Knowledge

Methods for representation and means for modelling of chunks of knowledge about abstract software for reuse are proposed. At the theories level of abstraction, abstract software implementations are represented either in text form, as simplified models or as mathematical expressions, while at the methods level, diagrams, procedural text, video clips or audio descriptions are used. At the algorithms level, natural languages, symbols, mathematical notations, diagrams (i.e. structure charts and data flow diagrams) and pseudo codes are used [10] while at the pilot implementation level, an abstract software implementations can appear as an executable program or a video clip of a running program. The full taxonomy and descriptions of the methods of representation of abstract software is available in [4]. The purpose of availing the abstract software implementations in different forms of representation is to help subjects to understand them better for the sake of evaluations. The proposed modeling methods in the abstract prototyping systems include the use of the text editors, equation editors, 2D modelers, and 3D modelers. Representation templates are prepared in advance such that once an abstract software implementation is thought of, it is modeled to match one of the templates. Only specific information that describes the implementation in question needs to be added. This provides ways for modelling the implementations quickly and in a standardized way.

3.2 Processing of Knowledge

Two methodologies, namely, (i) for tracing connections among the abstract implementations, and (ii) for finding abstract software implementations for known *needs* are proposed. A methodology for tracing connections among abstract implementations allows for the related implementations to be traced in all levels of abstraction. This can be achieved as follows. Given, say, a set of theories for a given functionality, the user can browse the corresponding methods, algorithms, and pilot implementations, which satisfy the project needs (or key specific requirements). The usefulness of this search is that the developers can explore the implications of choices that they are making. The method for finding abstract software for known needs works as follows. Knowing the *needs* and the *key specific requirements*, similar abstract software implementations, along with the evaluations and analyses carried out in the past can be traced. The advantage here is that should there be a similar abstract software implementation, the development process can be reduced to only just fine-tuning it, to incorporate very specific missing features.

3.3 WWW Abstract Prototyping

The stakeholders of potentially widely used applications, including e.g. for supporting the designers are distributed in geographic separated locations. Internet is an effective method for incorporating such geographically distributed stakeholders in abstract prototyping. In the Internet set up, the remote subjects can access the abstract prototyping system's information gathering saver and deliver their opinions. Similarly, they can browse and visualise the relevant knowledge about abstract software implementations published on the web via the Internet. This provides an effective means for collecting information from a wide spread spectrum of potential stakeholders.

4 Design and Implementation of the KRP Utility

Figure 4 shows the architecture of the knowledge representation and processing (KRP) utility. There are three main functions, namely, (i) the knowledge update, (ii) the multi dimensional knowledge search (MDKS), and (iii) links to search engines and other applications. Several types of information can accompany the representations of the abstract software. Such information items include the name, the functionality, and the level of abstraction. Other important information items such as the author's name and date can also be incorporated. The knowledge update functionality provides tools for updating and modification of knowledge.

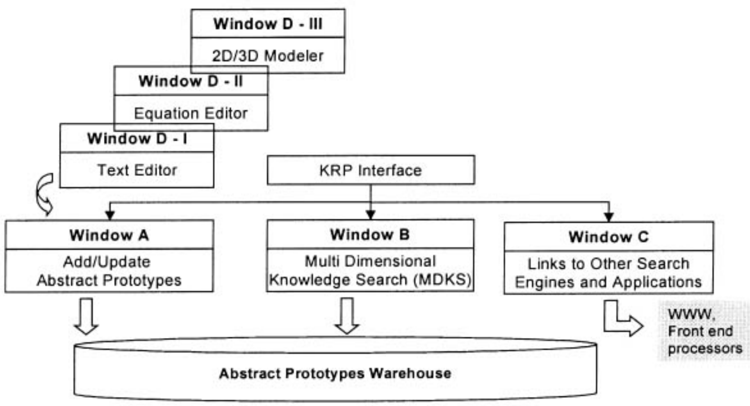


Figure 4. The data structure of the knowledge representation and processing utility

The strategy we used to demonstrate the KRP concept was to make use of the available elements of the widely used commercial software products as much as possible. The database has been implemented in Access, a relational database management system (RDBMS). The real value here is that the capabilities and features in Access such as entering, editing, and validating data in tables; using queries; and Internet-related features for creating HTML documents are available to support knowledge handling in a seamless user-friendly environment. The existing software tools, namely the (i) equation editor, (ii) drawings editor, and (iii) word processor are summoned to serve as functionality for modelling and presentation of abstract software implementations.

The MDKS tool offers means for searching abstract software implementations based on one or more constraints specified by the user. The current MDKS prototype utility provides the developers with means for finding (i) all abstract implementations for a functionality, and (iii) for browsing the abstract prototypes in the database (Figure 5). The user is required to specify the search dimensions (i.e. combinations of search criteria) for the intended exploration.

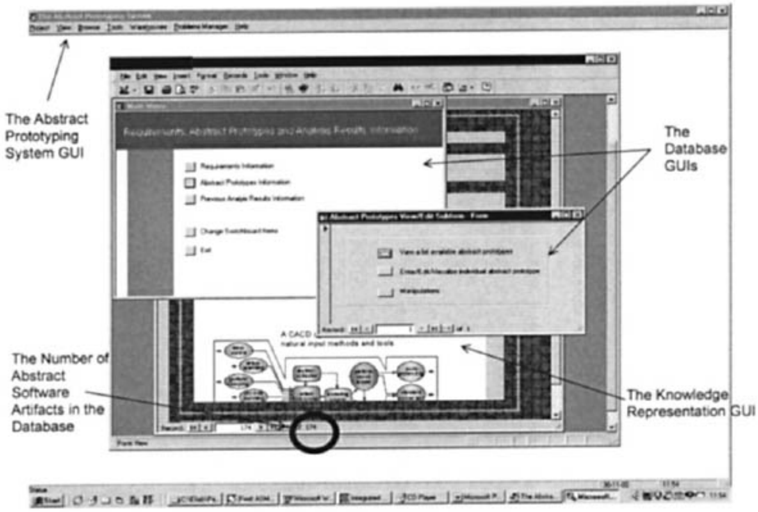


Figure 5. A printout of the KRP GUI

5 Case Study

The abstract prototypes of the elements of the CACD system that is being developed in our research group [11] are being tested in the framework of the abstract prototyping technique. 174 abstract software implementations (mainly foundational theories and underlying methods for the proposed conceptual design system) have been modelled and stored in the abstract prototyping systems' database (Figure 5). Apart from the storage role, the KRP utility offers the developers with means for modelling of the implementations and for publishing or searching the implementations on the web. This allows various stakeholders in geographic separated locations to access knowledge about the implementations. Expert judgements of various case applications show that this approach works quite well and is very effective. Subjects in remote locations can read, watch or listen to the facts and illustrations via the web. This makes them have good understanding of the abstract prototypes and increases the chance of achieving accurate assessment of quality of the design support software tools. The MDKS utility is being used in tracing the relationships among various abstract software implementations in different levels of abstraction. This is quite useful, especially during synthesis, i.e., when, for instance, the consequences of adopting an alternative are explored.

6 Discussion and Conclusions

The KRP utility has been developed to provide the subjects and the developers of design support software tools with means for handling knowledge about abstract prototypes during pre-implementation testing. This utility is being used in the process of development of a CACD system. Case applications indicate that it provides effective means for presentation, browsing and visualisation of abstract software implementations, and for exploring the implications of choices made by the developers. The use of the KRP scheme in the framework of the abstract prototyping technique makes the subjects to become more knowledgeable and help them make accurate judgements on quality of the implementations. Knowledge about the implementations is presented either as facts, informative descriptions or as executable implementations. Knowledge can also be published across networks and made accessible even from remote geographical locations. This provides the subjects, regardless of their geographical locations, with quick and effective way of accessing knowledge about the abstract implementations of design support software tools.

References

[1] Paulk, M. C., Curtis, B., Chrissis, M., B. and Weber, C. V.,"Capability Maturity Model, Version 1.1," IEEE Software, Vol. 10, No. 4, July 1993, pp. 18-27

[2] Whiteside, J., Bennett, J., and Holtzblatt, K., 1988, "Usability Engineering: Our experience and evolution, In Handbook of Human Computer Interaction (M. Helander ed), Elsevier Science Publishers.

[3] Opiyo, E. Z., Horváth, I., and Vergeest, J. S. M., 1999, "Abstract Prototyping of Design Support Tools: Methodology and Preliminary Results", Proceedings of the 1999 ASME DETC, 25th DAC Conference; September 12-15, 1999. Las Vegas, Nevada; Paper No. DETC/DAC-8551; ISBN 079181967-1.

[4] Opiyo, E. Z., Horváth, and Vergeest, J. S. M., "Pre-implementation Testing of Human-Computer Interaction Methods for Shape Conceptualisation", Proceedings of the 12th International Conference on Engineering Design, August 24-26, 1999, Munich, Germany, Vol. 3, pp. 1853-1856.

[5] Roseman, M., and Greenberg, S., "TeamRooms - network places for collaboration", In Proceedings of CSCW'96, pp. 325-333, 1996.

[6] Lotus Notes Homepage, http://www.lotus.com/home.nsf/welcome/notes, 2001

[7] NetMeeting Homepage, http://www.microsoft.com/windows/netmeeting/; 2001

[8] TeamWave Homepage, http://www.teamwave.com/; 2001.

[9] Tomek, I.., and Giles, R., "A Virtual Environment to Support Software Development Teams", Concurrent Engineering: Research and Applications, Vol. 7 No. 2, 1999, pp. 139-145.

[10] Jones, W. G. "Software Engineering", John Wiley and Sons, 1990.

[11] The Integrated Concept Advancement (ICA) Homepage, http://www.io.tudelft.nl/research/ica/, 2001.

First authors' name: E. Z. Opiyo
Institution/University: Delft University of Technology
Faculty/Department: Design, Engineering and Production/Design Engineering
Address: Jaffalaan 9, BX 2628, Delft
Country: The Netherlands
Phone: +31 15 278 6935; Fax: +31 15 278 1879; E-mail: e.opiyo@io.tudelft.nl

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

ENHANCED SYSTEMATICS FOR FUNCTIONAL PRODUCT STRUCTURING

U Pulm and U Lindemann

Keywords: Functional modelling, product architecture, product data management, product modelling, systems modelling.

1 Introduction

Many systematics for functional analysis and structuring exist in engineering design methodology and product development, but they do not seem to be very popular, especially in industrial practice. On the one hand, there is a lack of understanding their importance as their benefit is not evident. On the other hand, there are too many different models and it is not clear yet which one serves which problem best or is the "right one" at all. We want to take a critical look on this topic in order to achieve a better understanding and emphasize a more flexible use of functional structures in a wider context. For that we take, merge and enhance basic elements of existing approaches to constitute a comprehensive systematics integrating different aspects. We come to a first basis for a general abstract product model or a consistent model throughout the development process, that contains and connects functions, component parts, and requirements, but no detailed features. Though the approach is relevant for all types of products and functions, we focussed predominantly on mechanical products, which fulfil complex functionality, and strongly connect user and technical functions. Our case studies were an mechanical automotive seat and a bicycle, processed in two directions: The automotive seat had to be methodically optimised for an easy entry function in two-door cars, so that a functional structure had to be built up. The bicycle was chosen as an example, in which the main task was to evaluate PDM systems regarding their use in early phases of the product development process.

2 The functional structuring

2.1 Integration in the development process

In theory, the functional structuring is placed between the analysis and the synthesis of a system, since it is part of the analysis ("functional analysis"), and also an abstract basis from which specific solutions can be generated [3]. But practically, if done at all, the functional structuring often seems to be a self-contained part of the developing process, mostly neglected by finishing the analysis and superseded by the product structure in the later synthesis, breaking the continuity in the process. Next to the hardly spread use of methods in general, one point may be that there is no standard systematics and the real significance is not evident. Before considering this lack of integration, there is a discussion of existing systematics and related problems.

2.2 Existing approaches

Existing approaches for functional structuring can be classified into flow-oriented, hierarchic and networked structures. They can be closed or open systematics, which means that they either offer a limited set of functions or just a standard way to describe them. Many approaches differ only in the amount of the offered functions [9][10]. While flow-oriented structures represent the operating of the system, hierarchic structures seem to focus on user functions [1]. And while flow-oriented structures are based on a tangible turnover, i. e. materials, energy, information, etc., networked structures model logical connections. Of course, the borders between these classes are blurred. Next to this, two special approaches shall be pointed out. The Unified Modelling Language, UML, as a general modelling tool, does not regard functions especially or only on a very high level, i. e. the customer view "use cases". But the UML, especially with class diagrams, seems to be useful to describe and structure an artefact on the functional level, for which it has to be adapted [11]. TRIZ, a method based on contradictions between useful and harmful functions, is a special purpose method to find new solutions on the existing structure. But it does not seem to be optimised for mere mechanical products and it is not able to model a product throughout the whole development process [5].

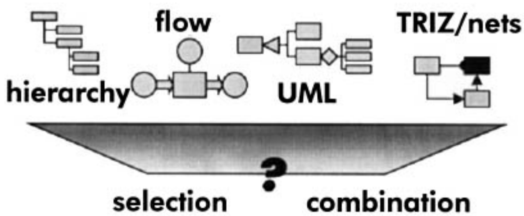


Figure 1. Choice between or combination of functional structures?

One problem for the use of functional structures is the choice and recombination of these and other methods and tools (fig. 1.). Other tools are e. g. SADT or IDEF0. Since there are as many methods and theories for functional structuring as there are authors engaged with this topic, every one cannot be discussed here. Free in the way to build up a functional structure for a system, we start from some general problems which are explained below, take elements from different methods, and come to a more general view of functions and product structures.

2.3 Abstraction of components

Functions are often understood as the solution-neutral abstraction of a component part or assembly (in the following subsumed to components). This delimitation between components and functions seems to be one major problem. It is often nearly impossible to build up a functional structure neutral to a general solution or special parts (which are in fact given since one has an idea of the product), without becoming trivial. It has been shown that no genuine, abstract function structure exists [6]. Neutrality is only given on a high level of a functional structure, and even there it is hard to say if the product or the function is on top. In fact, a hierarchical functional structure goes from solution-neutrality on the top level to a more and more specific solution principle when the structure is detailed. To show both neutral and specific functions in one model allows to see abstract purposes and behaviours, and change specific solutions together. In the case of an automobile, the function may be "transport persons". This function as an abstraction leads to other possible solutions such as planes, trains, bicycles etc. In this case, the customer does not buy the function, he buys the concrete product, and this is what has to be developed. The formulation of the function is inexpressive or trivial. Otherwise, a function is related to a specific subsystem of the product, and the function may be left out in another overall context by a possible change of the system's border. Even the function "transport persons" can be too specific to a solution, if an overall context is regarded ("enable communication", "have fun" etc.). Components and functions are connected in a hierarchy, the abstraction. Though it is not possible to build up an one-to-one relation between components and functions since a function may need more components or a component may serve more functions, there is a particular connection between functions and components. It is often difficult to find the right level, which depends on the task, and a risk of becoming trivial or fixed to a solution. A strict delimitation between functions and components neglects many aspects of a product's structure and seems to be inexpedient. A combined analysis of functions and components is therefore more effective [12].

2.4 The kind of structure

A major problem is the choice between hierarchic, flow-oriented and networked structures. Flow-oriented structures are in fact inadequate, if the turnover of the product (e. g. forces) is hardly tangible. In case of the seat this may be the adjustment of the seat height. Hierarchies offer a clear structuring of the product and an explicit abstraction/ascertainment or detailing/summarization. But hierarchic structures neglect cross connections, which are often more important than the hierarchy itself. Networked structures, which are sometimes hard to separate from flow-oriented structures, can represent different kinds of relations, but become quickly impenetrable, particularly if abstractions are included, which are difficult to model. The structure of a system or a product depends on the viewpoint of the problem. This concerns also the differentiation between a functional structure, a part structure, etc. We would like to combine these structures to a general view of the system, from which special views can be derived.

2.5 Related approaches

Related approaches offer a possibility to integrate components, functions, requirements, and some other aspects. These are computer-based tools, which allow modelling the abstract product in semantic networks, giving a standard in representing functions [2][13]. But the method to build up the functional structure is still missing. Other related approaches integrate different functional structures in order to achieve a method for mechatronical, i. e. interdisciplinary systems [7]. In contrast we would like to concentrate in the first step on mechanical complex products, and enhance this on any system in further steps.

3 Objective target

We come to the perception, that functional structures in a wider meaning or the abstract representing of a system have three different aims:

1. the abstraction of a system in order to synthesize new solutions (in this case the method, which helps to find a beginning)
2. the analysis of a system in order to understand it and to have a basis for communication (i. e. to make it understandable for other persons or to explain it)

3. a data basis for product data management throughout the whole development process (where also components have to be regarded).

We like to merge existing approaches in order to fulfil these three requisitions. Therefore we want to enhance functional structures to a more general approach, i. e. in fact a leveraging from functional structuring to abstract product structuring. We do not want to replace existing methods in general, but to combine and enhance them to a more flexible tool and a comprehensive basis, from which special structures can be extracted, i. e. a consist model for the whole development process.

4 Method of resolution

The aim is to connect the aspects hierarchy, flow, and network as well as functions, requirements, and components, and to abandon the obstacles concerning strict differentiations and delimitations. Therefore we would like to propose elements, a possible procedure, and general principles, which together build a systematics for a flexible use of functional product structures. In contrast to existing approaches concerning computer-based frameworks, in which the mentioned aspects can be integrated, we focus on the methodical background to build up such an abstract product structure, either in a computer-based tool or just on paper. Our approach is to collect basic elements from different methods, i. e. flow-oriented structures, hierarchies, UML and TRIZ as well as product structures from PDM-systems, and connect them to a comprehensive systematics. This flexible use and situative adaptation of methods is a fundamental aspect of a meaningful methodology and its understanding [15].

4.1 Elements of the abstract product structure

The elements of the proposed structure are essential "parts" of other product or functional structures and can be divided and classified as a first step as shown in fig. 2.

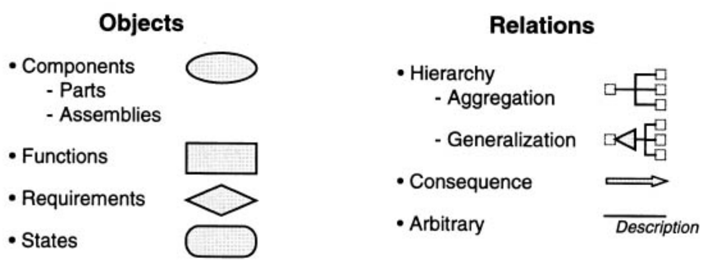


Figure 2. Elements of the abstract product structure

The major objects are components (parts or assemblies). Next to this, there are functions. They can also be a harmful effect of a component, which in complex mechanical systems arise especially by contradictions concerning different states of the system, which are further objects as well as requirements. More important seem to be the relations. The major structure is a hierarchy which can be built up by aggregation ("part of"-relation, AND-relation) or by generalization ("is a"-relation, OR-relation). There are different consequence relations such as flows of forces, reinforcements, necessities in building up the system, or restraints. At last relations can be build up arbitrarily and specially described. Of course, the transitions between these classes are blurred. This repertory is only a first step and an example for combining different basic elements to a comprehensive abstract product structure. It can be enhanced if necessary.

4.2 Proceeding

The first step in building up the structure is to distinguish between technical and other (economic etc.) requirements. Economic requirements accompany the development process on a different level; technical ones can be connected to components and functions. The next step is to build up different hierarchic structures for functions, components, and requirements (fig. 3.a). The functional structure goes from user functions to more technical functions, whereby a parallelism to the product structure is to consider. States are integrated in the functional structure, where a distinction of cases is necessary. Harmful functions or contradictions can be integrated and handled similar to TRIZ, but it is recommended to start with the positive functions. By cross-connecting elements within a structure, these structures separately become structures of different networked layers (fig. 3.b). Important relations can also be tied between different layers. In appropriate areas, flows can be included. Finally, the structures are connected on the highest possible levels of detail, where only one part is connected to only one function (or requirement), no matter if each subordinate element is also included (fig. 3.c). Further or detailed connections result from the hierarchical following structure. This is a compromise to the fact that functions and parts are not equal one another. In the next step, they should be combined and concentrated to one structure (fig. 3.d). Functions and parts can be deliberately alternated in the structure. This allows a differentiation, when an abstract function is required or when there is, for any reason, a default part. If it is not clear whether the product or the function is on the top, either both elements are integrated or there has to be a prioritisation.

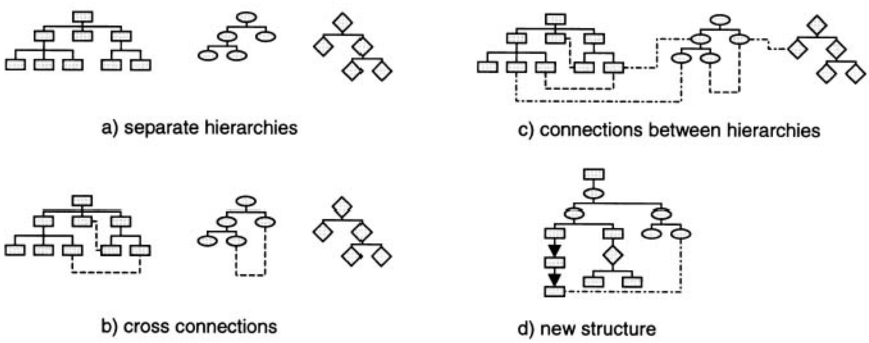


Figure 3. Proceeding in structuring

This procedure was the method used to develop our approach and need not be followed in each case. If a structure is built up on paper in order to manually analyse the system, it is possible to build up the finally structure directly. If the structure is documented in a computer-based system (especially PDM) and serves for the whole development process, this procedure is recommended, and the system can merge the structures automatically to one.

4.3 General principles and result

Next to the method of functional structuring, there are also different definitions of function [4][14], e. g. the behaviour of an artefact or the relation between input and output of a system.

We propose that a function has to be simply understood as the purpose of a system or component. It is recommended to describe a function by a noun and a verb, which enables the flexibility of language, without any narrowing classification. The advantage is that each part can be described by a function, and need not have a clear input or output. A harmful effect may be on the same level as a function but is not a function itself. A main principle is the formalism of the approach. The formalism has to be followed in order to find a beginning for the functional analysis (this understanding as a starting-point is relevant for each method). But as soon as this formalism does not seem to be suitable for the present problem, it should be rearranged. There is a dominant hierarchic structure, which connects different networked layers of the same degree of abstraction. In this hierarchic structure it is (often) possible to find the right degree of abstraction, as there are only few levels. The hierarchic structure can be realized by a usual tree or by detailed frames or by a combination of both. It is also possible to build up two reverse hierarchies which was in fact useful in the case of the automotive seat with its normal functions and the superposed easy entry system. Here, one structure was build up in form of a tree, and the other one with frames, as shown in the partial results of the automotive seat in fig. 4.

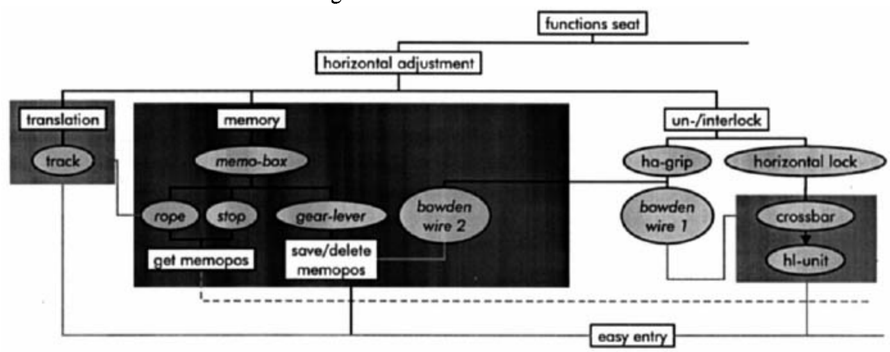


Figure 4. Partial functional structure of the seat

5 Understanding the meaning of functional structures

It is important to understand that a functional structure is neither a formalism nor an automatism to find new solutions. It is just a model of the system and a medium for abstract dealing with the system. Functional analysis is system analysis, which means that the strict separation between functions and components is no constraint. The formalism and the abstraction help and guide to concentrate on understanding the system, but this formalism can be broken, if other ways seem to be more expedient. So components can be included into the functional reasoning, especially if some of the parts are not to change. It is also often only a question of wording, if the component or the function is named. It is more important to think about the purpose or the function. This leads to the major problem in functional structuring or abstract product structuring in general, similar to the neutrality problem: Separate components of the system are described by their functions and structured or set in relation respectively, but the structure itself is hardly part of the consideration and later variation. One solution is to build up different functional structures. But our hypothesis is that no functional structuring systematics can represent each problem (and solution); so a flexible use is recommended. Such a functional structure can not be an automatism necessarily leading to new solutions, at least not to all solutions. Even TRIZ and all other functional structures only stimulate thinking about the system in a special way. The solution is the break between analysis and synthesis, which so gets a special meaning. It allows withdrawing from the existing system after the general problem and the starting-point are well known. So the system with its structure can be changed. The process of building up the functional structure is more important than the structure as a result (process-orientation instead of result-orientation). Some ways to work with functions and its structure itself have been shown [8]. It is better to build up a "wrong" structure by oneself than to get the "right" one presented. The visualisation of this structure affects this. Thus our approach shall not abandon this break between analysis and synthesis, but it shall enhance the functional analysis for a partial use in later synthesis. To achieve this, separate areas of the structure can be easily changed in synthesis and more detailed analysis structures can be drawn out from the basis structure described above.

6 PDM systems

A related work was to evaluate and enhance PDM systems for their use in early phases of the development process. PDM systems seem to aim at late stages by offering a relatively static and component oriented product structure. Requirements and functions were not presentable. Furthermore, the representation is mere hierarchic without any cross connections. These connections are stored in the data basis, but can only be shown by redundancy. We enhanced a PDM system with functions and requirements by customizing. In this system, the above mentioned proceeding can be used, whereby the last step, the integration in one structure, is automatically realized. Still the representation allows only a mere hierarchical structure, but the advantage is the advisability, e. g. one can read from the structure, which experiments (understood as a requirement) belong to a part and in the following which other parts belong to one of these experiments. The disadvantage is the still missing general overview of the system and the lacking flexibility in structuring.

7 Outlook

This structure described here is only the result of the above mentioned case studies. There are empirical studies necessary how far the theses are generally accepted. In fact, it is a dialectic alternation between particular and general approaches. Thus, an experimental comparison between different functional structures and this approach is aspired. And it is only a starting-point for a wider abstract product model for whole product spectrums including alternatives, degrees of freedom in customer choices etc. This includes enhancing the consideration to mechatronical or any system. It is also to describe how special structures, e. g. TRIZ, can be drawn out from this structure and how other methods can support this abstract structure, e. g. matrixes between functions and parts. Further research leads to the direction of building a tool for planning and modelling the structure itself, i. e. the structure in front of its elements. How far PDM systems can make this possible, and how far such a structure serves as a consistent basis for the whole development process, still has to be clarified.

8 Summary and conclusion

In this approach we took a critical look on functional analysis and enhanced abstract product structuring by the strategy of rearranging elementary methods. Our emphasis is on the flexible use of design methodology and a understanding of it neither as a formalism nor as an automatism but a tool to engage with a system. Analysis and synthesis shall not be separated totally. We think that the described approach or structural basis, respectively, serves for both analysis and synthesis, while special purpose structures can be drawn out. We do not intend to present the perfect systematics, and we do not want to abandon other functional analysis systematics. We would like to show a new possibility of abstract product structuring which has to prove its justification. The advantage is the compromise of handling complexity and having a comprehensive overview and understanding of the system, as well as the better understanding of functional structures in general. It is a basis for a consistent abstract product model throughout the whole development process, which serves for a real comprehensive and flexible kind of consideration of technical systems, and away from formal systematics.

References

[1] Akiyama, K.: Funktionenanalyse. Landsberg/Lech: Moderne Industrie 1994.

[2] Ambrosy, S.: Methoden und Werkzeuge für die integrierte Produktentwicklung. Aachen: Shaker 1997. Zugl. Muenchen: TU, Diss. 1996.

[3] Ehrlenspiel, K.: Integrierte Produktentwicklung. Muenchen: Hanser 1995.

[4] Hansen, F.: Konstruktionswissenschaft. Muenchen: Hanser 1974.

[5] Herb, R.; Herb, T.; Kohnhauser, V.: TRIZ. Landsberg/Lech: Moderne Industrie 2000.

[6] Hubka, V.: Theorie technischer Systeme. Berlin: Springer 1984.

[7] Kallmeyer, F.: Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme. Paderborn: HNI, Diss. 1998.

[8] Kavakli, M.; Suwa, M.; Gero, J., Purcell, T.: Sketching interpretation in novice and export designers. In: Gero, J.; Tversky, B. (eds.): Visual and spatial reasoning in design. Sydney: Key Centre of Design Computing and Cognition 1999, p. 209-219.

[9] Koller, R.: Konstruktionslehre für den Maschinenbau. Berlin: Springer 1994.

[10] Pahl, G.; Beitz, W.: Konstruktionslehre. Berlin: Springer 1997.

[11] Rumbaugh, J.; Jacobson, I.; Booch, G.: The unified modelling language reference manual. Reading: Addison Wesley Longman 1999.

[12] Steinmeier, E.: Realisierung eines systemtechnischen Produktmodells. Einsatz in der PKW-Entwicklung. Aachen: Shaker 1999. Zugl. Muenchen: TU, Diss. 1998.

[13] Szykman, S.; Racz, J. W.; Sriram, R. D.: The representation of function in computer-based design. In: Proceedings of the 1999 ASME Design Engineering Technical Conferences. Las Vegas: ASME 1999. CD-ROM

[14] VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. Duesseldorf: VDI-Verlag 1993.

[15] Zanker, W.: Situative Anpassung und Neukombination von Entwicklungsmethoden. Aachen: Shaker 1999. Zugl. Muenchen: TU, Diss. 1999.

Udo Pulm

Department of Product Development, Technische Universitaet Muenchen, Boltzmannstr. 15, D-85748 Garching, Germany, Tel: +49 (0)89 289 15155, Fax: +49 (0)89 289 15144, E-mail: pulm@pe.mw.tum.de, www.pe.mw.tum.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

CREATING VIRTUAL PROTOTYPES – INTEGRATING DESIGN AND SIMULATION

S Finger, X Chan, R Lan, and B Chan

Keywords: Computer-aided design, design representations, functional modelling, product modelling, prototypes.

1 Introduction

Building on the concept of reflective practice [1], our goal is to create a design system in which the designer can interact with both the form and the behaviour of the design, thereby enabling the exploration of a larger space of design alternatives. This paper presents a technique for integrating geometric modelling, physically-based modelling, and bond graph modelling in a CAD environment. We present a method for deriving bond graphs, a formal representation of system behaviour, from physically-based simulation modelling. The ability to derive behaviour from geometry allows a designer to understand and manipulate the interaction between the form of a design and its behaviour. In this environment, designers can create geometry, add attributes and constraints, and then see and analyse the design's behaviour using physically-based modelling. The research focuses on the transformation between form and behaviour, with the help of simulation feedback in order to support a design process model of evolutionary interactive design activity.

The end goal of the design activity is often a system of interconnected parts in which each part contributes to the function and behaviour of the whole. The physical behaviours of the entities can be diverse, but the designer usually has a well-defined functional goal. A designer may want to change the geometry of the entity without changing the goal, or to change a geometric feature in order to change the system functionality. In an ideal design process, where functions and entities are described accurately without ambiguity, mapping from the function specification to entity description is sequential and straightforward. In the real design process, design is not a simple mapping process but a stepwise refinement process in which the designer seeks a solution that satisfies constraints [2]. This introduces the concept of evolutionary design, which implies the back-and-forth characteristics of real design.

To transform form into abstract behaviour, we use real-time, physically-based modelling to simulate the behaviour of the form. During the simulation process, a constraint model, which holds the simplified geometry and physical properties of the design, is solved by the constraint solver for the force, velocity, new position, transformation, rotation and other numerical behavioural values, under influences and constraints like gravity, friction, collision, non-penetration and degree-of-freedom. The resulting values for force, velocity, *etc.* can be used to construct a behaviour graph. Since the interactive simulation is real time, the behaviour graphs can be updated continuously.

Conversely, to transform abstract behaviour into form, prototypical entities having parametric geometry and properties that can achieve well-defined behaviours can be represented in the knowledge base [3]. Given a behaviour graph, the geometry of an object can be constructed by matching the elements of the behaviour graph with the behaviours of the prototypical entities, assuming that each prototypical entity achieves a well-defined behaviour corresponding to an abstract behaviour. To assemble the entities requires the propagation of the geometric parameters for design coupling. The region of propagation needs to be defined to ensure the convergence of the solution. In both cases, interactive simulation provides the link between the actual behaviour and the expected behaviour.

This paper focuses on the transformation from form to behaviour and the construction of the behaviour graph from the real-time simulation.

2 Design representation

In order to develop transformation methods for mechanical design the following four issues must be addressed:

- Representing design requirements and specifications: Design requirements and specifications include behavioural requirements and physical requirements. Behavioural requirements describe the desired behaviour of the overall system at an abstract level, for instance, in terms of frequency, power, input and output, etc. Physical requirements describe the physical appearance of desired product like volume, colour, weight, *etc.*

- Representing physical component characteristics: Physical components, like design requirements and specifications, have both behavioural and physical characteristics. A complete design can be defined as a configuration of a set of physical components. Each physical component is represented as an object that has a behavioural representation, a physical representation, and an explicitly represented interaction between the two.

- Modelling relationship between the behavioural and physical representation of components: Unlike other design activities, mechanical design aims to generate designs that are composed of highly integrated, tightly coupled components. The interactions among the components are essential to the function and economic execution of the design. In this sense, the representation of the behaviours of mechanical components must be linked to the representation of their physical characteristics.

- Developing mappings from design requirements to physical components: Formal grammars that can generate and parse valid strings in a language have proven useful in engineering design. They have been used to generate and transform geometric and functional representations of design [4], [5], [6], [7].

2.1 Behaviour representation

Representing the behaviour of components in a formal, concise and accurate way can greatly improve the computability of a design process. For example, Lai created a formal, English language-based system, called FDL, for the representation of the function and structure of mechanical designs [8]. IDDL uses a similar approach [9]. A different approach to representing system behaviour is the bond graph. Bond graphs were created by Paynter to provide a convenient and uniform representation for modelling the behaviour of dynamic physical systems [10]. They have been broadly used in physical systems including those within the mechanical, electrical, hydraulic, thermal, and biological domain. Bond graphs enable mechanical and other physical systems to be represented in a manner equivalent to electric circuit diagrams. For instance, a spring in a mechanical device acts like an electrical capacitor by storing and releasing energy.

Bond graphs represent the physical systems using idealized elements: ports (vertices) and bonds (edges). Bonds connect ports, and power flow through bonds and ports to be dissipated, stored, supplied and transformed. Elements are divided into three types: 1-port elements, 2-port elements and N-port elements. 1-port elements dissipate power, store energy, and supply power. They can be further divided into passive 1-port elements, like springs and dampers, and active 1-port elements, like force and velocity. 2-port elements transform power, for example gear pairs. N-port elements correspond to connections among the elements. Two types of N-port elements are defined: 0-junctions and 1-junctions, which correspond respectively to "same force" and "same velocity" connections.

Bond graphs enable a uniform classification of variables to describe the interconnection of elements inside a system and the input and output for each subsystem. Based on bond graphs, Welch and Dixon defined a behaviour graph for conceptual design of mechanical systems. Because bond graph only represents scalar values (the magnitudes of efforts and flows) and neglects the thermodynamic property, mass and momentum, they developed behaviour graphs to overcome these shortcomings [11]. Figure 1 shows a lever with its corresponding behaviour graph. Behaviour graphs capture spatial location and orientation information, which are important in mechanical design where form, behaviour, and function are intertwined. The generalized efforts and flows acting on these elements are represented not only by their magnitudes, but also by their vectors.

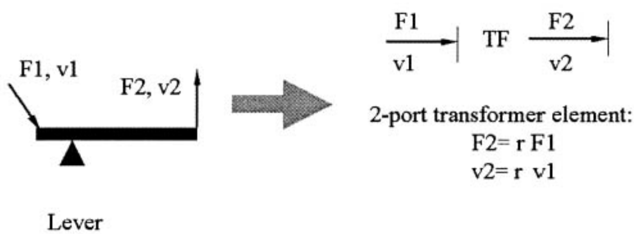


Figure 1. A lever represented in a behaviour graph

2.2 Geometric representation for conceptual design

During the conceptual and embodiment design phase, the specific geometry is not the main concern of a designer. However, to illustrate solutions and generate design alternatives, models of some kind need to be created. Work by Guan *et al.* overcomes this weakness by establishing geometric models using vague geometric information, because for most product design, geometry is an important consideration throughout the design process [12]. In Guan's work, the geometric properties of each object can be classified into its shape and size, as well as its orientation and location. The properties are usually described by geometric parameters. In the early design stages, these parameters are not and need not be clear to a designer. The work defined the vagueness of the geometric property into two types: incomplete and approximate.

The geometry of objects in conceptual design and the material attributes inherent in the design object are both aspects of form [13]. The geometric aspect is described using geometric parameters like length and width, and topological primitives like face, edge, and vertex. Material attributes are described using attributes such as density and Young's modulus. Conceptual objects have no default geometric or material attributes, but exist as symbols at a desired location or region.

2.3 Interactive simulation

Witkin *et al.* have proposed a mathematical formulation for constraints to support the use of physical simulation as an interactive medium for building and manipulating some classes of models [14]. His work especially concerns making objects' behaviour consistent with the constraint forces acting on them. To achieve the goal of merging the process of model creation with an ongoing simulation, the implementation allows the addition and deletion of constraints and the automatic forming and solving of new set of state equations. Following on this work, Baraff developed several new formulations of the previous work, to let users directly manipulate simulated objects as they moved around and to allow on-the-fly modifications to the simulation environment [15]. To simplify the problem, we only consider simulating objects that are composed of solid rigid bodies.

In Baraff's work, the simulation is viewed as the process of numerically solving the differential equation to evaluate the state of the system $Y(t)$ over time t:

$$\frac{d}{dt}Y(t) = f(Y(t),t) \tag{1}$$

The goal of the system is always to advance from a given state $Y(t_0)$ to a new state $Y(t_0 + \Delta t)$, where Δt is the simulation step size. When the system is simulated as the second-order world following Newtonian dynamics, Equation 1 can be written as:

$$\frac{d}{dt}Y(t) = \frac{d}{dt}\begin{pmatrix} x(t) \\ \theta(t) \\ mv(t) \\ I\omega(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ \omega(t) \\ F(t) \\ \tau(t) \end{pmatrix} \tag{2}$$

The system experiences inertia and objects tend to keep moving until brought to rest by forces in the simulation.

Continuously evaluating the right-hand side at particular time t at small time interval Δt results in the force F and torque τ acting on each object at any specified instant of time, and velocity $v(t)$ and angular velocity $\Delta(t)$ when simulating the dynamic of the system. The force F and the torque τ are the total forces, external and internal, acting on an object. External forces are forces like gravity, velocity-based damping, user-specified interactions, such as interactive forces applied by the user. Internal forces arise to satisfy the bilateral and contact constraints in the system, which include friction.

Interactive simulation must deal with two kinds of constraints: bilateral constraints and contact constraints. Bilateral constraints typically arise in representing idealized geometric connections like universal joints and point-to-point or point-to-surface constraints. Bilateral constraints are relatively static and are explicitly created or deleted by the designer. Contact constraints are transitory as objects come into contact with one another and then move apart.

The system uses collision detection algorithms to decide when to create a new contact constraints and when to eliminate them. To detect collisions and contacts requires information about the geometry of objects. A separate collision/contact detection component is applied to check each proposed state $Y(t_0)$ for possible intersection between objects. Given a computed state $Y(t_0 + \Delta t)$ the system uses the collision-detection algorithms to check for intersections. If no intersection is found, the simulator then moves the system to time $t_0 + 2\Delta t$. If an intersection is found, the simulator attempts to find the time t_0 between t_0 and $t_0 + \Delta t$ when the collision first occurred. The system assumes that at an initial state $Y(t_0)$, the system is free of intersections, if the state computed $Y(t_0 + \Delta t)$ is also free of intersections, the system has no collisions in the interval $[t_0, t_0 + \Delta t]$. The weakest feature of the simulation system is that fast moving objects, relative to the step size Δt, can tunnel through other objects without a collision. However, the assumption of discrete time greatly simplifies the collision detection problem making real-time interaction feasible.

3 Transforming form and behaviour

Bond graphs provide an explicit and abstract way to analyse a designed system. The flow and effort variables of velocity/angular velocity and force/torque are computed in the simulation. Since the constraint model corresponds to the solid model of design objects, transformation from form to behaviour can be performed.

When the user changes the form of designed objects through the design interface, the change needs to be reflected in the abstract behaviour diagram to keep the consistency of the state model. Once the form is modified, the design may not have the same behaviour. For example, in the rack and pinion illustrated in Figure 2, the pinion drives the rack. The mechanism transforms torque and angular velocity to velocity and force. If the designer changes the diameter of the pinion causing it to lose the coupling with the rack, even though the pinion still has outputs of angular velocity and torque, the behaviour of the rack no longer moves, since it does not have any input. Another situation is that the rack, moving in the horizontal direction, loses contact with the pinion. Although the pinion still has the same output, the rack again has a different behaviour.

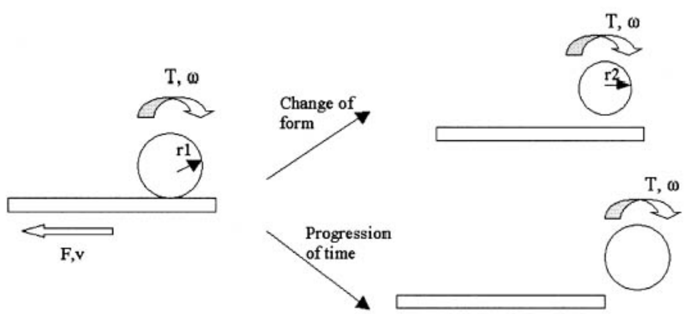


Figure 2. Two conditions that require a redefinition of behaviour

Our approach is to characterize behaviour through simulation feedback. It is relatively easy for a designer to observe the behaviour of objects. The interactive simulation enables the computation of behaviour variables like velocity, acting force, contact feature, *etc.* Given a simulation step Δt, the system can compute the value of those variables.

4 Implementation

The form and behaviour of a design is held in the design object's frame by two essential parts: the geometry of the design objects with physical properties like material and mass, and the behaviour model of the design objects. The geometry of the design objects is represented by a non-manifold modeller that allows the attachment of attributes. The behaviour model is stored using bond graphs, derived from the simulation within a CAD system. Specifically, we collect the geometry of objects and rebuild them in our simulation environment, from the model representation, create bond graphs, and simulate the system under the same influences and constraints. We can model only rigid body dynamics; that is, the dynamics of bodies do not change shape under force over time.

4.1 Simulation implementation

Because we use existing CAD packages for entering geometry, we are primarily concerned with the geometric representation within the simulation software.

In physically-based modelling, we use the formalism method called Differential Equation System Specifications (DESS) [16]. These systems have continuous states and continuous time. In rigid body simulation, the first task is to find the physical system's mathematical representation. The simulation process is shown in the Figure 3.

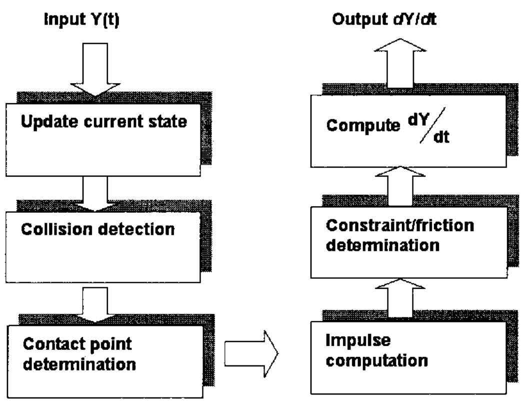


Figure 3. Simulation process

This simulation process enables us to find the system state from time t to $t+\Delta t$. First, we update the initial conditions and determine whether a collision occurs. If it does, we find the contact points and compute the impulse forces. Finally, we use numerical methods to compute the state equation, taking into account the constraint conditions.

The Coriolis library was created by Baraff [15]. The library is a constraint solving kit that, given a system of rigid bodies, provides general equality and inequality constraint formulation.

It can solve the constraints in real time, dynamically calculating topological information of rigid bodies such as body location, transformation matrix, rotation matrix, *etc.*, kinematics and dynamics properties like velocity and acceleration of rigid bodies, acting forces, *etc.* For the convenience and intuition of constraint construction, pin joints, rotation and transformation of object, springs and unisprings, nails and other common constraints are built in function calls. The constraint solving kit is an object-oriented library that has as its functions the methods of a hierarchy of classes. The library is able to solve motions of both 2D and 3D rigid body systems. However, the library does not support design activities, and the library does not have display functions. The library takes model from point and edge construction and has hooks to support standard OpenGL rendering.

4.2 Bond graph implementation

Within our framework, the behaviour model is represented by components (1 or 2 port), joints, and ports. The components are the rigid bodies in the simulation or force/torque sources; the joint is defined as an association between two components. Two components interact through ports, which are points where energy flows in or out of a component. Components have a single port. Joints have two ports, and represent a constraint between the two components that they connect.

In bond graphs, each component represents a certain behaviour. For example, we define two special components that represent two kinds of sources:

- Effort source *Se***:** output effort, e.g., force;
- Flow source *Sf*: output flow, e.g., velocity.

The two special components have a fixed effort direction (fixed causality). The Se-element represents effort-out and the Sf-element represents effort-in.

In physical systems, the energy flow between two components has only one direction. In bond graphs, we use the direction of effort to represent causality. Because of this property, we use directed graphs to represent the bond graphs of a system. Given a system schematic, we can derive the bond graph and the directed graph. With the directed graph data structure, we can store the components and joints in the nodes, traverse the graph, and obtain the system equations.

5 Conclusion

Existing CAD tools enable the designer to create and change the form of the design, but the computer cannot aid the designer in determining whether the design meets the behavioural requirements. The computer interaction with the designer for most CAD systems is limited by its lack of design knowledge. Research in artificial intelligence has addressed developing a computational understanding of the design process and developing representations for function and behaviour. This research explores a different solution using physically-based modelling that enables intelligent interactions between the computer and the designer by enabling the automatic capture and formal representation of the behaviour of a design-in-progress. This formal representation of design behaviour can be updated whenever designer changes the geometry, material properties, or constraints in the CAD model.

References

[1] Schön, D.A., The Reflective Practitioner, Harper-Collins, New York, 1983

[2] Tomiyama, T, "From General Design Theory to Knowledge-Intensive Engineering," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 8(4), 1994, pp.319-333

[3] Finger, S. and Rinderle, J.R., "A Transformational Approach to Mechanical Design Using a Bond Graph Grammar," Proceedings of the First ASME Design Theory and Methodology Conference, 1989, pp.107-116

[4] Stiny, G., Pictorial and Formal Aspects of Shape and Shape Grammars, Birkhauser, Basel, 1975

[5] Fitzhorn, P., "A Linguistic Formalism for Engineering Solid Modeling," Graph-Grammars and their Applications to Computer Science, Lecture Note Series, Springer-Verlag, New York, 1987, pp.202-215

[6] Heisserman, J., Generative Geometric Design and Boundary Solid Grammars, Department of Architecture, Carnegie Mellon University, Pittsburgh, PA, 1991

[7] Deng, Y-S., Feature-Based Design: Synthesizing Structure from Behavior, Ph.D. thesis, Mechanical Engineering Department, University of Pittsburgh, 1994

[8] Lai, K. and Wilson, W. R. D., "FDL: A Language for Function Description and Rationalization in Mechanical Design," Computers in Engineering, ASME, New York, 1987, pp.87-94

[9] Tomiyama, T., Kiriyama, T., Takeda, H., Xue, D. and Yoshikawa, H., "Metamodel: A Key to Intelligent CAD Systems," Research in Engineering Design, 1989, pp.19-34

[10] Paynter, H. M., Analysis and Design of Engineering Systems, MIT Press, Cambridge, MA, 1961

[11] Welch, R.V. and Dixon, J.R., "Representing Function, Behavior and Structure during Conceptual Design," DTM'92, ASME, 1992, pp.11-18.

[12] Guan, X., MacCallum, K.J. and Duffy, A.H.B, "Classification of Geometric Design Information and Manipulation for Vague Geometric Modeling," Knowledge Intensive CAD, Volume 2, Chapman & Hall, London, 1997, pp.224-236

[13] Chan, B., Supporting Conceptual Design: A Methodology and Infrastructure for CAD, Ph.D, Civil and Environmental Engineering, Carnegie Mellon University, 1998

[14] Witkin, A., Gleicher, M. and Welch, W., "Interactive Dynamics," Computer Graphics, 24,(2), 1990, pp.11-21

[15] Baraff, D., "Interactive Simulation of Solid Rigid Bodies," IEEE Computer Graphics and Applications, 15(3), 1995, pp.63-75

[16] Zeigler, B.P., Praehofer, H. and Kim, T.G., Theory of Modeling and Simulation, Second Edition, Academic Press, 2000

Susan Finger
Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA, Tel: +1 412 268-8828, Fax: +1 412 268-7213, E-mail: sfinger@cmu.edu, URL: http://www.cs.cmu.edu/~sfinger

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

FUNCTIONAL PRODUCT MODELLING – NEW METHODS FOR THE GENERATION OF PRODUCT FUNCTIONS

R Stetter, G Aßmann, and M Viertlböck

Keywords: Functional modeling, product modeling, systems modelling.

1 Introduction

Function structures are a proven means that assists designers to model their product on an abstract level, and through this, to develop a sensible product structure without restricting the search space to specific solutions. In literature, different forms of function structures (flow-oriented, hierarchical or combined) are proposed and elucidated. Furthermore, procedures are proposed which describe how a function structure can be derived from an existing product (function analysis). This paper gives an overview of existing methods and includes a comparison of the advantages and disadvantages of each. It was found that little guidance is provided regarding how a solution-neutral function structure can be developed. In order to address this inadequacy, four methods were developed by means of analysing a sample of products and their development processes.

These methods can be used in combination with conventional hierarchical function modeling techniques, for example, the Function Analysis System Technique (FAST) [14]. They support designers to develop an abstract function structure of their product in a solution-neutral manner. Moreover, these methods can be used to find and generate new product functions in order to support the innovation process in product development.

2 Background

In brief, a function can be characterised as a description of something a product should do. Pahl&Beitz [1] define a function as the general and intended correlation between the input and the output of a system with the objective to fulfil a task. Frequently, this rather narrow definition is widened, for example, in the field of value analysis. In value analysis, a function is understood to be any task that is to be fulfilled by means of either an object, which is existing or needs to be developed, or by means of a process or an activity [2]. In this paper, we follow this definition and see functions to be working principles, which present on the one hand, the technological functionality and on the other, emotions, appearance, prestige, cost and process properties. Each property of a product can then be presented by a function. Consequently, not only the product plays an important role when describing a function but also human beings. For this purpose, 'use functions' and 'recognition functions' can be distinguished.

The value of functions to design activity is that they can be the interface between marketing, design and product development. Functions serve as a means for 'translating' the abstract customer requirements into more concrete product development tasks. An important benefit of the use of function systems (i.e., structured depictions of a series of functions, often referred to as function structures) for an industrial company is the possibility to detect new, interesting, and innovative fields for product development.

Of course we have to think about the question, 'why do we need function modelling in mechanical engineering?'. If we have a look at various products, we will see that often the important functions of products are identical throughout time. The quality of function is improved continuously. A good example is the function of trucks which is shown in Figure 1. The basic function, 'transportation of goods' has not changed for a very long time whereas the quality of 'transportation of goods' has been improved dramatically i.e. speed, security, controlled climate conditions etc. (in this context it is also very interesting to notice that only very few functions are dying out.)

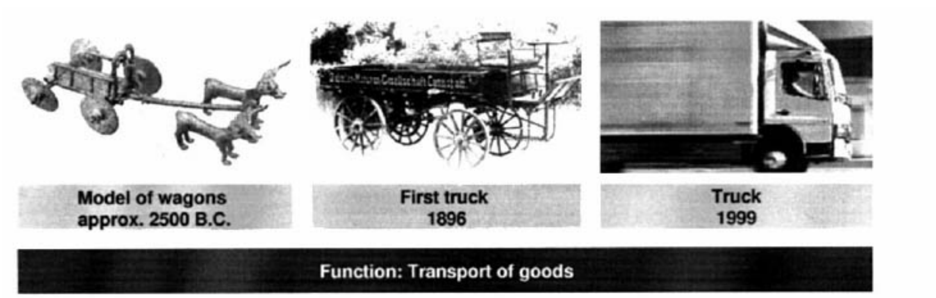


Figure 1 Development of the quality of functions of trucks throughout the centuries

The generation of new functions is often the basis and possibly the most important thing for the success of new products because they have to be innovative and must have new functions to be different when compared to competitors. An example for a completely new function was: 'keeping cars on the road in the case of going into a skid'. The invention of that function led to the installation of millions of miles of safety barriers all over the world (Figure 2).

Figure 2 Highway without safety barriers (1940)

Because of the tremendous importance of finding new functions, it is necessary that we support the developer with methods that support the generation of completely new product functions in a systematic way.

Extensive work was and is currently being carried out in this field of research. However, many research groups focus on the further development of a means for depicting functions, i.e., function structures, and on the use of function systems for the analysis and representation of existing products (see section 3). In the presented research work, we focus on the generation of functions. Essentially, we present methods to support the process of creating ideas for new products on an abstract level. This is based on the hypothesis that the occupation with concrete products may limit the imaginative capacity of the designers. By means of generating new function systems, we aim on the one hand, to create products which could not be thought of through conventional means and on the other, at providing a 'translation' for innovative ideas from marketing and design.

3 Function systems

In general, two kinds of function structures (i.e., depictions of a function system) can be distinguished. Flow oriented function structures focus on the flow of matter, energy and signal through a system. Closed flow oriented function structures aim at representing any function by means of a limited set of general valid functions. Well known closed flow oriented function structures were developed by Rodenacker [3], Krumhauer [4], Koller [5], Roth [6] and Ehrlenspiel [7]. On the contrary, open flow oriented function structures do not limit the set of functions. Pahl&Beitz [1] recommend the use of open flow oriented for vivid application. Recently, Otto&Wood [8] adapted this function structure for reverse engineering. Other types of open flow oriented function structures are proposed by Ropohl [9], Ullman [10], Grabowski et al. [11] and, in the field of artificial intelligence, Chandrasekaran [12].

The second general kind of function structures are hierarchical function structures. Essentially, hierarchical function structures show how a set of sub-functions enables a superordinate function. A closed hierarchical function structure is presented by Kirschmann et al. [13]. An open hierarchical function structure is used in value analysis [2, 14]. Additionally, an approach which combines hierarchical and flow oriented function structures was proposed by Szykman et al. [15].

In contrast to this variety of possibilities for analysing and depicting function systems, only two approaches are presented in research literature for the generation of function systems: the 'holistic method' and the 'bungled work method'. The 'holistic method' [9, 2] describes a procedure for the generation of a function system starting with the system environment and proceeding over the main function to the subsystems and their sub-functions and structural connections. The 'bungled work method' proposes to start with the sub-systems and to combine their inputs and outputs flows to a super system. This method mainly supports a designer in the development of a function structure for an existing system. Only the 'holistic method' is really appropriate for generating function systems for products that do not yet exist. However, in this method, little more is described than a general procedure. The four methods described in this paper accompany the 'holistic method' and give concrete recommendations of how a function system for a product that is not yet existing can be generated.

4 Methods for the generation of product functions

In the following section, four methods for generating new product functions will be explained through an example product. For this case study, and example, a refrigerator has been chosen for the sake of simplicity and clarity. Needless to say that these methods have been tested on a variety of products.

4.1 Function - product - function

This method is based on an iterative change between function and product – an alternation between abstract and concrete levels – and is structured hierarchically. After a function-product-tree for a given product has been generated, the product levels can be eliminated to get a function system (see Figure 3).

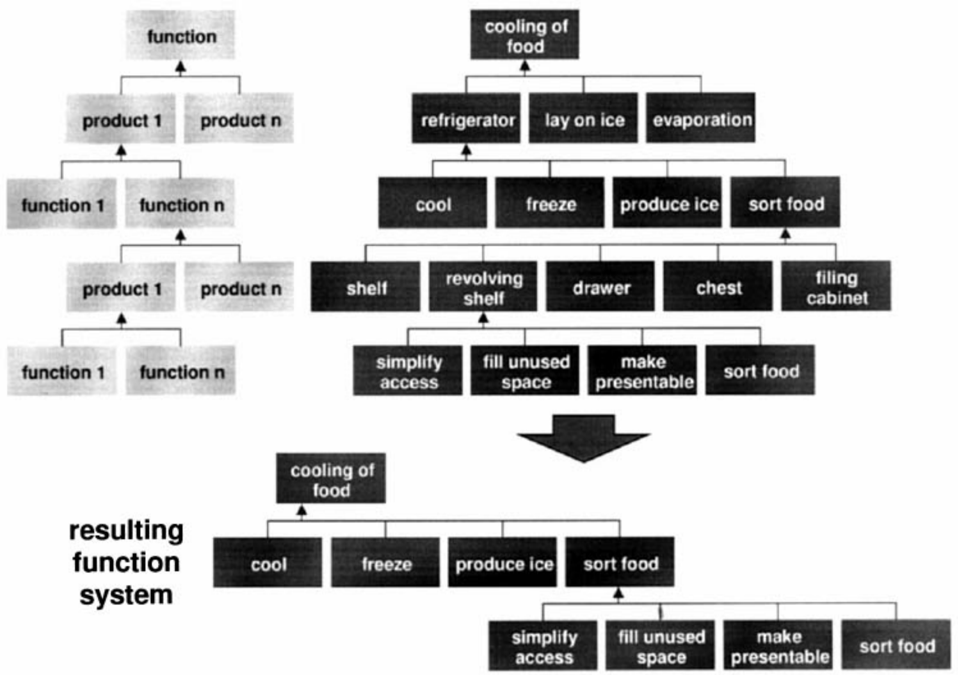


Figure 3 Function - product – function method: abstract explanation and example

The resulting function system represents the product on a higher level of abstraction as the function-product-tree. Additionally, the described product elements on the product levels can serve as first solutions for the function system. Further advantages are that:

- as the method is solution oriented, the support for the function generation is very good,
- the basic product functions can be found very quickly and
- existing functions can be found that lead to a better understanding of existing products or systems.

A disadvantage can be seen in the orientation towards known products or systems.

4.2 Function – counter function

This contradiction oriented method is based on disruptive functions. The method could be used by a moderator to ask designers for critical functions and disruptive functions. Through this, the view on the product and the functions, respectively, can be enlarged and passive functions can also be found. The structured sequence is shown in Figure 4. A new product function is generated at the end of this sequence.

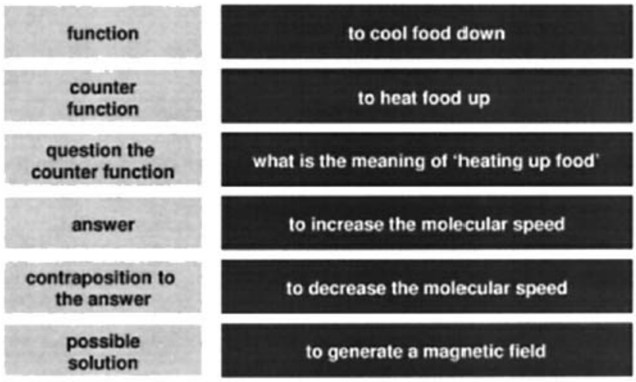


Figure 4 Structured sequence of the method function – counter function

4.3 Decomposition of functions

The systematic decomposition of functional elements is derived from the object oriented software modelling [16], a modelling technique used to build up and describe relationships between objects. The two basic relationships between objects that model their interfaces and relationships are the aggregation of objects, and the generalisation or specialisation of properties.

The aggregation of objects describes the 'is a part of'-relationships between objects. If this is adapted to functional modelling, a function can be detailed while splitting it into its sub-functions. In this application the main function 'cooling the food' can be detailed while asking 'What is a part of cooling food?'. Another variation of this method is to decompose even the single words of the sentence 'cooling food', e.g. 'What does food consist of?' and 'What is a part of cooling – or the cooling process?' (see Figure 5). The results are basic functions from which new product functions can easily be derived like 'storing food at different temperatures'.

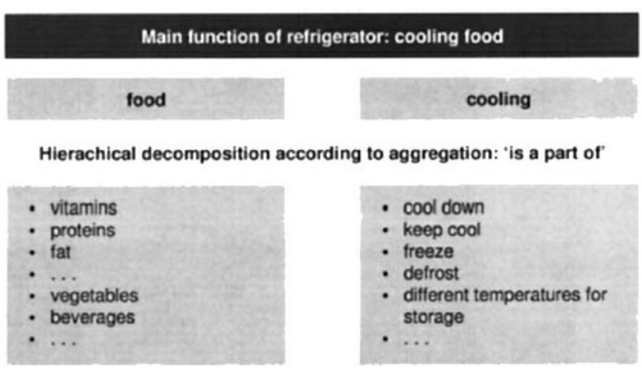


Figure 5 Hierarchical decomposition according to aggregation

While the aggregation of objects describes an 'is a part of'-relationship, generalisation or specialisation models inherited relations between objects and can be translated with 'is a part of'. For functional modelling, supporting questions could be either 'What is a part of cooling food?' or a more applicable case 'What is a part of food?' and 'What is a part of cooling?'. Figure 6 shows some examples for decomposing the main function 'cooling food' with this method. The generation of new product functions follows the same way as mentioned above.

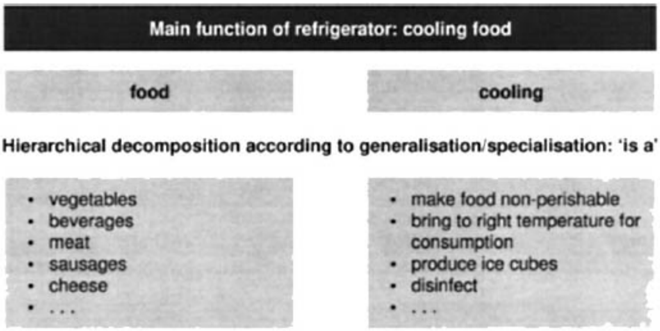


Figure 6 Hierarchical decomposition according to generalisation/specialisation

The decomposition of functions and the decomposition of the single verbs, nouns and adjectives of a function, respectively, leads to a complete description of functions that can be used either in team-sessions or alone. The application of this method is easy, however due to its tendency for accuracy, an immense number of sub-functions may result. Therefore an iterative evaluation and selection process must be considered.

4.4 Object in the system

The method 'object in the system' relies on the systematic approach of describing objects and their relationships. Due to the view on the main functions of the whole product system, further interactions of the product with different categories can be found. These categories should take environmental and social aspects into account as well as the technical components of the product. These categories are:

- product
- single product components
- people
- society and environment.

In Figure 7, the application of this method can be seen for the example mentioned above. From every category, questions to support the generation process of product functions can be derived. The general questions should be for every category 'Which influences does the product have on e.g. people?' or vice versa 'Which influences do the people have on the product?'. This question used with all of the four categories, is a guideline for an interdisciplinary brainstorming session.

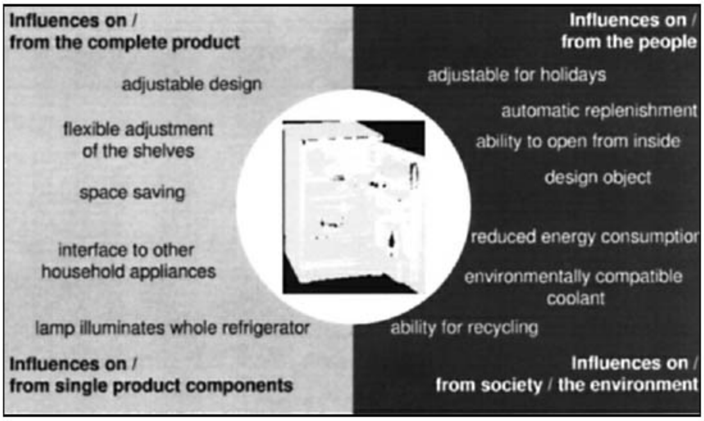


Figure 7 Example application of the method 'object in system'

Used in brainstorming sessions this method is dedicated to enhance the field of applications of a product. However, the focus on technology and product technology is not distinctive and the results are sometimes hard to reproduce.

5 Conclusion

The presented methods support the generation of functional product structures in the early phases of a product development process. The resulting structures reflect the problem to be solved instead of giving a certain solution that solves the problem. Therefore, they allow to develop a functional product structure without limiting the search space. Moreover, these methods can also be used to support the generation of new product features and are therefore an important part of the conceptual phase of the product innovation process.

These methods can be seen as a beginning or basis for the further development of innovative methods for generating product functions. The necessity and importance of these methods can be derived from the small steps that can currently be found between product generations and from the huge success a product may have if a bigger step is performed. For example, the new function 'carry in jacket pocket' of the Sony Walkman was the basis for a long-term market success.

As a guideline to support team-work, these methods are designed to give aid in moderated meetings. Additionally, the resulting functional product structures can be used as a means for communication, for example, between marketing, design and engineering. Nevertheless these methods have to be adjusted to the specific and individual processes in industry before application.

There can be no denying the fact that this set of methods is not yet complete. On the contrary, we would like to initiate a movement towards the creation of more methods that help designers to develop new product functions as a first but inevitable step to innovative products and market success.

References

[1] Pahl, G.; Beitz, W.: Konstruktionslehre. Methoden und Anwendung. Berlin: Springer, 1993.

[2] VDI: Wertanalyse. Idee - Methode - System. VDI-Verlag GmbH: Düsseldorf, 1991.

[3] Rodenacker: Methodisches Konstruieren. Berlin: Springer: 1976.

[4] Krumhauer, P.: Rechnerunterstützung für die Konzeptphase der Konstruktion. Diss. TU Berlin, 1974.

[5] Koller, R.: Konstruktionslehre für den Maschinenbau. Grundlagen zur Neu- und Weiterentwicklung technischer Produkte. Springer: Berlin, 1994.

[6] Roth, K.: Konstruieren mit Konstruktionskatalogen. Berlin: Springer, 1994.

[7] Ehrlenspiel, K.: Integrierte Produktentwicklung. München: Hanser, 1995.

[8] Otto, K.; Wood, K.: A Reverse Engineering and Redesign Methodology for Product Evolution. In: Proceedings of the 1996 ASME DETC in Irvine. New York: ASME International, 1996.

[9] Ropohl, G.: Systemtechnik - Grundlagen und Anwendung. München: Hanser, 1975.

[10] Ullman, D.: A New View on Function Modeling. In: Proceedings of the ICED'93. Zürich: Edition Heurista, 1993.

[11] Grabowski, H.; Rude, S.; Langlotz, G.: Die Funktionsmodellierung auf dem Weg zur Wissensverarbeitung in CAD-Systemen. In: Tagungsband CAD 1996. Berlin: Springer, 1996.

[12] Chandrasekaran, B.: Functional representation and causal processes. In: Yovits, C. (Ed.) Advances in Computers, Vol. 38, 1994.

[13] Kirschman, C., Fadel, G.M. and Jara-Almonte, C.: A Function Based decomposition in Mechanical Design. In: Proceedings of the 1996 ASME DETC. New York: ASME International, 1996.

[14] Akiyama, K. (1994) Funktionenanalyse. Der Schlüssel zu erfolgreichen Produkten und Dienstleistungen. Landsberg: Moderne Industrie.

[15] Szykman, S.; Racz, J.; Sriram, R.: The Representation of Function in Computer-Based Design. In: Proceedings of the 1999 ASME DETC. New York: ASME International, 1999.

[16] Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorensen, W.: Objektorientiertes Modellieren und Entwerfen. München: Hanser, 1993.

Dr.-Ing. Ralf Stetter, ralf.stetter@audi.de
Dr.-Ing. Gert Assmann, gert.assmann@skm.siemens.de
Dr.-Ing. Markus Viertlböck, markus.viertlboeck@bmw.de

Contact: Prof. Dr.-Ing. U. Lindemann, Lehrstuhl für Produktentwicklung, Technische Universität München, Boltzmannstraße 15, 85748 Garching, Germany.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

CREATING EQUATION HANDBOOKS TO MODEL DESIGN PERFORMANCE PARAMETERS

C R Bryant, M A Kurfman, R B Stone, and D A McAdams

Keywords: Design Representations, Functional Modeling, Systems Modeling, Engineering Analysis

1 Introduction

In an attempt to reduce product design time, engineers are currently exploring alternative ways to design and engineer their products. In the next century, tools that are able to capture general knowledge about the design process will replace the CAD tools currently used in engineering design. These new tools will capture not only a geometric representation of a product, but also a product's function, behavior, simulation models, and mathematical simulation models. One challenge in developing these tools is relating specific product function to their design equations. This paper discusses the initial research of extracting performance parameter equations from a product's functional model.

Constructing models is a fundamental challenge in engineering design. Practical challenges include developing sufficient understanding of physics and nature to enable the construction of design models, determining a relation between the underlying physics and customer relevant performance, and selecting model of the appropriate complexity for the problem at hand. The focus of this paper is on the second of these challenges: through product function, the underlying physics, and thus design parameters, are connected to performance, and thus customer needs.

To allow the generation, storage, and eventually computation, of knowledge that connects physics to design function, a common language is needed. Stone and Wood [1] have developed a common language to describe the mechanical design space. Their language, which they refer to as the functional basis, contains a set of definitions for functions and flows to describe a product function in a verb-object format. Their language will allow for the storage and retrieval of design information. Using the functional basis language to describe the product's function, a method is developed to generate descriptive equations for performance parameters during the conceptual design of a product.

In this paper, five different products were reversed engineered to provide function and equation data for a preliminary handbook for extracting performance parameters. One product, a water filter, is presented in detail to illustrate the validity of describing a product in terms of performance equations. It clearly illustrates our approach to creating mathematical models from functional models for conceptual design or product redesign. The results show that it is possible and useful to translate functional models into equation chains that relate back to customer performance parameters. Concepts like functional modeling and customer needs evaluation take the idea of design as an art that may or may not be successful and mold it into a science that consistently gives good, repeatable results.

2 Background

The high demand and relatively low supply seller's market of the 1950s and 1960s led to the, at the time, binary decision to produce innovative or mass produced products. Today, the market demands that businesses parlay into both types of production, creating high volume products that are tailored to customers' needs [2]. Thus, what used to be considered abstract, thought-provoking ideas about design are being developed into useful, consistent, repeatable processes for producing robust and successful products.

References on the history of functional design show research as far back as 60 years. Value engineering research performed during the 1940s as well as research into a common vocabulary for helicopter failure information during 1976 led to recent research into creating a functional basis for design [1, 3, 4, 5]. The current form of the functional basis is shown in Tables 1 and 2. Using the basis, a functional description (often called a sub-function) of a product is formed as a verb-object pair where a function word fills the verb spot and a flow fills the object position.

Table 1. Flow classes and their basic categorizations.

Class	Basic	Class	Basic	Class	Basic
Material	Human	Signal (cont.)	Control	Energy (cont.)	Hydraulic
	Gas	Energy	Human		Magnetic
	Liquid		Acoustic		Mechanical
	Solid		Biological		Pneumatic
	Plasma		Chemical		Radioactive
	Mixture		Electrical		Thermal
Signal	Status		Electromagnetic		

Table 2. Function classes and their basic categorizations.

Class	Basic	Class	Basic	Class	Basic
Branch	Separate	Control Magnitude	Actuate	Signal	Sense
	Distribute		Regulate		Indicate
Channel	Import		Change		Process
	Export		Stop	Support	Stabilize
	Transfer	Convert	Convert		Secure
	Guide	Provision	Store		Position
Connect	Couple		Supply		
	Mix				

Many reasons exist for starting product design with a functional model. Two of the main reasons addressed in this paper are maintaining the focus on the goals set by the customer performance parameters and increasing exposure to the possible incorporation of technology from other *similar* products. Product similarities can be somewhat surprising. For example, a SKIL screwdriver is 0.64 similar to a Krups cheese grater in terms of customer needs and functionality [6]. Another benefit is the ability to break the overall device function into easily solved sub-functions. Historically, smaller problems are easier to solve, and the flow interconnectivity allows these smaller problems to present themselves without losing the overall goal of the device function. Other benefits of functional design include having records of the product design to refer back to in case of similar product design/redesign and benchmarking [7] and increasing communication ease between different designers [8].

Product repositories are currently being developed to help classify products into families that share sub-functions and customer needs. The repository provides a quantitative measurement to determine the similarity of products based on their functionality and customer needs [6, 9]. Similar products are termed *families*. The products within these families can then be compared to each other during redesign to determine possible improvements to solutions for customer need satisfaction.

3 Approach

For this research, five consumer products operating on fluid flows are investigated as a potential product family. These products, which include a water filter, a portable electronic paint roller, a juicer, an air filter, and a pneumatically powered plane with a pump, are anticipated to share similar sub-functions. The functional models for each product are obtained by first finding customer needs and using the needs to create a black box model describing the material, energy, and signal flows that travel through each product [1, 10, 11]. A black box model treats the product as a box that describes the basic overall function of the device. No flows internal to the product are described during this step; rather only input and output flows are specified. The next step is to develop chains of sub-functions that begin as an input flow, end as an output flow, and describe the flow internally through the product. These function chains are easiest to define by "being the flow" as it moves through the product, describing each operation and transformation to the flow as it travels through the product. The active verb-object sub-functional description format and the functional basis are utilized to derive the function models. The final step combines the individual sub-function chains into a comprehensive functional model of the product. The functional model derivation method is shown schematically in Fig. 1.

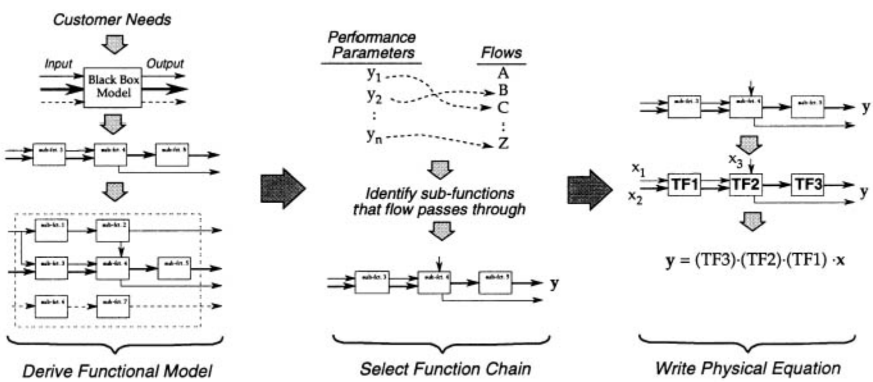


Figure 1. An overview of the approach taken to map customer needs to a physical equation.

Once a functional model exists, the physical equations describing a performance parameter of interest are extracted. Each sub-function of a functional model is in fact a high-level transfer function (TF) of the form $\mathbf{y} = \mathbf{C}\ \mathbf{x}$. The input flows, $\mathbf{x}$, represent the input variables and the output flows, **y**, will contain the performance parameter of interest. The sub-function description, such as *convert hydraulic energy to mechanical energy*, is a high level description of the TF dynamics, **C** (using the function and flow words from Tables 1 & 2). Placing these equations into a functional flow diagram similar to the functional model allows us to the see the mathematical equations behind the flow operations and transformations. From this diagram, chains of sub-functions relating to customer performance parameters can be extracted. The performance parameters given specific numerical requirements can then be used to determine constraints within the product design. These steps are illustrated in Fig. 1 as well.

4 Results

In order to illustrate the methodology followed to derive the functional model and equation model, we focus on the water filter first. The remaining four products will be presented at the end of the results section in order to show the product family relationships between devices.

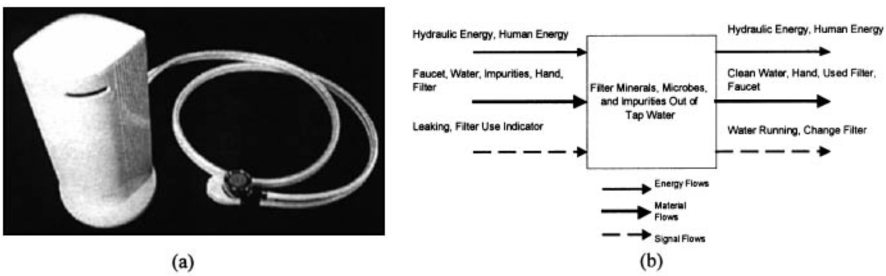


Figure 2. The PUR® water filter (a) and its black box model (b).

The black box model created for the water filter is shown in Figure 2. Based on customer needs, input flows are identified as the impure water with its hydraulic energy, the filter, the water faucet, human energy and hand. Signal inputs include the ability to tell if the device is leaking at the faucet junction and if the filter needs replacing. Outputs to the device include the purified water, any hydraulic energy not lost within the system, human energy, hand, and the full filter. Output signals include the visual cue of running water and filter status.

Next, function chains are created to show operations and transformations that occur with the flows as each one travels through the product. For instance, following the water as it flows through the water filter, first it is regulated by a hand-operated switch that determines whether the water is guided into the filtering system or guided directly through the nozzle and out of the system. If the water does not bypass the filtration system, it is guided down a tube and up into the core of the device where the water is measured by a turbine and gear system. If the water measurement exceeds the design specification for the amount of water that can safely be purified by the filter, then the water flow is stopped by a ball valve. Otherwise the water continues into the filter where it is refined, and the impurities are stored in the filter. Finally, the water is guided through the exit tube and is exported from the system by the nozzle at the end. If this verbal description of the path of the water flow is broken down into the verb-object form, a chronologically ordered chain of sub-functions can be created. The material branch of water shown in Figure 3 (as part of the complete functional model) illustrates the sub-functions associated with this verbal description of the water flow.

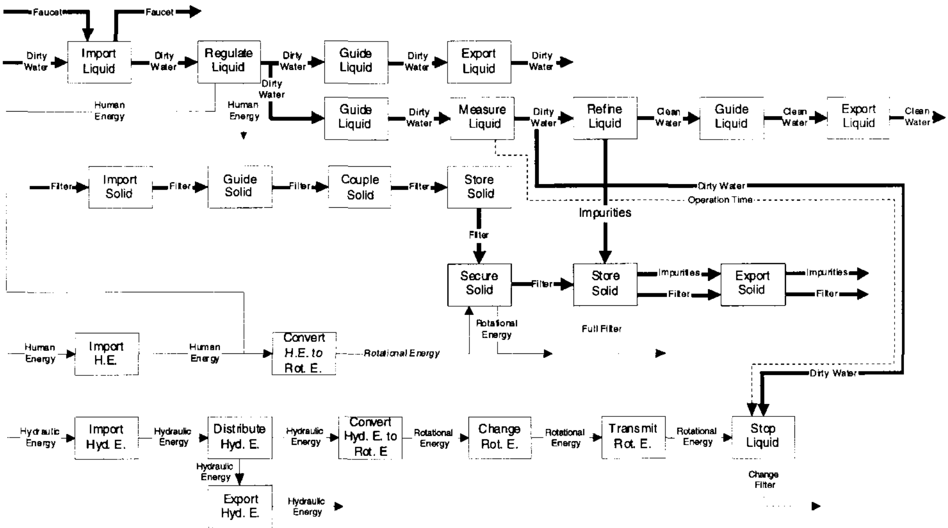


Figure 3. The functional model of the PUR® water filter.

From the functional model presented in Figure 3, the next step is to create a corresponding set of physical equations in place of the verb-object language used in the original sub-functions. The basic operation of the filter is to detour tap water through a filtering device until the filter is full, at which time the flow of water through the device stops. A potential redesign, based on customer need, is to increase the amount of water that the filter can process before reaching the filter's capacity. It is fairly easy to identify important sub-functions related to the redesign need: *refine liquid, store solid, measure liquid* and *stop liquid. Refine liquid* and *store solid* address the filter component and generate geometric constraint equations (i.e., volume, mass, filter porosity, etc.). The remaining two sub-functions address the dynamic issues of measuring and eventually stopping the unfiltered water. Converting *measure liquid* and *stop liquid* into physical equations (Fig. 4) identifies that additional information is needed to complete the kinematic expression describing the valve closure specific to the water filter. However, the sub-function *stop liquid* identifies that information as rotational energy – the input flow to the sub-function. The physical equations developed for the sub-function chain feeding into *stop liquid* are also shown in Fig. 4. Combining the equations (in similar manner as block diagram algebra for control systems) creates the overall equation describing the performance parameter of operation time as:

$$\theta_{close} = h \cdot \left[\frac{r_1 r_3 r_5 r_7 r_9}{r_2 r_4 r_6 r_8 r_{10}} \right] \cdot \left[\frac{1}{R_{turbine,eff}} \right] \cdot \left[\frac{Q_{in} - Q_{out}}{A_{turbine}} \right] \qquad (1)$$

This equation parameterizes the design problem. Solving for h, the operation time, identifies design variables related to turbine dimensions ($A_{turbine}$, $R_{turbine,eff}$), the gear train connecting the turbine and valve (gear radii r_i) and the angle required to close the valve (θ_{close}). From this point analysis strategies can be directly applied. The equation model makes it easy to quickly see the interdependency of design parameters and try out various combinations of parameters to determine the best overall design.

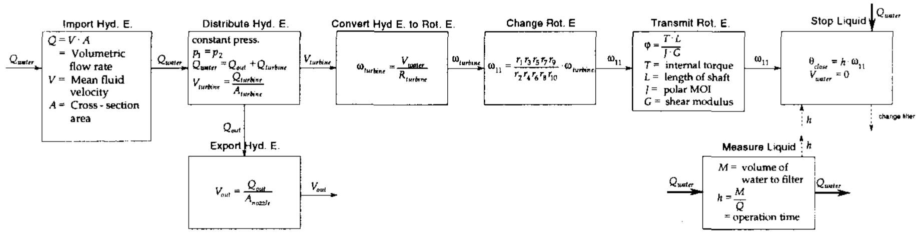


Figure 4. A physical equation model of the PUR water filter, derived from the overall functional model, used to develop the physical equation describing operation time performance.

While the equations in this example are derived from a specific product, *the individual sub-functions can be generalized and applied to any product with that functionality.* A possible implementation of this tool is a searchable matrix of sub-functions and physical equations set. If a sub-function description has a set of possible physical equations, then the designer would choose the best equation based on the form information available (a simple matrix entry is shown in Fig. 5).

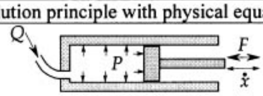
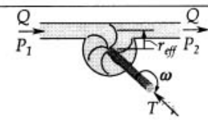
Sub-function description	Solution principle with physical equations	
hyd. ener. → [convert hyd. ener. to mech. ener.] → mech. ener.	 $F = P \cdot A$ $\dot{x} = \frac{Q}{A}$ where F = translational force P = hydraulic pressure A = cross-sectional area of piston Q = volumetric flow rate $\dot{x}$ = translational velocity	 $T = (P_1 - P_2)A \cdot r$ $\omega = \frac{Q}{A} \cdot \frac{1}{r}$ (approximately) where T = torque ω = angular velocity

Figure 5. Two forms and physical equation sets correlated to the sub-function convert hydraulic energy to mechanical energy.

The similarities between the five products examined in this work are summarized in Fig. 6. The products share many functions in common and the common functions lead to similar physical equations describing performance. Similarities such as these are the key to making the concept of a designer's equation handbook feasible.

5 Conclusions

The previous detailed example has illustrated that it is both possible and useful to translate functional models into equations chains that relate back to customer performance parameters. In addition, given the functional likenesses between products, the concept of a handbook of equations catered to the methodology of design remains a feasible and valid goal.

Material Flows							Energy Flows						
	Water filter	Juicer	Power paint roller	Air filter	Airplane	Airplane pump		Water filter	Juicer	Power paint roller	Air filter	Airplane	Airplane pump
Channel liquid		1					Actuate electrical ener.		1	1	1		
Export liquid	1	1	1				Actuate pneumatic ener.					1	
Guide liquid	1		1				Actuate rotational ener.					1	
Import liquid	1		1				Convert electrical ener. to rotational ener.		1	1	1		
Refine liquid	1		1				Convert electrical ener. to static ener.				1		
Regulate liquid	1						Convert electrical ener. to acoustic ener.		1	1	1		
Store liquid		1	1				Convert human ener. to translational ener.					1	1
Export gas				1	1	1	Convert human ener. to rotational ener.	1					
Guide gas				1		1	Convert pneumatic ener. to translational ener.					1	
Import gas				1	1	1	Convert rotational ener. to pneumatic ener.				1		
Refine gas				1			Convert pneumatic ener. to translational ener.		1	1		1	
Separate gas					1		Convert translational ener. to acoustic ener.					1	
Store gas					1		Convert translational ener. to pneumatic ener.			1			1
Channel solid		2					Convert translational ener. to rotational ener.					1	
Connect solid				1			Export human ener.	1	1	1	1	1	1
Couple solid	1					1	Export pneumatic ener.			1	1		1
Export solid	1	1	1	1		1	Export static ener.				1		
Guide solid	1			1			Export translational ener.		1			1	
Import solid	1	1	1	2		1	Import electrical ener.		1		1		
Secure solid	1					1	Import human ener.	1	1	1	1	1	1
Separate solid		1					Import pneumatic ener.					1	
Store solid	2	1	1	2		1	Measure pneumatic ener.						1
Supply solid				1		1	Regulate translational ener.					1	
Import human	1	1	1	1	1	2	Store pneumatic ener.					1	
Export human	1	1	1	1	1	2	Supply electrical ener.			1			
							Transmit rotational ener.		1	1	1	1	
							Transmit translational ener.					1	
							Transmit weight	1	1	1	1	1	1

Figure 6. Matrix showing coincident sub-functions between products. The number indicates the frequency of occurrence for each sub-function in the product.

This example represented a very simplified case of the types of similarities that could occur among products. The results show that energy and signal flows lead to dynamic equations, and material flows lead to constraint equations that relate specifically to the product being described. In practice, energy flows tend to be the most transferable between products that may not seem similar at all. For instance, few ways exist to convert translational energy into rotational energy besides a cam system, whether for a pneumatically powered toy airplane or a car engine. Therefore, the general equation(s) for both systems will be the same. Where the difference lies will be in the basic assumptions and the scale of the variables used to complete the equations. In an effort to further evolve design methodology, this paper establishes the beginning of a handbook of reference equations that can be developed from a product's functional model.

6 Future work

The products analyzed in this experiment represent a specific subset of products with fluid flow characteristics. Future work in this field will expand and build on this research and include other product sets representing groups in heat transfer, structural engineering, electrical engineering, kinematics, pneumatics, and other classifications. Further research should be assembled into a singular reference to aid in the design and redesign of consumer products of all types.

Acknowledgements

This work is supported by the National Science Foundation under grant DMI-9988817 and the University of Missouri-Rolla Opportunities for Undergraduate Research Experiences program. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators.

References

[1] Stone, R. and Wood, K., "Development of a Functional Basis for Design," Journal of Mechanical Design, Vol. 122, No. 4, 2000, pp 359-370.

[2] Erens, F. and Verhulst, K., "Architectures for Product Families," Computers in Industry, Elsevier Science B.V, Vol. 33, 1997, pp 165–178.

[3] Miles, L., "Techniques of Value Analysis Engineering," McGraw-Hill, 1972.

[4] Collins, J., Hagan, B., and Bratt, H., "The Failure-Experience Matrix - a Useful Design Tool," Transactions of the ASME, Series B, Journal of Engineering in Industry, 98, 1976, pp 1074-1079.

[5] Hirtz, J., Stone, R., McAdams, D., Szykman, S. and Wood, K., "Evolving a Functional Basis for Engineering Design," Proceedings of DETC2001, DTM Conference, DETC2001/DTM-21688, Pittsburgh, PA, 2001.

[6] Stone, R., Wood, K., and Crawford, R., "Using Quantitative Functional Models to Develop Product Architectures," Design Studies, Vol. 21, No. 3, 2000, pp 239-260.

[7] McAdams, D. and Wood, K., "A Quantitative Similarity Metric for Design by Analogy," to appear in The Journal of Mechanical Design, 2000.

[8] Kurfman, M., Stone, R., Rajan, J., Wood, K., "Functional Modeling Experimental Studies," Proceedings of DETC2001, DTM Conference, DETC2001/DTM-21709, Pittsburgh, PA, 2001.

[9] McAdams, D., Stone, R., and Wood, K., "Functional Interdependence and Product Similarity Based on Customer Needs," The Journal of Research in Engineering Design, Vol. 11, No. 1, 1999, pp 1-19.

[10] Otto, K. and Wood, K., "A Reverse Engineering and Redesign Methodology for Product Evolution," Proceedings of DETC1996, DTM Conference, 96-DETC/DTM-1523, Irvine, CA, 1996.

[11] Pahl, G. and Beitz, W., "Engineering Design: A Systematic Approach," Springer-Verlag, 1988.

Robert B. Stone
102A Basic Engineering Bldg., University of Missouri Rolla, Rolla MO 65409-0210, U.S.A., Tel: (573) 341-4086, Fax: (573) 341-6593, E-mail: rstone@umr.edu, URL: www.umr.edu/~rstone.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

TOWARDS A MECHANICAL SYSTEMS MODELLING AND OPTIMAL EMBODIMENT METHOD

B Eynard and P Lafon

Keywords: Product Modelling, Functional Modelling, Geometric Modelling, Optimisation Techniques

1 Introduction

In the development process of mechanical systems a large part of the design time is spent in routine tasks, as the embodiment and calculation choices of sub-assembly. This paper aims to show the benefit of a product modelling and optimisation techniques and some speculations on their implementation as an aided tool in conceptual and embodiment design phases of mechanical systems [1]. We will first present the product modelling and the optimisation techniques in order to describe in a second time their application in a case study based on the design of a ball bearings sub-assembly.

2 Product Modelling

In [2], [3] we proposed a product model in order to formalise the knowledge generated by designers during the product development process. Let us specify that these works are based on the graph theory [4] and on two French works [5], [6] dealing with product modelling in conceptual design.

Knowledge allows to describe the product according to a hierarchy of successive states, each time more detailed. This description is based on the concepts of function, technological entity and boundary entity. Considering a given state of product knowledge noted $ep_k \in MP$ (definite number of states of product knowledge), it can be specified as a quintuplet noted $<F_k, ET_k, EF_k, li_k, att_k>$ where the mathematical functions li_k and att_k characterise the existing relations between the elements of :

- F_k definite number of functions ;
- ET_k definite number of technological entities ;
- EF_k definite number of boundary entities.

The function facilitates the transcription of the functional requirements list (FAST diagram, ...) and the design progress in order to satisfy the technical objectives like guiding, transmitting, linking, ... The technological entity ensures the translation of concepts and principles in the components of geometric or physical model usually used by designers. It allows to specify the material concretisation of the product. The technological entity characterises the physical components which support the technical concepts and principles satisfying the functions. The specification of a technological entity is provided by the specific interaction or attach points which are maintained with its environment (connection points with the function). All these interaction points represent the boundary of the technological entity and specify the nature of the objects linked by the function. These interaction points are named boundary entity. The functions, technological and boundary entities are specified through their attributes. These attributes should provide the key characteristics of mechanical system like speed and torque for a transmission system or strength for a mechanical contact. Some of these attributes will be used in the optimal embodiment step. Figure 1 describes the graphical formalism used in order to model a mechanical system. Based on the concepts of functions, technological entities, boundary entities and attributes we are able to specify the main characteristics and elements used for the formulation and the solving of the optimal embodiment design problem.

3 Optimal embodiment design

The optimisation step consists in the embodiment and calculation of a sub-assembly gradually specified by the designer and whose knowledge, at a given time, could be described in the product model. This embodiment activity can only be performed at a sufficient advanced step of the design process. Step where the whole made choices allows to specify the technological solutions as well as their needed data satisfying the basic mechanical pairs. This step allows designers to quickly evaluate the sub-assembly regarding a non-explicit set of evaluations criteria. By modifying one or more data they can also evaluate the influence of these data on the design solution. The optimal embodiment of the sub-assembly requires the formulation of a "generic" optimisation problem able to take in account most of the design cases of the sub-assembly. Generally this kind of optimisation problem is expressed with a non linear objective function to be minimised and use several non linear constraint functions. When the sub-assembly requires the use of standardised or normalised mechanical components (bolted connection, bearings, …), vector with mixed variables (discrete and continuous variables) are used in the formulation of the optimisation problem. Some specific optimisation algorithms have been developed to solve this kind of problems [7], [8]. Those algorithms are based on a particular class of evolutionary optimisation method : the evolutionary strategies. Evolutionary strategies mimic the natural process where a population of individuals (an individual = a value of the vector variable) undergoes transformations of recombination, mutation and selection. By means of randomised processes, the population evolves toward better and better regions of the search space defined by the constraint functions of the problem.

4 Application

The paper now develops the case study of an optimal embodiment of ball bearings in order to show the efficiency of coupling a product modelling and an optimisation techniques. We will consider a sub-assembly satisfying the technical function "Guiding a rotational motion" belonging to a power transmission system. In this study case, we deal with a turning pair realised with bearing components splitting it up into a spherical pair and a spherical-and-cylinder pair. First we will show the main step of the product model evolution in this case. Second we will present the formulation of the optimisation problem and its solving. Finally, we will develop some speculations on the use of the described method in an industrial context.

4.1 Breakdown of the product model

The specifications of the product are obtained through a progressive refinement of the product model. The graphical formalism used in order to model a mechanical system and its breakdown detailed models are presented. Figure 1 shows the first level of the product modelling specifying the preliminary state of product knowledge. The function "Guiding a rotational motion" linked the two technological entities *Body1* and *Shaft2* where the attach points are *AxB1* and *AxS2*. Concerning the attributes we can specify for the technological entity *Body1* : a reference point PtB1, a reference axis AxB1, a Trihedron and kinematics specifications (strength, torque and application point AB1).

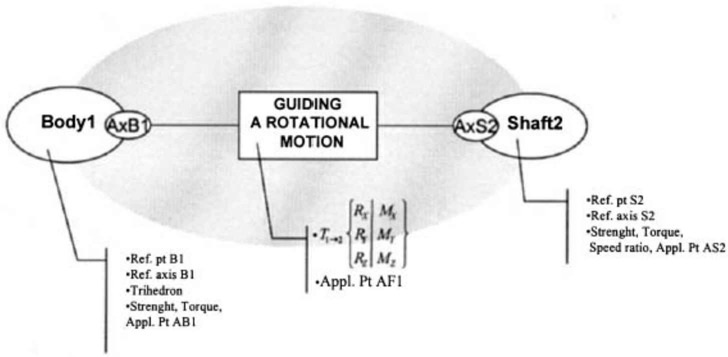


Figure 1. First state of product knowledge

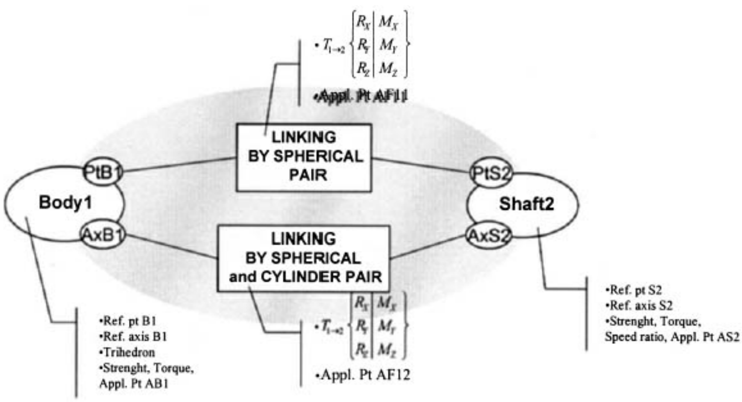


Figure 2. Second state of product knowledge

In a second step, refining the preliminary state of product knowledge, we detail the splitting up of the function into two sub-functions in order to specify the kinematics behaviour of the mechanical system (figure 2). The sub-functions are represented in a parallel breakdown. The other able kind break down is the series one (see bellow with the example of functions linking the *Body1 – Bearing2 – Shaft2*). The breakdown should be performed according to the technological behaviour of the mechanical system and always controlled and assessed regarding the functional requirements list. Regarding our case study, figure 3 describes the last state of product knowledge which is the modelling of the chosen technological solution using ball bearings components. Based on this choice and the specified attributes of functions and technological entities, we are able to perform the formulation and the solving of the optimal embodiment design problem.

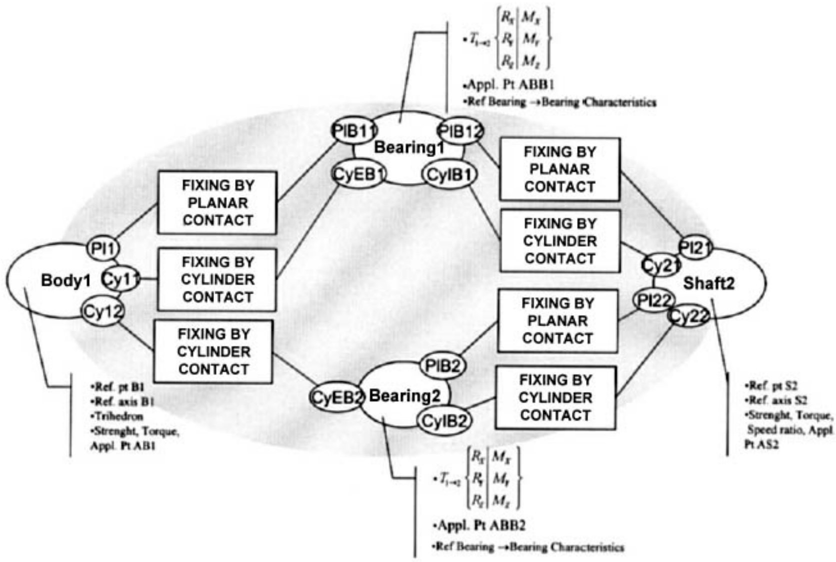


Figure 3. Last state of product knowledge

4.2 Formulation of the optimisation problem

The optimal embodiment of the bearings consists in choosing the two ball bearings and calculating their position to minimise the mass of the bearings and the shaft. This minimisation represents the objective function to be achieved. We propose a generic optimisation problem taking into account all the case of a turning pair realised with bearings. In this study we consider the case where this turning pair is split up into a spherical pair and a spherical-and-cylinder pair (see figure 4).

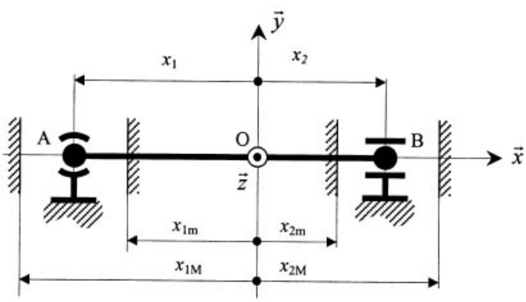


Figure 4. The turning pair

At this earlier step of the design, very few data are available. However the optimal embodiment only requires the knowledge of the strength supported by the turning pair, the geometrical bulk, the life duration L_v (in hours) and the speed ratio ω (tr/min) of the pair. Here the different components of the strength (R_{XO}, R_{YO}, R_{ZO}) and the torque (M_{YO}, M_{ZO}) are supposed to be given at the point O. Thus, we are able to formulate an optimisation problem with 4 variables : 2 continuous variables (x_1; x_2) specifying the position of the bearings and 2 discrete variables (R_1; R_2) referencing the number of the bearings in a manufacturer catalogue. This problem contains 9 constraints functions written as mathematical inequalities. Those constraints functions define the sub space of the whole solutions of this problem. Constraints functions g_1 and g_2 express that the life duration L_v of the sub-assembly specified in the functional requirements lits must be inferior to the life duration of bearings (2) and (3). Constraints functions g_3 to g_9 impose the different limitations on the position and the geometrical bulk of the ball bearings (4) to (10).

$$F(R_1, R_2, x_1, x_2) = m_1 + m_2 + x_2 - x_1 \tag{1}$$

$$g_1(R_1, R_2, x_1, x_2) = P_1\left(\frac{60 L_V \omega}{10^6}\right)^{1/3} - C_1 \leq 0 \tag{2}$$

$$g_2(R_1, R_2, x_1, x_2) = P_2\left(\frac{60 L_V \omega}{10^6}\right)^{1/3} - C_2 \leq 0 \tag{3}$$

$$g_3(R_1, R_2, x_1, x_2) = x_{1m} - \left(x_1 - \frac{b_1}{2}\right) \leq 0 \tag{4}$$

$$g_4(R_1, R_2, x_1, x_2) = \left(x_1 + \frac{b_1}{2}\right) + x_{1M} \leq 0 \tag{5}$$

$$g_5(R_1, R_2, x_1, x_2) = x_{2m} - \left(x_2 - \frac{b_2}{2}\right) \leq 0 \tag{6}$$

$$g_6(R_1, R_2, x_1, x_2) = \left(x_2 + \frac{b_2}{2}\right) + x_{2M} \leq 0 \tag{7}$$

$$g_7(R_1, R_2, x_1, x_2) = d_{1m} - d_1 \leq 0 \tag{8}$$

$$g_8(R_1, R_2, x_1, x_2) = d_{2m} - d_2 \leq 0 \tag{9}$$

$$g_9(R_1, R_2, x_1, x_2) = D_1 - D_M \leq 0 \tag{10}$$

In this optimisation problem, there are some discrete parameters, whose values depend on the number R_1 and R_2 of the ball bearings. These discrete parameters, extracted from the manufacturer catalogue, are : m_1 and m_2 the mass of the bearings, C_1 and C_2 the dynamic load rating, d_1 and d_2 the bore diameter, D_1 and D_2 the outside diameter, b_1 and b_2 the inner ring width. Here P_1 and P_2 are the equivalent dynamic load calculated [9] with the dynamic radial (F_{R1} and F_{R2}) and thrust load (F_{A1}) given the following relation (11), (12) and (13) :

$$F_{R1} = \frac{1}{|x_1 - x_2|}\sqrt{(x_2.R_{YO} - M_{ZO})^2 + (x_2.R_{ZO} + M_{YO})^2} \quad (11)$$

$$F_{A1} = |R_{XO}| \quad (12)$$

$$P_2 = F_{R2} = \frac{1}{|x_1 - x_2|}\sqrt{(x_1.R_{YO} - M_{ZO})^2 + (x_1.R_{ZO} + M_{YO})^2} \quad (13)$$

It is very easy to take into account the direction of assembly of the subset bearings + shaft. This can be done by adding another constraint function g_{10}. For example, an imposed assembly from the right side can be expressed as bellow in (14) :

$$g_{10}(R_1, R_2, x_1, x_2) = D_2 - D_1 \leq 0 \text{ if } x_1 \leq x_2 \quad (14)$$

or :

$$g_{10}(R_1, R_2, x_1, x_2) = D_1 - D_2 \leq 0 \text{ if } x_1 > x_2$$

4.3 Some results

Two typical results of this optimal embodiment are represented. Evolutionary strategies are based on stochastic method, so to obtain significant results, we executed 10 runs of 5 generations. For each generation we have used a population of 30 individuals and 200 offspring's. This calculus take a few seconds on a HP9000 workstation. This results were obtained with a manufacturer catalogue of 32 radial contact single row ball bearings.

Table 1. Data used for the tow examples

R_{XO} = 2000 N	ω = 970 rpm	x_{2M} = 100 mm
R_{YO} = 4200 N	L_V = 1800 hours	d_{1m} = 30 mm
R_{ZO} = 11600 N	x_{1m} = -100 mm	d_{2m} = 30 mm
M_{YO} = 0	x_{2m} = -20 mm	D_M = 130 mm
M_{ZO} = 0	x_{1M} = 20 mm	

Table 2. Optimal embodiment results

	Case 1		Case 2	
	R1	R2	R1	R2
Ref SNR	6306	6209	6209	6207
Life duration (hours)	1800	2154	1891	1800
Value of the objective function	823.80		767.28	

We want to illustrate the influence of the bad choice of a assembly direction. In case 1, we have imposed an assembly direction from the right. In case 2, no direction were imposed Data and results are summarised in table 1 and 2. These data come from the attributes declared in technological entities of the product model. Figures 5 and 6 show the position and the dimensions of the ball bearings for this tow cases. We can notice that case 2 offers a much better technical solution because the life duration of ball bearings is similar where as the mass (indicated by the value of the objective function) is lower than the mass for case 1.

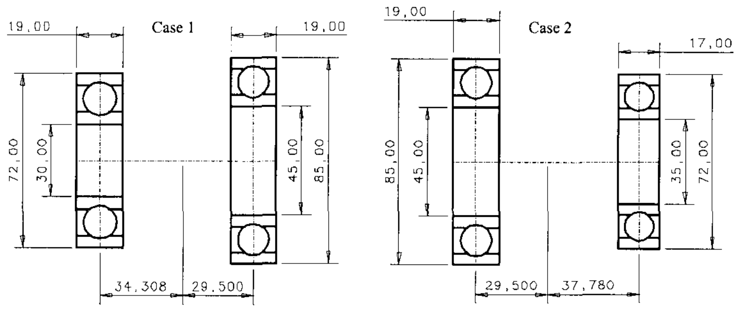


Figure 5. The two cases of optimal embodiment.

4.4 Speculations on the method implementation

The presented draft method based on product modelling and optimisation techniques allows to evaluate the possibilities provided by a computer system supporting the proposed method and aiding the designer during the conceptual and embodiment design phases. Such a system should have the following integrated modules and features :

- Regarding the product modelling, a drawing and managing assistance will be necessary. Indeed the several states of model may become quite complex to understand and to handle. A module for browsing in the models could then also provide an efficient aid.

- Concerning the optimisation techniques, a large set of generic cases of technological solutions should be identified. Based on this set, the relevant formulation of optimisation problem have to be performed for each identified case. Then the solving algorithm requires to be implemented with the needed functionnalities in order to manage the generic cases and their solving.

- Finally, a link with a CAD System should be provided. This link will allow the implementation of the method in large scope of business case. A strong integration with a CAD System could benefit because this system have currently an important use in mechanical and automotive industries and particularly in design department. Then the method could become an interesting support in conceptual and embodiment design.

5 Conclusion

In this paper we use a product modelling technique and we apply to the case study of a technical function "Guiding in rotation" realised by ball bearings components. This modelling technique is based on the concepts of function, technological and boundary entities. In this way, knowledge breakdown rules and guidelines in order to structure the hierarchical description of the product model have to be provided. For being able to optimise the ball bearings sub-assembly, it is necessary to add the relevant and characteristic attributes to each function, technological and boundary entity. These attributes are the link between the proposed product modelling and optimal embodiment techniques. The formulation of the optimisation problem with non-linear and mixed variables uses various attributes of the studied mechanical system but also of manufacturer catalogue. The solving of this problem has led to the development of specific optimisation algorithms.

To summarise, the coupling of product modelling and optimal embodiment techniques allows the development of a generic method usable in the conceptual and embodiment design of mechanical systems. A case study of ball bearings sub-assembly illustrates a draft proposal of the method.

References

[1] Pahl G., Beitz W., "Engineering Design : a Systematic Approach", Springer, London, 1996

[2] Eynard B., "Modélisation du produit et des activités de conception – Contribution à la conduite et à la traçabilité du processus d'ingénierie", PhD dissertation of Bordeaux 1 University, France, 1999

[3] Eynard B., Belloy Ph., Lafon P., "Towards an Integration of Knowledge in Mechanical Engineering", Proc. 3rd International Conference on Integrated Design and Manufacturing in Mechanical Engineering, Montréal, Québec, Canada, May 2000

[4] Graham R., Grötschel M., Lovász L., "Handbook of Combinatorics – vol. 1", Elsevier Sciences, Amsterdam, 1995

[5] Dupinet E., "Contribution à l'étude d'un système informatique d'aide à la conception de produits mécaniques par la prise en compte des relations fonctionnelles", PhD dissertation of Ecole Centrale de Paris, France, 1991

[6] Tollenaere M., Constant D., "Linking conceptual and embodiment design of mechanical systems", Proc. 10th International Conference on Engineering Design - ICED 97, Tampere, Finland, August 1997

[7] Moreau-Giraud L., Lafon P., "Evolution strategies for optimal design of mechanical systems", Proc. Third World Congress of Structural and Multidisciplinary Optimization (WCSMO-3), Buffalo, USA, May 1999

[8] Giraud L., Lafon P., "Optimal design of mechanical components with genetic algorithm", Integrated Design and Manufacturing in Mechanical Engineering, Kluwer Academic Publisher, Dordrecht, 1999

[9] Beitz W., Küttner K.-H., "DUBBEL : Handbook of mechanical engineering", Springer Verlag, 1994

Corresponding author's : Dr Benoît EYNARD
Lab. of Mecanical Systems and Concurrent Engineering (LASMIS), Troyes University of Technology, 12 rue Marie Curie, BP 2060, F.10010 TROYES CEDEX, FRANCE, Tel : +33 3.25.71.56.71, Fax : +33 3.25.71.56.75, E.mail : benoit.eynard@utt.fr, http://www-lasmis.utt.fr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A COMPUTATIONAL MODEL FOR SUPPORTING CONCEPTUAL DESIGN OF AUTOMATIC SYSTEMS

E A P Santos, V J de Negri, and J E R Cury

Keywords: Computer-aided design, functional modeling, systems modeling, engineering diagrams.

1 Introduction

The conceptual design is the phase of a design process that generates, from a detected and clarified need, a conception that supplies this need optimally, under resource and design restrictions [1][2]. However, despite the many attempts, there have not been well-succeeded methodologies to development of design or design algorithms that can support the conceptual design of technical systems [3][4].

In this context, the present work proposes a model for a computational system for the support of conceptual design of automatic systems. Such systems are mentioned in the literature as mechatronic systems [5]. Being so, concepts from such research area are used. This work also introduces the control theory for discrete event systems as a formal tool for the synthesis and analysis of controllers.

The work takes the functional analysis at the design process as a starting point. The model proposes the use of functional modules, elaborated from the composition of proposed functions for automatic systems. The computational system is supposed to load information on the modeling of the functional modules, specifications and to enable one controller for each one of the modules, called local controllers. Next, after the designer has finalized the functional structure, the tool will, by means of verification algorithms, check the non-blocking property in the concurrent operation of the controllers.

Initially, the paper shows the modeling of automatic systems, highlighting the restrictions in the use of previously established models. In this sense, the use of the Channel/Instance net (De Negri [6]) is proposed as a central model of the physical design and the control design. The discrete character of the functions in automatic systems is discussed. Section 3 presents the control theory for discrete event systems as introduced by Ramadge and Wonham [7], as well as the extensions proposed by De Queiroz and Cury [8]. Finally, the whole computational model and an example of application are presented.

2 Modeling automatic systems

The development and application of design methods for automatic systems come after the use of methodologies for mechanic systems, and being so, methodologies that cover automatic systems naturally use fundamental concepts from the mechanic design area. The present work explores the functional analysis as a fundamental tool for the development of the automatic system, extending some concepts for a better fitting to the systems under study. According to this proposal, the section approaches the functional analysis, its limitations and the points to be considered at the functional modeling of automatic systems.

2.1 Functional modeling

There is one specific difficulty in using functional structures according to german schools of design theory, coming from the many problems related to its interpretation. According to Svendsen and Hansen [3], the functional analysis is done on well known, in simple systems and no attention is dedicated to explain the method or the laws on which the decomposition is based. The functional analysis has not got until then, a theoretical basis. Another problem is the diversity of interpretations for the term *function* found in the literature, which results in a dubious interpretation of functional structures. As pointed out by Umeda and Tomiyama [9], some functions are not properly represented, considering the energy, matter and information flows. This fact tends to cause problems when the construction of computational tools for design support is proposed.

This is observed in the function structures used in Pahl & Beitz [1], Hubka & Eder [2] and Buur [10], which do not establish a clear distinction between attributes and energy or matter. Consequently, the diagrams lead to two interpretations for the designer: a) As functions that receive matter and/or energy and transform them in matter and/or energy that will be present in other(s) place(s); b) As functions that, due to the value of the attributes of the energy and/or matter in the addressed arcs for the function block, cause the modification of the value of the attributes of the energy and/or matter indicated into the arcs that leave the function block.

However, the authors state that, despite functional structures are still essentially semantic and highly abstract, they are the first models to incorporate information on the structure of the product or the system. According to the proposal of the present work, it is desired that since the early phases, the designer could make use of representations that naturally lead the functional model to the final physical structure of the system to be developed. Being so, the diagrammatic representation of a structure must, at a certain part of the conceptual design, has the same importance of the interconnections between machines and devices. It's intended to obtain a non-ambiguous model, in order to enable the control system design, from the functional analysis. In this sense, the use of the Channel/Instance net (which is discussed next) is proposed.

2.2 The Channel/Instance Net

The Channel/Instance net [6][11] consists of a diagrammatic representation that makes use of two basic elements: the active functional units (instances), represented by rectangles, and the passive functional units (channels), represented by circles, being both linked by oriented arcs. The channels indicate system components that give support, so that the resources can flow without causing changes on its state. As examples, one can list pipes, fixate axes, wires, transporting belts, buffers, memories, etc. The instances correspond to the place where the activities happen, such as machine components, working stations, chemical reactors, objects (software), among others. In the present work, the use of C/I nets is proposed as a central model for automatic systems. The following aspects justify this choice: makes use of fundamental elements (channels and instances), making possible the dual descriptions of functional and structural perspectives; it's totally free from implementation or production solutions; explicates the physical interconnection between machines or devices (the channels are the place where the matter flows by).

2.3 Functional modeling of automatic systems

Many authors have identified general functional structures usual on automatic systems. A common point is that besides the primary function (total function), a set of auxiliary functions must exist in order to help the total function [10]. According to De Negri [6], the operation of an energetic or material system depends on the action of an information system, human or not, which is able to extract information from this first one, process them, and later on, use them to alter the energetic and material flow. This way, an automatic system can be modeled as an information system that is attached to an energetic and/or material system by internal information channels. Besides the exchange of information between these two sub-systems, there's also the input and output of energy, matter and information in relation to the external environment. This perspective is modelled according to Figure 1. The information system consists of instruments, programs, human being or any other means that process signals. The material energetic system is an abstraction of the machines, devices, processes, etc, able to make physical or chemical changes. An automatic system can be refined until there is evidence of the measuring and acting systems that concretize the internal information flow.

One can see in Figure 1 that the model shows simultaneously two perspectives: structural and functional. The functional model specifies what the system does or should do. As discussed previously, functionality is the main perspective for the system specification. The structural model describes where the functions are implemented, representing then, the final objective of the system, which is to establish its components. The term *structure* refers to the set of elements in the system and the set of relations that interconnects these elements. For the global design of the automatic system, it's necessary to elaborate its behavioural perspective, that is, to show how and when the functions are executed. This fact points to the use of new approaches, in order to also include in the model, information that take the formal design of the control system.

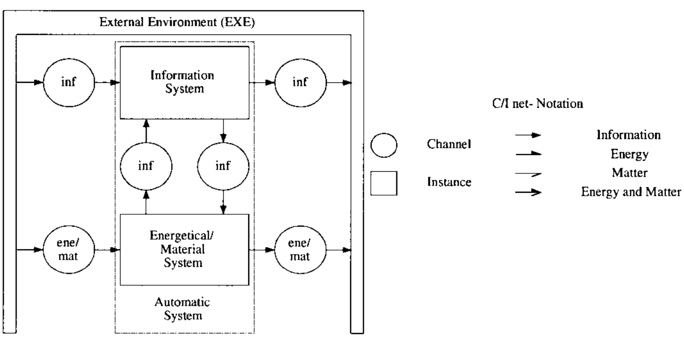


Figure 1. General model for automatic systems [10].

2.4 States and transitions

According to several authors, the function of an automatic system not only depends on the combination of the input variables, but also on the previous activities (memory) of the system. The same input combination can enable different functions at different points in time. Being so, the functional structure of an automatic system is variable and modifies with the momentary state of the system [10]. The automatic system can then be modeled with a machine of finite states, being able to be described in terms of states and transitions.

However, Buur [10] points out to a possible conflict between the two ways of modeling functions of automatic systems: the transforming approach [1], where function is described as a continuous matter, energy and information flow, and in the discrete approach, where the system changes the working method from one state to another. While a complete structure of the transformation does not explicitate the different states of a system, in a modeling through state machines the information flow is not clear (information is required for a state change). In this sense, authors make use of both transformation functions and discrete models (state transition diagrams, Petri nets) in the same design model [10] [12].

The interpretation of Buur [10] does not consider the representation of the composed behaviour of the functions in an automatic system, that is, the composition of the models of each one of the functions. From this representation, restrictions to the system functioning can be imposed, or, in other words, a behavioral perspective can be inferred to the automatic system. The discrete models used together with functional structures [12] are limited to representations, and are not used in a formal of synthesis of the information system. For such reason, is introduced the Supervisory Control Theory for Discrete Event Systems.

3 The Supervisory Control for Discrete-Event Systems

Discrete event systems (DES) are dynamic systems that evolve in accordance with the abrupt occurrence of events. They are generally asynchronous (not clock driven) and non-deterministic (some events may occur spontaneously). Such systems are encountered in a variety of fields, for example in manufacturing, robotics, computer and communication networks, traffic, and logistics.

The supervisory control of DES in accordance with behavioral specification is a new area that is receiving increasing attention. The approach introduced by Ramadge and Wonham [7] is based on information feedback on the occurrence of events and controlled automata concepts. It offers two important advantages over other approaches: the resulting controlled behaviors do not contradict the behavioral specifications and are non blocking: the supervisor and control laws obtained are correct by construction; and the controlled behaviours are maximally permissive within the behavioral specifications: all events that do not contradict the specifications are allowed to happen.

The theory for DES considered in this work is based on controlled automata concepts [13]. The behaviour of DES, such as automatic systems, is modeled naturally by finite state automata. For the purpose of supervisory control system development, plant behaviors and corresponding behavioral specifications are modeled in the form of directed transition graphs.

We will limit the discussion to centralized and local modular supervisory control. Centralized supervision [7] implies that overall supervisory task is carried out by one single supervisor embodying the closed-loop behaviour of the plant. Furthermore this supervisor must be provided with complete information on the occurrence of events in the plant. Local modular synthesis [8] induces a natural decentralized structure for supervisors. Each local supervisor only needs to exchange information with its corresponding local plant. In case of changes in the plant or in the specifications, respected the non-blocking condition, the control modules can be redesigned, based only on local information. A distributed control system with more flexibility and higher computer simplicity is obtained. Those results will be used in this work for the system control design, for all the inner advantages of this approach.

Taking into account the supervisory control system architecture adopted (e.g., centralized, local modular) the transition graphs representing the physical behaviour of the system to be controlled and the transition graphs representing the corresponding specifications are fed to a computer program which will return, if these exist, the supervisor(s) embodying the maximally permissive controlled behaviour of the system within the specification, and the corresponding control law(s).

4 A computational model

The main objective of this computational model is to give support to the construction of a computer-aided tool for the design of automatic systems. The designer will be able to build, in this computational environment, functional structures from commonly used modules in automatic systems. The modules carry information about the model of each function, the operation specification and the resultant supervisor. This proposal is illustrated in Figure 2.

In Figure 2a, a functional module is built from the series composition of two instances. Each one of these instances is modelled as a finite state automaton, as in Figure 2b. Observe that each state of the automate represents the inactive or active instance, respectively. The working specification for this module is the non-occurrence of overflow and underflow in the link channel between these two instances, that is, Instance 1 cannot be activated when the channel is busy, and, on the other hand, Instance 2 cannot be activated if the channel is empty. This specification is illustrated in Figure 2c. By applying the results of the supervisory control theory [7], one can synthesize a controller that matches the specification. Such controller is specified in Figure 2d.

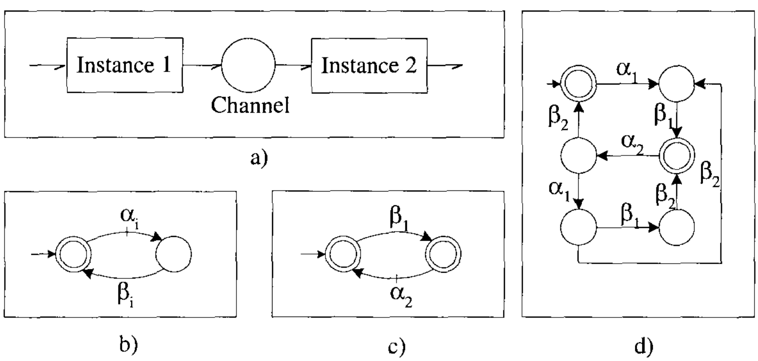


Figure 2. a) Functional module; b) Automata to Instance i, i = 1, 2; c) Automata to specification; d) Automata to controller.

In this way, many functional modules are stored in the computational tool. An example of application is shown in figure 3. The functional structure represents a unit for processing parts. There are two main activities to be developed: one drilling operation (I.3) and one testing operation (I.5). This last one must check whether the hole was drilled properly. The instances I.1 e I.7 represent the activities of supplying and catching parts to be processed, respectively. The instances I.2, I.4 e I.6 represent the activities of part transport in the unit, that is, the positioning of parts to be drilled and tested, in this order.

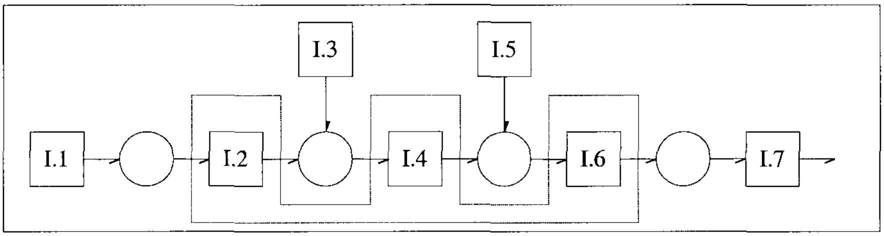


Figure 3. Functional structure for a unit of processing parts.

As formerly mentioned, the proposed computational model must store information on the functional module, shown in figure 3. That is information on the models of each instance and specifications on each operation. Such models are presented on Figure 4.

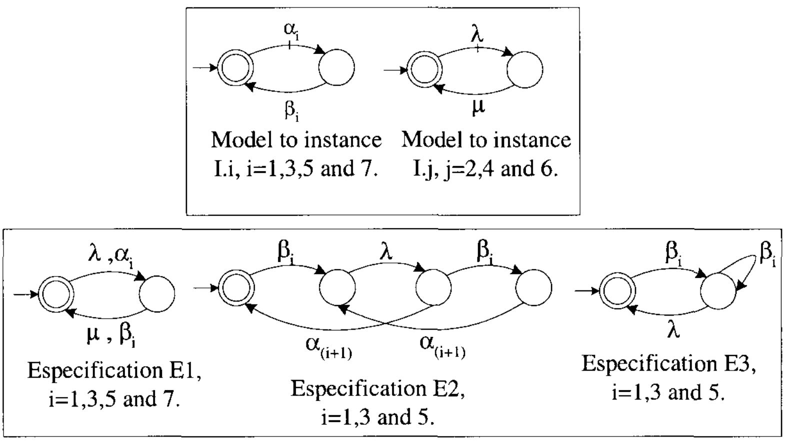


Figure 4. Models of instances and specifications of functional module shown in figure 3.

By applying specific algorithms the computational tool returns controllers to each of the specifications, which are called local controllers. After this, it is necessary that the computational tool, using appropriated algorithms, checks for the modularity property [7] between controllers. Such property, if detected, assures that the concurrent operation of the local controllers is non-blocking. Finally, the tool turns the local controllers, represented until then by automates, into a PLC programming language, like the Ladder diagram or the SFC (IEC 848) [14].

One can observe that the controllers of the functional modules are designed before the phases of principles solution search for the functions. The designer will be able to choose among many technologies, one that matches the designed local controller. So, one can naturally obtain a hierarchical and decentralized control structure, with their intrinsic features, such as good maintenance, error detection and easy modifications on the physical structure of the system. For example, the designer will be able to substitute a specific mechanism in operation by another one without having to modify the whole control structure of the system.

A possible set of solution principles for this part processing unit is shown on figure 5. The reader should notice that activities I.2, I.4 and I.6 are worked out by the same solution principle, a rotating table. The main activities I.3 and I.5, drilling and testing, respectively, can be carried out by pneumatically powered mechanisms. The supply of parts, activity I.1, can be carried out by a conveyor belt. The process of collecting the parts, activity I.7, can be carried out through vacuum.

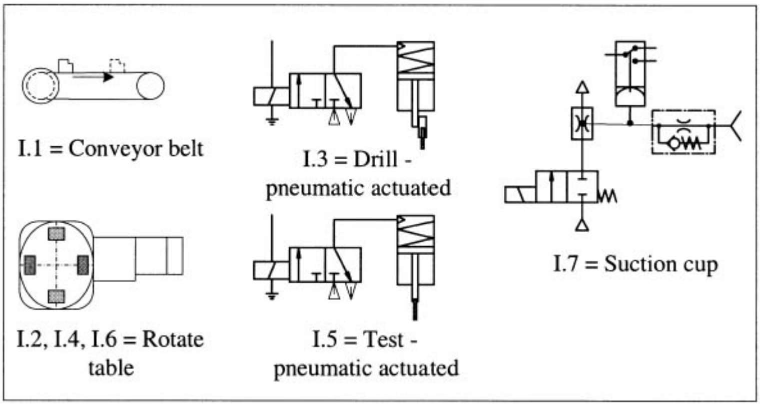


Figure 5. Principles solution for functional module shown in figure 3.

Notice that the functional module, which is shown in figure 3 already brings the piece of information which says that activities I.2, I.4 and I.6 are carried out by the same mechanism, however, such mechanism has not been determined yet. A rotating table was later chosen. The tool is able, however, to store the same functional module, once functions I.2, I.4 and I.6 are carried out by different mechanisms, which results in models of different instances and specifications, if compared to those shown in figure 4. Being so, the designer will have flexibility in the functional analysis of the automatic system to be developed.

5 Conclusion

The present work aims to present a computational model to help the conceptual design of automatic systems. This model is the first step for the conception of a computational tool. In opposition to the sequential engineering, where plant conceptions and control system are considered separately, the support computational model to the design aims to consider the aspects that are related to the different technologies or areas, so that the design process is more efficient and reliable.

The authors still have to face some important challenges. Initially, despite local modular approach [8] brings computational economy, the high complexity of the modularity test, is still a high cost aspect of the synthesis, and can be an obstacle in the solution of larger scale problems. A better efficiency in the modularity verification process is to be achieved. Another point that has been studied is the construction of controllers from initial levels of the functional structures. In this case, it covers specifications that handle more abstract information (the closer the functional structure is from the total function).

References

[1] Pahl, G., Beitz, W., Engineering design – a systematic approach, Springer – Verlag, UK, 1988.

[2] Hubka, V., Eder, W. E., Theory of technical systems, Springer-Verlag, Germany, 1988.

[3] Svendsen, K.-H., Hansen, C. T., Decomposition of mechanical systems and breakdown of specifications, Proc. ICED'93, The Hague, 1993, pp. 119-126.

[4] Malmqvist, J., Computer-aided conceptual design of energy-transforming technical systems, Proc. ICED'93, The Hague, 1993.

[5] Popovic, D., Vlacic, L., Mechatronic in Engineering Design and Product Development, Marcel Dekker, Inc. New York, USA, 1999.

[6] De Negri, V. J., Model structuring of automatic systems and its application for a test bench of hydraulic systems, Doctorate thesis, Federal University of Santa Catarina, Brazil, 1996. (in Portuguese).

[7] Ramadge, P. J., Wonham, W. M., The control of discrete event systems, Proc. of the IEEE, 77 (1): 81-98, 1989.

[8] De Queiroz, M. H., Cury, J. E. R., Modular Control of Composed Systems, Proc. of the American Control Conference, Chicago, June 2000.

[9] Umeda, Y., Tomiyama, T., Functional Reasoning in Design, IEEE Expert, Intelligent Systems and Their Applications, 12 (2): 42-8, 1997.

[10] Buur, J., A theoretical approach to mechatronics design, Denmark: Institute for Engineering Design, 1990.

[11] Hanisch, H. -M., Petri-Netze in der Verfahrenstechnik, Oldenburg, München, 1992.

[12] Shakeri, A., A methodology for development of mechatronic systems, Phd Thesis, Norwegian University of Science and Technology, Norway, 1998.

[13] Carrol, J., Long, D., Theory of Finite Automata, Prentice-Hall International Editions, Englewood Cliffs, USA, 1989.

[14] International Eletrotechnical Comission, IEC 848, Preparation of function charts for control system, Switzerland, 1988.

Eduardo Alves Portela Santos
Pontifical Catholic University of Paraná, ZIP CODE 80215901, Brazil, Phone:+55413301689, Fax:+55413301669, eaps@lcmi.usfc.br, http://www.las.pucpr.br;
Victor Juliano De Negri
Federal University of Santa Catarina, LASHIP/EMC/UFSC, P. O. BOX 476, Florianópolis, Brazil, Phone:+55483319396, victor@emc.usfc.br, http://www.laship.ufsc.br;
José Eduardo Ribeiro Cury
Federal University of Santa Catarina, CNPq 300953/93-3, LCMI/DAS/UFSC, P. O. BOX 476, Florianópolis, Brazil, Phone/Fax:+55483319934, cury@lcmi.ufsc.br , http://www.lcmi.ufsc.br.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

AN ENVIRONMENT DECOMPOSITION-BASED APPROACH TO DESIGN CONCEPT GENERATION

Y Zeng and P Gu

Keywords: process modeling, evolutionary design, human-product-environment system

1. Introduction

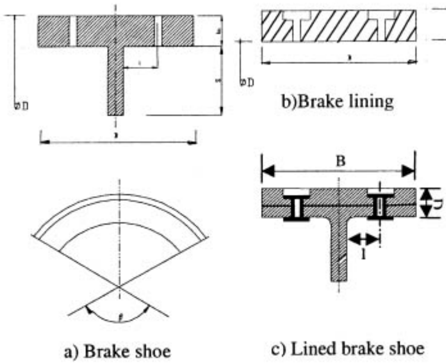
Conceptual design is one of the most critical design stages where some of the most important design decisions are made. Models and methods have been proposed to generate design concepts, based mostly on the systematic design approach. Conceptual design process is generally divided into the following steps: identify design specifications, establish function structures, search for solution principles for fulfiling the subfunctions, combine solution principles to fulfil the overall function, select suitable combinations to define design concepts, and evaluate concepts against technical and economic criteria[1]. Clearly, the definition of function and the establishment of function structure are fundamentaly important for the conceptual design process. However, one of the difficult aspects is that the generation of design concepts and the development of product descriptions are closely coupled. Establishing function structure is especially difficult for original design where no product structure exists. This makes conceptual design very challenging. Tomiyama and his co-workers [2] proposed to establish a function structure by capturing the design knowledge that transforms design requirements (in some cases, functions) into product descriptions or behaviors. However, definitions of product function and functional knowledge are subjective and domain-dependent if applying to a broader range of design problem solving. This paper proposes an environment decomposition based approach to the generation of design concepts from design specifications. This approach is based on a new definition of design requirements in the context of product environment, which is made possible through a function normalization process[3].

The following section provides a running example to illustrate the ideas presented in this paper. Section 3 decribes the design concept generation process based on environment decomposition. The concluding remarks are given in Section 4.

2. Running example

A rivet setting tool design is used as a running example to illustrate the ideas presented in this paper. This example was adopted from the book by Hubka et al[4]. Only the design concept generation process will be covered for the purpose of this paper.

The task is to design a tool for riveting brake linings onto brake shoes for internal drum brakes. Figure 1 gives the details and dimension of brake shoe, brake lining and rivets.



Data:

- Brake drum:
  inside diameter = 250~400mm
- Rivet to DIN standard 7338
  Shank diameter = 3~5mm
  Material: C_u
- Maximum shoe width
  B_{max} = 100mm
- Maximum total thickness of parts:
  D_{max} = 15mm
- minimum distance between rivet center line and rib
  L_{1min} = 100mm
- Riveting stroke: ~3mm
- Riveting force: ~500N

Figure 1 Form of brake shoe and lining[4]

The following is a list of design specifications for this example:

1) Functional requirements: riveting of brake lining to brake shoe.
2) Physical requirements: the form and dimensions of the tool must be consistent with the form and dimensions of brake shoe and brake lining given in Figure 1.
3) Ergonomic requirements:
   - User: car mechanic
   - Hand force: ~200N
   - Foot force: ~400N
   - Working height: 0.5~1.0m
   - Safety: against accidents
4) Operational requirements
   - Service life: 5 years
   - Good transportability
   - Maintenance free
5) Appearance requirements : no special requirements
6) Manufacturing requirements: manufacturable in workshop of ... Co. Ltd
7) Financial requirements: the maximum manufacturing costs ~ 190 Canadian Dollars.

Figure 2 shows two examples of the generated design concepts.

3. Environment decomposition based design concept generation

The objective of this paper is to establish a model of conceptual design process that starts from design specifications and ends with design concepts. Obviously, this process involves three elements: design specifications, product descriptions, and design knowledge. The details of product description modelling was discussed in other papers[5][6]. This section will include three subsections: design specifications, design knowledge, and the design process that transforms design specifications into design concepts using design knowledge.

3.1 Design specifications

An engineering system can be divided into two parts: product structure and its environment. For the example shown in Figure 2a), the product structure consists of all its components including spring, rack and pinion, stop, counter-weight, closure and perform heads. The environment could include natural environment such as gravity field, function environment such as the forces that the user may impose and the physical properties of brake lining and brake shoe, financial environment such as the price of each component, manufacturing environment such as available manufacturing tools, and so on. Theoretically speaking, everything else related to the product except the product itself can be seen as its environment. The interactions between the system structure and the environment are actions and responses. In the context of function environment for the running example, actions and responses can be force $\mathbf{F_h}$ from human hand, and the forces by closure and perform heads $\mathbf{F}_r^u$ and $\mathbf{F}_r^d$. They can be seen as the boundary between the structure and the environment. Another element of the boundary is the physical properties of the contacting enviornment components, which is directly connected to the product structure. For the same example, brake lining and brake shoe are part of environment components while the car mechanic is another. The car mechanic's hands can be seen as a contacting environment component. Graphically, an engineering system can be represented in Figure 3a).

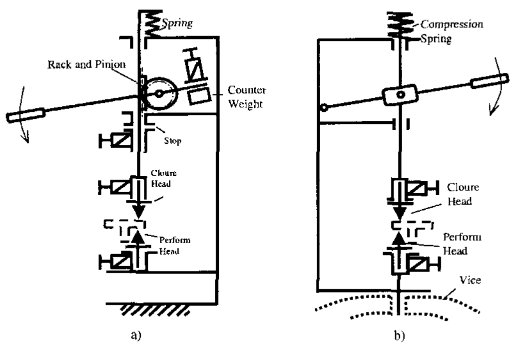


Figure 2 Example design concepts[4]

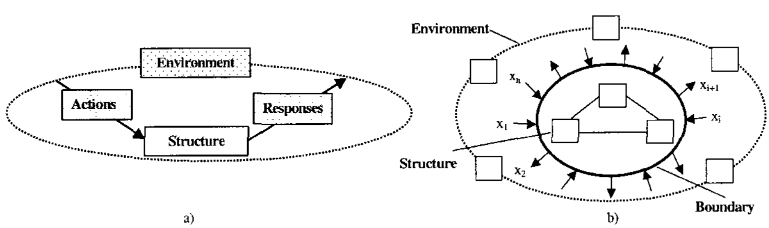


Figure 3 Engineering system

In engineering practice, only the boundary is important rather than the constituents of the enviornment. For instance, how the car mechanic looks is not the concern for this design problem, though it can be a part of environment. As a result, the environment can be represented by the boundary between product structure and product environment (It is shown in Figure 3b):

$$\mathbf{E} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_n\} \tag{1}$$

where **E** is environemnt and $\mathbf{x}_i$ can be either actions, responses, or product physical attributes.

It can be seen from Figure 3b) that only two possibilities exist to constrain an engineering system: the constraints on either product structure or product performances. Correspondingly, we have two types of design requirements: structural and performance, though there could be other design requirements in engineering applications, such as requirements for safety, for manufacturability and for serviceability. However, all these requirements can be modelled in these two forms through a normalization process[3]. Each design requirement can be mathematically represented as,

$$\mathbf{r}^d = \lambda(\mathbf{x}_i, [\mathbf{x}_i]) \tag{2}$$

where $[\mathbf{x}_i]$ is the constraint, quantitatively and/or qualitatively, on $\mathbf{x}_i$. λ is a logical operator such as =, ≥, ≤, and so forth. Then, design requirements for a design problem can be writtean as

$$\mathbf{R}^d = \{\mathbf{r}^d : \mathbf{r}^d = \lambda(\mathbf{x}_i, [\mathbf{x}_i]), \mathbf{x}_i \in \mathbf{E}\} \tag{3}$$

where $\mathbf{R}^d$ is a set of design requirements.

The design process is to evolve the set of design specifications and product descriptions until all design requirements are satisfied[7][8]. In the evolution process, the earlier environment and product description can be seen as the constraints and requirements on the later design. As a result, the design requirements in Equations (2) and (3) can be represented as a set of environment elements. $\mathbf{x}_i$ and $[\mathbf{x}_i]$ are just the elements of product environment in different satges of a design process.

For design specifications of the running example, the product-environment system should be firstly formulated as shown in

Figure 4. The environment can be written as:

$$\mathbf{E} = \{\mathbf{F}_f \cup \mathbf{F}_h, \mathbf{F}_r^u, \mathbf{F}_r^d, \mathbf{W}_b, \mathbf{G}_b, \mathbf{h}_w, \mathbf{x}_{mf}, \mathbf{x}_f, \mathbf{x}_s, \mathbf{x}_t, \mathbf{x}_{mt}\} \tag{4}$$

The symbols are self explanatory in the figure except that $\mathbf{G}_b$ is the geometric model of the brake lining and brake shoe. All design specifications can be represented in the following tables.

Table 1 Performance requirements

Type	Input	Output
Functional	$\mathbf{F}_h$ or $\mathbf{F}_f, \mathbf{F}_h \le 200\text{N}, \mathbf{F}_f \le 400\text{N}$	$\mathbf{F}_r^u, \mathbf{F}_r^d, [\mathbf{F}_{rmin}] \le \mathbf{F}_r^u, \mathbf{F}_r^d \le [\mathbf{F}_{rmax}]$
Financial	*Market information*	x_f, $x_f \le$CAN$190.0
Manufacturing	Manufacturing factors	Difference between design and product

Table 2 Structural requirements

Type	Environment	Product Descriptions
Physical	Geometric model of brake lining and brake shoe	Forms and dimensions of closure and perform heads
Ergonomic	Comfortable working height	h_w

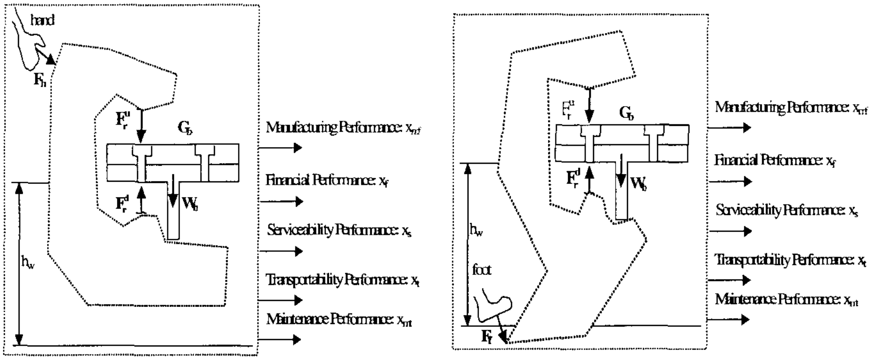


Figure 4 Initial environment for riveting tool design problem

3.2 Design knowledge

The basic design knowledge about a product is the knowledge about its causality. It adresses the causal relations between actions and responses with reference to the product. This can be simply represented as in Figure 5a) with a symbolic representation in Figure 5b).

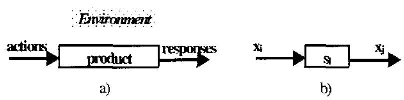


Figure 5 Basic form of product knowledge

The objective of most scientific explorations is to find the law governing the causal relation between actions and responses. Some of the laws are deterministic, such as Newton's law, the others are nondeterministic, such as chaotic dynamics. Figure 6 is an example regarding the relations of two forces, which can be used for the running example given before.

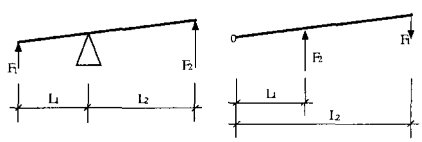


Figure 6 Example product knowledge

In this example, L_1 and L_2 are physical properties of structure while F_1 and F_2 are action and response, respectively. The knowledge can be written as

$$\forall \mathbf{lever}, \mathbf{F}_2 = \frac{\mathbf{L}_2}{\mathbf{L}_1}\mathbf{F}_1 \wedge \mathbf{F}_1 \rightarrow \mathbf{F}_2 \tag{5}$$

This relation from actions to responses is called product performance.

$$\forall \mathbf{S}\, \exists \mathbf{P} \subseteq \mathbf{A} \times \mathbf{R} \tag{6}$$

where **P** is a set of product performances, **A** is a set of actions, and **R** is a set of responses. In the above example, the performance can be named after 'transmit'. **A** and **R** are F_1 and F_2, respectively. Mathemmatically, it is described as

$$\forall \mathbf{lever}\ \exists \mathbf{transmit} \subseteq \mathbf{F_1} \times \mathbf{F_2} \qquad (7)$$

However, the name of the above relation is artificial and subjective. The word 'transit' can be replaced by 'change', 'increase to', and many others. As will be seen in the following discussion, only actions and responses as well as structure will be involved in solving design problems. This is different from function based approaches. The following is the only form of knowledge used in the design process model presented in this paper,

$$\forall \mathbf{s_l}\ \exists \mathbf{x_i} \exists \mathbf{x_j} \exists \mathbf{k_l^m}, \mathbf{k_l^m} : \mathbf{x_i} \rightarrow \mathbf{x_j} \qquad (8)$$

3.3 Environment decomposition based design concept generation

Zeng and Gu[7][8] formulated a design process model, mainly based on a dynamic model of product descriptions. There were several points that are not quite operational. In this section, a modified design process model is proposed based on the definition of design specifications and design knowledge described above. Figure 7 is the scheme of this model.

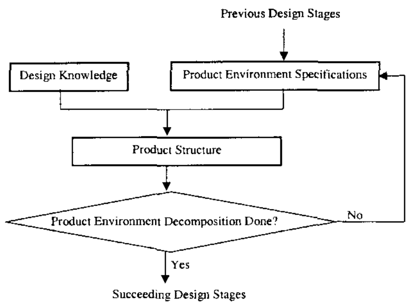


Figure 7 Environment decomposition based conceptual design process

This design process model can be described as follow:

1) Extract one environment element from the environment set;
2) If there is a piece of design knowledge mapping the extracted environment element to another action or response, then product structure s attached to this knowledge will be a component of design concepts. The extracted environment element is replaced by that mapped environment element;
3) Add component s to existing product structure **S**;
4) Detect the performance conflicts between the newly generated product component and existing product structure;
5) Form a new environment set;
6) If further environment decomposition cannot be done, then proceed to succeeding design stages, or else go to step 1.

Mathematically, the process can be formulated as

Design(R^d,S)

{

repeat

{

$\exists x_k \in E$; //decomposition of product environment

if $\exists s_l\, \exists k_l^m : x_k \rightarrow x_n$

if $\exists s_l\, \exists k_l^m : x_n \rightarrow x_k$ //application of design knowledge

$S = \xi(s_l, S)$; //combination of component into partial product

$E' = (E/\{x_k\}) \cup \{x_n\}$; //updating of product environment

$X_l = K_l(s_l)$;

$X_p = K_p(S)$; // properties of component and partial product

$E'' = X_l \updownarrow X_p$; //conflicts between partial and component

$E = E' \cup E''$; //updating of product environment

until no more environment decomposition can be done

}

This process can be described in Figure 8:

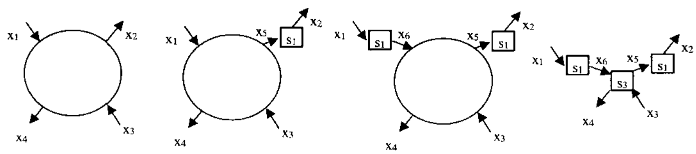


Figure 8 Graphic explanation of environment decomposition based conceptual design process

Figure 9 gives two intermideate steps of generating a design concept of riveting tool. In each step, there could be many alternatives. However, to save space, we only list one alternative.

For Figure 9a) the updated environment and product descriptions are:

$$E = \{F_h, F^u, F^d, W_b, G_h, x_{mf}, x_f, x_s, x_t, x_{mt}\} \qquad (9)$$
$$S = \{h_w, G_h\}$$

For Figure 9b) the updated environment and product descriptions are:

$$E = \{F_l, F^u, F^d, W_b, G_h, G_l, x_{mf}, x_f, x_s, x_t, x_{mt}\} \qquad (10)$$
$$S = \{h_w, G_h, G_l\}$$

It should be noted that only forces contributing to the function of components are given in Figure 9.

4. Concluding Remarks

This paper proposed a design process model based on the environment decomposition. It deals with problem decomposition in a way different from widely used function/task decomposition. A predefined function structure is not necessary for this model. Performance knowledge is the only knowledge needed for this process. The development of this model comes from defining the environment as the boundary between product and environment. A computer aided conceptual design software is under development based on this model.

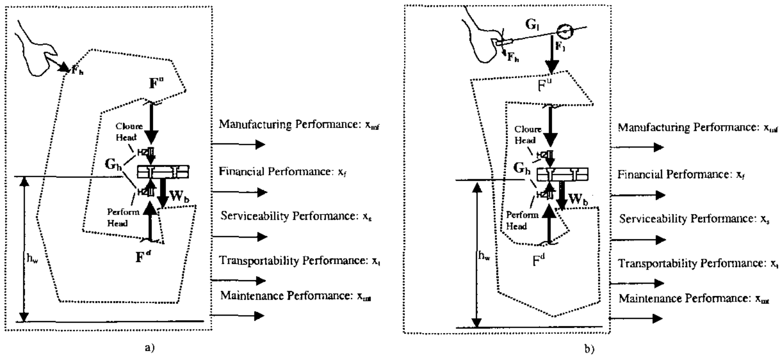


Figure 9 Two intermediate environment for the running example

References

[1] Pahl,G., Beitz,W.(1988), *Engineering design: a systematic approach*, Springer-Verlag, Berlin

[2] Lossack,R., Umeda,Y., Tomiyama,T.(1998), Requirement, function and physical principle modeling as the basis for a model of synthesis, Proc. of the 1998 Lancaster International Workshop on Engineering Design, Lancaster, UK, pp.287-298

[3] Zeng, Y.(2001), An Axiomatic Approach to Modeling of Product Design Concept Generation Process Using Set Theory, Ph.D. Thesis, The University of Calgary (in preparation)

[4] Hubka,V., Andreasen,M. and Eder,W.(1988), *Practical Studies in Systematic Design*, Butterworths

[5] Zeng, Y and Gu, P. (1999), A science-based approach to product design theory Part II: Formulation of design requirements and products, *Robotics and Computer-Integrated Manufacturing*, 15(4), pp.341-352

[6] Zeng, Y. and Gu, P. (1999), A set-theoretic model of product design requirements and products, *Proceedings of the 12th International Conference on Engineering Design*, pp.1117-1120

[7] Zeng, Y. and Gu, P. (1999), A science-based approach to product design theory Part I: Formulation and formalization of design process, *Robotics and Computer-Integrated Manufacturing*, Vol.15, No.4, pp.331-339

[8] Zeng, Y. and Gu, P. (1999), A set-theoretic model of design process, *Proceedings of the 12th International Conference on Engineering Design*, pp.1179-1182

Yong Zeng and PeihuaGu
Department of Mechanical and Manufacturing Engineering
The University of Calgary, Calgary, Alberta, Canada T2N 1N4
Fax:+1-403-282-8406, E-Mail: (yzeng, gu)@enme.ucalgary.ca

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

PRODUCT SHAPE IMAGE CREATION SYSTEM FOR INDUSTRIAL DESIGN USING EMERGENT PROCESS

K Sakita and M Igoshi

Keywords: Industrial design, aesthetics, process modelling, image, emergent process

1 Introduction

Image effects play important roles in human creative thinking processes, such as designing. A core part of the industrial design process is creating image structure of a product shape, which corresponds to defining a product's functional structure in engineering design. Principles of composition (the ordering and construction of form, color and texture) are then selected, combined and applied to the product shape.
We first proposed an image movement model (IMM), which represents the designer's image world and the industrial design process using the concept of image effects [8]. We proposed an extended image movement model (EIMM), which represents the image creation process of the product shape using chaotic dynamics [9]. In EIMM, image and image structure are represented by computer holograms. Holograms which represent mages are selected and merged into a single hologram which represents an image structure based on multiple correlation coefficients of computer holograms. An image structure is coded and structured by wave length of reference light of computer hologram. The hologram is then resolved into multiple layers. Chaotic dynamics is then applied to a selected range of spatial frequencies of layers which changes the correlation coefficients among the layers of the hologram. The changed layers are then merged into a new single hologram. It leads to structural change and growth of an image structure, and then product shape image is changed. This model represents creating process of a novel image of the product shape in industrial design.

2 The product shape creation process in industrial design

2.1 The industrial design process

In industrial design, product shape image creation process is composed of three subprocesses. The first process is image structure creation, which entails modifying and merging the motif of the product shape and its subimages. This process is similar to the functional structure creation process in engineering design [1][2]. The second process is principles of composition selection. The principles of composition are selected according to the image structure. This process is similar to the functions and parts selection process in engineering design. The third process is the creation, selection, and definition of form, layout, color, and surface. These are created and selected according to the image structure and selected principles of composition. This process is similar to the embodiment design process in engineering design. In engineering design, functional structure (functional network) is an important result of conceptual design. Corresponding by image structure (structure of meaning) is an important result of early stages of industrial design.

2.2 Image structure creation

The industrial designer selects the motif of the product shape from other artifacts, animals, nature, and so on according to the purpose of the product. This motif is the main product shape image. The industrial designer then selects subimages from other artifacts, animals, nature, and so on to detail the product shape. The subimages are influenced by modified according to, connected to, and merged with the motif of the product shape. The image structure is composed of the motif (the main product shape image) and the modified subimages. In the creative process, industrial designers create novel images by finding meaningful long distance relationship between the image in current field or category and the image in other fields or categories. This long distance relationship of image is called jump of image. An indusrial designer finds this meaningful relationship naturally and automatically after a long trial-and-error process of searching, creating, and modifying images. This is almost a subconscious process. In the creative industrial design process, the motif of product shape is derived from this long distance relationship of images. And the motif selects and modifies subimages and creates the image structure. It is clear that the selection and creation of the motif is important for the creation of the image structure and the direction of product shape making. This process is called the process of analogy [4]. And the image structure is called a schema [5][6].

2.3 Principles of composition selection

Principles of composition are the order and construction of form, color, and texture [7]. Principles of composition include proportion, balance, and rhythm of form, color, and texture. An industrial designer selects and combines the principles of composition and applies them to the industrial design of the product shape according to his or her image structure.

2.4 Elements of product shape creation and selection

From the point of view of industrial design, elements of the product shape are form, color, and texture [7]. An industrial designer selects, modifies, and combines form, color, and texture to create a product shape according to the principles of composition that he or she selected and combined plus his or her image structure.

3 The extended image movement model

3.1 The formulation of industrial design process

Figure 1 illustrates the concept of the extended Image movement model (EIMM). Design Aspect (DA) represents the collection of aspects of use that must be considered when the product is designed. Functional Requirements filter (FR) represents the product's functional specifications. Design Constraints filter (DC) describes the design constraints. Image filter (IM) reflects the designer's image structure. Design Parameters (DP) indicate the attributes of the product's shape. The Design Aspects are filtered by the Functional Requirements filter, the Design Constraints filter and the Image filter into the Design Parameters.

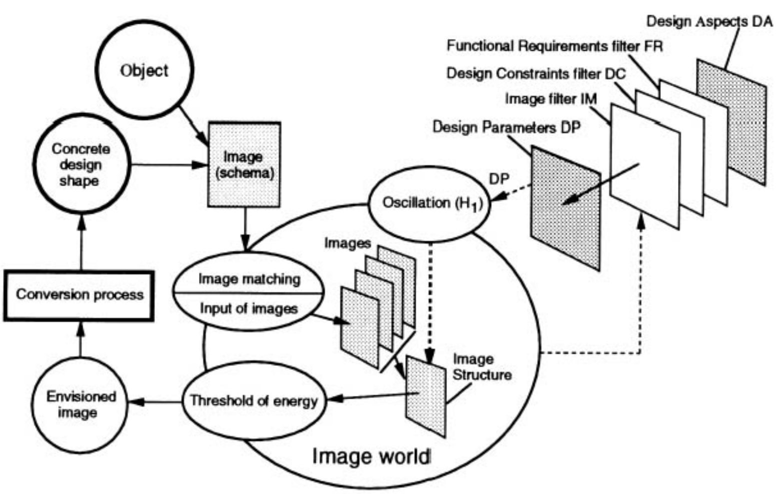


Figure 1. The extended image movement model

The image world is a set of image structures which is represented by a set of computer holograms. Images of objects and the designing product shape in the environment are captured as computer holograms. And these images are cumulated in the image world as computer holograms. The tendency of the image world reflects to image filter. Design parameters affect image world. If energy of an image in the image world exceeds a threshold, the image is envisioned. A concrete design shape is made from the envisioned image through a conversion process. The concrete design shape is then recaptured into the image world as a computer hologram.

In this way, in industrial design this process is driven cyclically. Because the design parameters are just a set of attributes [3], it is not possible to make a product shape image completely from design parameters. Images are needed to create product shape image completely.

3.2 Shape model using computer hologram

Ordinary shape models are not suitable to represent shapes when chaotic dynamics is applied. This is because the resultant shape becomes broken down in form and confused in data structure. As computer holograms are composed of spatial frequencies, their data structure and operation are simple. Therefore we proposed and developed a shape model using computer holograms (Fresnel type and Fourier transform type) for the representation of image and image structure in EIMM [10]. This shape model represents shapes by potential values of point charge models that are located in three-dimensional space. The point charge models in space are recorded into a two-dimensional computer hologram. Locations, strengths, and reducing factors of the point charge models in space are reconstructed from the computer hologram. Finally, shape, which is represented by potential values of reconstructed point charge models, is reconstructed. Locations of point charge models represent a skeleton of the shape. This shape model is able to carry out set operations, local operations, and three-dimensional shape feature detection and matching.

3.3 Representation of an image structure

Objects in the environment and product shapes are recorded as shape models using computer holograms. Similarly, digital pictures of objects and scenes are recorded in computer holograms as images. It is possible to derive skeletons of objects, shapes and pictures from the computer holograms. It is easy to calculate correlation coefficients between objects and product shapes that have been recorded in computer holograms in order to perform three-dimensional shape matching and shape feature recognition. Because it is possible to merge many computer holograms into one, to reconstruct objects respectively, and to reconstruct a merged object, it is also easy to make a merged image. Coding of images that are represented by computer holograms is carried out using amplitude, phases and wave length of reference light of computer hologram. From these characteristics of a computer hologram, an image structure that was initially composed of many computer holograms is represented by one. The image world is composed of many computer holograms that represent image structures.

3.4 Construction of an image structure

An image structure is constructed by matching, coding, and merging of computer holograms. Three-dimensional shape matching between the selected main image (motif) and relevant image ,which are represented by computer holograms, is carried out globally and locally. An image that has high correlation coefficients with the selected main image both globally and locally becomes a candidate element (subimage) of the image structure. The subimage is coded and structured using wave length, amplitude, and phase of reference light. The coded subimage is merged into the selected main image. Finally, the selected main image and subimages are coded, structured, and merged into one computer hologram. The computer hologram represents an image structure. An image structure is reconstructed using several wave lengths of reconstruction lights of the computer hologram. This is envisionning process of an image structure.

4 The creation of a novel image using emergent process

4.1 Image structure and design parameters

Design parameters define only the framework of the product shape image. An image structure of the product shape is necessary to create a unique final product shape image. An image structure is structured and layered using wave length of a reference light of a computer hologram. Each layer (also a computer hologram) of the image structure and relevant wave length of the reference light corresponds to some design parameters.

4.2 Chaotic dynamics as an emergent process

Chaotic dynamics generates an emergent process. Poincare catastrophe is a kind of chaos. Poincare catastrophe occurs in the system of billiards. In formula (1), H denotes the Hamiltonian of this system. H_0 denotes the non-oscillation Hamiltonian, H_1 denotes the oscillation Hamiltonian, and ε denotes the connecting parameter.

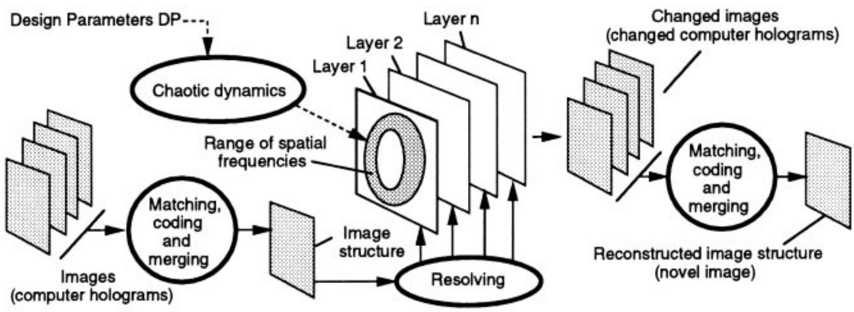


Figure 2. Creation of a novel image

$$H = H_0 + \varepsilon H_1 \tag{1}$$

Oscillation Hamiltonian, shown in formula (2), is introduced into this system, where j_i is an integer, θ_i is an angular frequency, t is time, and A and B are constants.

$$H_1 = A\cos\left(\sum_{i=1}^{n} j_i\theta_i t\right) + B\sin\left(\sum_{i=1}^{n} j_i\theta_i t\right) \tag{2}$$

Poincare resonance occurs in this system, and Poincare resonance leads to chaos. Different patterns of oscillation in the system lead to different patterns of chaotic orbit of the phase space. Different patterns of oscillation in the system are made by different combinations of integer values j_i, angular frequencies θ_i, and time t.

Each layer (computer hologram) of the image structure corresponds to some relevant design parameters and also corresponds to a value of the integer j_i and angular frequency θ_i in formula (2).

4.3 Creation of a novel image

As shown in figure 2, in order to apply chaotic dynamics to an image structure, the image structure is first resolved into layers (computer holograms). Chaotic dynamics is then applied to a certain range of spatial frequencies of each computer hologram. Chaotic dynamics of billiard system acts on each dot of the computer hologram. As a result of this process, the correlation coefficients between the computer holograms of a subimage and main image change. Also, some patrs of computer hologram that correspond to some of the design parameters change. Next, an image structure is reconstructed from the changed computer holograms. The application of chaotic dynamics results in a different image structure. The new image structure creates new product shape image. There is therefore the possibility of creating significantly novel images from the new image sructure. Chaotic dynamics is the driving force of growth of image structure in EIMM.

In the early stages of industrial design, the industrial designer makes many sketches on papers or computer display in order to develop his or her ideals of product shapes. If an industrial designer would make sketches on computer display by shape model using potential values and computer holograms, the shape data could be used directly to create product shape images. Also before the industrial designer draws sketches of product shapes, he or she carries out market research in order to capture information and images about the product, other relevant products, users, and so on. During this process, the industrial designer gathers many two-dimensional digital images about the product. Computer holograms can be made from two-dimensional digital images easily. Two-dimensional digital images are usable in EIMM supplementalily. The EIMM generates the possible ideas of the product shape from the information that is gathered and drawn by the industrial designer. The possible ideas of product shape that are generated by EIMM support and activate product shape image creation process of the industrial designer.

5 Conclusion

1. This paper presents a new industrial design process model called the extended image movement model (EIMM).
2. This paper formulates the creative process of industrial design in terms of image (schema).
3. This paper indicates that the process of creating the image structure of the product shape image is a core part of industrial design.
4. Shape model using computer holograms is applied to representing image and image structure.
5. This paper presents a method of creating and developing image structure using matching, coding, and merging of computer holograms.
6. This paper presents a method of creating novel images of a product's shape by applying chaotic dynamics to an image structure, whitch is represented by a computer hologram. And the method is implemented in computer.

References

[1] Hubka, V. & Eder, W. E., "Design Science", Springer-Verlag, 1996.

[2] Pahl, G. & Beitz, W., "Engineering Design A Systematic Approach", The Design Council, 1988.

[3] Suh, N. P., "The Principles of Design", Oxford University Press, 1990.

[4] Rowe, P. G., "Design Thinking", MIT Press, 1987.

[5] Fujioka, Y., "Images and Human Beings", Japan Broadcasting Corporation Publishing, Tokyo, 1974.

[6] Neisser, U., "Cognition and Reality - Principles and Implications of Cognitive Psychology", W. H. Freeman and Company, San Francisco, 1976.

[7] Dondis, D. A.,"A Primer of Visual Literacy", MIT Press, 1973.

[8] Sakita, K. & Igoshi, M.,"A product shape generator using an image movement model for industrial design", Proceedings of ICED97, Tampere University of Technology, Finland, August 1997, pp. 681-686.

[9] Sakita, K. & Igoshi, M., "Product shape creation process of industrial design expressed by extended image movement model", Proceedings of DESIGN 2000, Dubrovnik, Croatia, May 2000, pp. 147-150.

[10] Sakita, K., Takada, K., Dantas, D. O. & Igoshi, M., "Shape model for product development system of industrial design using computer holograms", VDI Report 1569 - Product Development in the Real World, VDI-EKV, pp. 213-219, November 9-10, 2000.

[11] Shimojyo, T., "Introduction to Chaotic Dynamics", Modern Science Company Ltd., Tokyo, 1998.

Kazuhiro Sakita, M. Art & Design, ph.D.
Soka University, Department of Information Systems Science
1-236, Tangi-cho,
Hachioji-shi, Japan
phone : +81(0)426 91 9415
fax : +81(0)426 91 9312
e-mail : sakita@t.soka.ac.jp

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

MODELLING AND EVALUATION OF PRINCIPLE SOLUTIONS OF MECHATRONIC SYSTEMS, EXEMPLIFIED BY TYRE PRESSURE CONTROL IN AUTOMOTIVE SYSTEMS

J Gausemeier, M Flath, and S Möhringer

Keywords: Functional modelling, modelling of principle solutions, semi-formal modelling, design reviews, mechatronics.

1 Introduction

Product properties of mechatronic systems can be fulfilled by working principles and solution elements from different engineering domains. This demands detailed knowledge about the system elements and their physical and logical interdependencies. The interdependencies need to be modelled, coordinated between the engineering domains involved and simulated in a cross-domain way. In industrial practice domain-specific thinking and methods are often predominant. Cross-domain specification methods taking the possibilities of different disciplines early and integratively into account are missing.

Furthermore alternative solutions should be evaluated in the early stage of conceptualisation; better design quality can be achieved and time and cost intensive loops can be avoided. Because of the complexity of mechatronic solutions and the close interactions however, an early evaluation seems to be difficult: The anticipation of future product properties is afflicted with uncertainty [1]. Effects of product properties on the fulfillment of objectives are often ambiguous [2]. In this regard two important challenges for the conceptual design of mechatronic products can be characterized: Experts from different disciplines need a common method to specify the results of product conceptualisation in a cross-domain way. It would generate the base for equal cooperation and communication and would make sure, that the product concept benefits from the close interaction of the engineering domains. Especially synergetic solutions from different domains lead to new products. In order to evaluate alternative solutions, a systematic approach is necessary starting from a consistent system of objectives until an integrated evaluation method for alternatives.

2 Reference Model for the development of mechatronic systems

The product development process ranges in the broadest sense from the product idea until a successful market entry. When developing mechatronic systems the integrative modelling and simulation of product properties from different engineering disciplines is a key task. The V-Model, known particularly from software development, has been adapted to support this proceeding [3]. It serves as a reference model for product development processes in mechatronics in order to represent the virtual product properties in comparison with the reality (Figure 1).

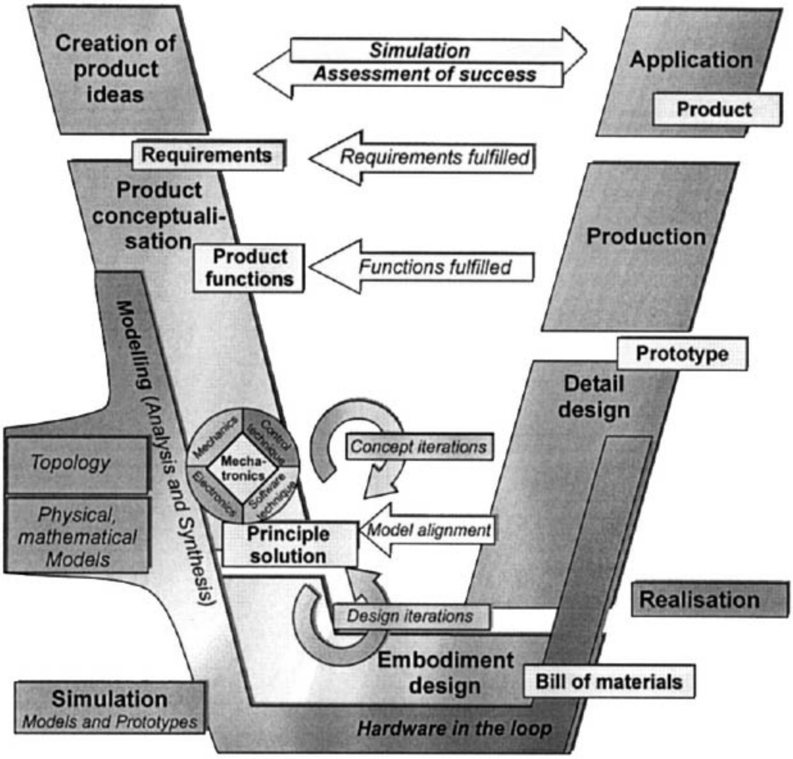


Figure 1. Reference model for the development of mechatronic systems, based on the V-Model

The left-hand branch of the V-Model describes the stages *creation of product ideas, product conceptualisatio*n and *embodiment design.* During these stages specifications for the future product based on simulation models and physical prototypes are determined. Product functions, behavior and shape need to be regarded in an integrative way; parameters like inertia or electrical capacity depend on the system geometry and can only be optimized in common. Via design iterations the virtual components will be tested successively in combination with real components. In the following the transfer to the stages *detail design*, *production* and *application* on the right-hand branch is made. Only when the product is used by the customer a complete evaluation of the specifications and the simulation results on the left-hand branch can be realized [4].

The principle solution is the result of product conceptualisation and represents an important milestone in the V-Model: It is considered as the coarse, but fundamental determination of the physical and logical mode of action of the future product. A common and equal elaboration by experts from the involved engineering disciplines is the decisive factor for quality and excellence of the principle solution. In order to ensure the equality of the engineering disciplines cross-domain modelling languages are needed; they help to understand problems and difficulties of the respective domains and to consider their possibilities and solutions in an integrative way.

3 Modelling of principle solutions of mechatronic systems

In the following a graphical method for the integrative and cross-domain modelling of principle solutions is presented [5], [6]. It is based on constructs for the modelling of the system structure and domain-own specifications of behavior and shape for the system elements. System elements are working principles and solution elements which are graphically represented and characterized be the name and the corresponding engineering domain. Working principles are represented by ovals whereas solution elements are represented by hexagons (Figure 2).

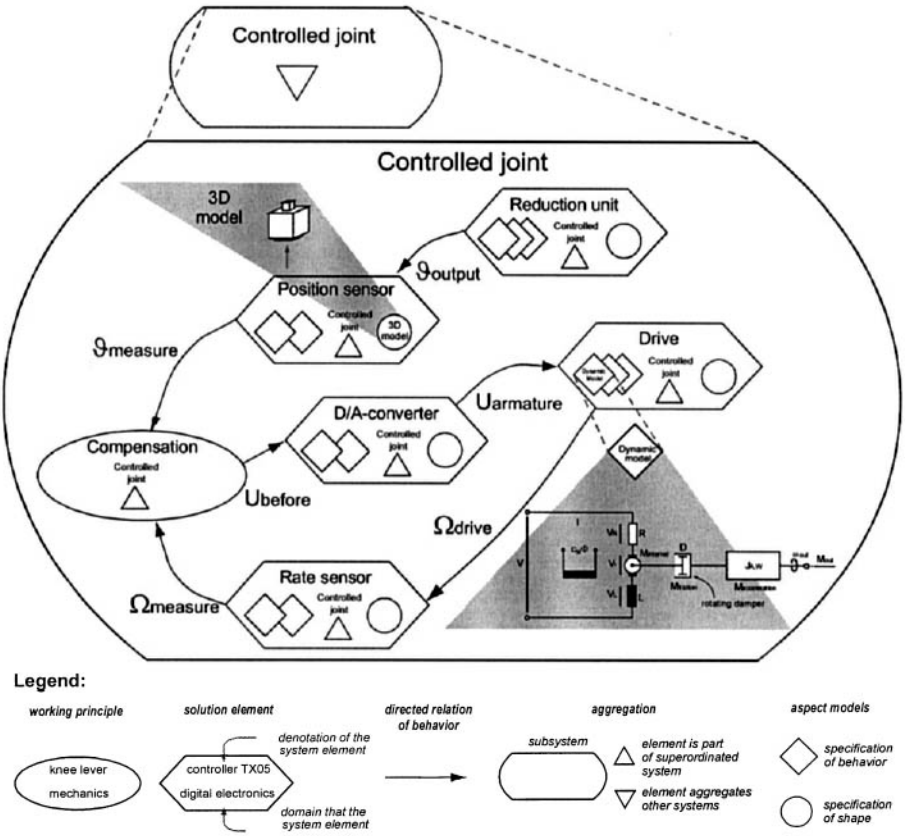


Figure 2. Example for the modelling of principle solutions of mechatronic systems – hierarchical network of working principles, solution elements and their relations

Describing the relations between system elements the principle solution can be modelled as an entire system. Relations of behavior show the cooperation between elements. Relations for boundary conditions describe non-functional dependencies as e.g. the determination of the hardware for working principles of the domain software. System elements can further be aggregated because of logical or spatial aspects. A logical aggregation indicates function units. For this purpose sub-systems are modelled. The spatial hierarchical modelling may be deployed if the spatial design governs the product design. Within mechatronic product design the spatial design is mostly less important than the functional aspects. Aggregated elements are grouped together in a sub-system (Controlled joint in Figure 2). The grouped elements are indicated by an arrow pointing upwards. Additionally the name of the sub-system is noted. The sub-system contains an arrow pointing downwards to indicate further elements on a more detailed level. The hierarchical modelling permits the aggregation over several levels. The upper levels hide detailed information. The hierarchical order offers a clear representation model of the principle solution on the upper levels. If needed the detailed levels can be inspected to obtain more information.

Behavior and shape of the system elements are specified by the proper models which are known and established within the different engineering domains. While for example principle sketches, symbolic-physical descriptions and 3D-CAD-models are used for the mechanical domain, kinematic and dynamic models are applied within the control domain. In Figure 2 for example the shape of the position sensor is specified by a 3D-CAD-model, the behavior of the drive by a dynamic model. It is not aim of this graphical method to introduce an "esperanto" for the specification of behavior and shape. Therefore the established domain-specific languages are used and integrated. This ensures as well a smooth transfer to the embodiment design where the product concept will be developed further within the specific domains.

4 Evaluation of alternative principle solutions

The development of mechatronic products requires a reliable evaluation of alternative concepts during the early design stages. This becomes especially important since the incorporated engineering disciplines of mechatronics can offer totally different solutions for the same function. An actuator may be built on a hydraulic or electromechanical working principle. Different control strategies with software and electronic components alter this complexity.

The value-oriented assessment analysis is a well known method for decision support within the product development [7]. This method requires a system of independent objectives that covers the critical requirements and boundary conditions responsible for the success or the failure of the product. This system can be built of a hierarchy of objectives as cost, weight and functional aspects to be fulfilled. To every objective a weight is assigned that expresses its importance to the overall system of objectives. The drawback of this method is the explicit independence of the objectives and the subjective manner setting up the weights of the objectives.

Saaty invented the Analytic Hierarchy Process (AHP) as a more detailed method to assess different solutions within a system of objectives [8]. By means of mathematical matrix operations first the system of objectives is ranked. The rank expresses the order of the importance of the objectives. Within the next step the fulfillment of each objective through the principle solutions is taken into account. These values of fulfillment are incorporated with the rank order of the objectives to obtain an overall value of the principle solutions. The AHP offers a more accurate way to assess different solutions with respect to a system of objectives. Furthermore the consistency of the system of objectives can be verified. Formerly the application of this method was only described within the context of risk management of socio-economic systems or large environmental projects. Since the method seems perfectly suited for the application within the conceptual design of products it is presented in the following example as an application for the product development of complex mechatronic products.

5 Evaluation example: tyre pressure control in automotive systems

According to a breakdown statistics of BMW 15% of tyre breakdowns are caused by sudden fail of pressure, while 85% of tyre breakdowns are lead back to creeping or slow fail of pressure. In Europe and in the USA every 75.000 kilometres such a tyre breakdown occurs. Therefore, tire and car manufacturers are interested in solutions which signal a tyre pressure air loss to the driver during the ride. Fail of pressure warning systems can be realised in different ways [9]:

- **Direct measuring systems:** a sensor inside the wheel or on the rim measures tyre pressure and / or temperature and transfers the data telemetrically to an evaluation module.

- **Indirect measuring systems:** Dummies such as natural frequency or rotational speed of the wheels are used to determine the tyre pressure. The evaluation accuracy is smaller in comparison with the direct measuring systems, but the hardware costs are far below.

- **Furthermore, the control system can be realised in different versions:** as an independent control module, as a control module connected via a BUS system, as a multifunctional control unity or an integrated system (integrated in an existing control module, e.g. ABS).

Different from the value-oriented-assessment-analysis the AHP first weight the objectives against each other. For this purpose functional objectives, malfunctions and boundary conditions expressing requirements are set within a matrix (Figure 3). Then each objective is compared with each other objective. A number in the matrix expresses the relative importance between two objectives (e.g. sensitivity to detect a pressure loss - recognition of gas diffusion of the tyres). The sensitivity is found to be essential more important than the recognition of gas diffusion. Therefore the element is filled in with a five.

		No.	1	2	3	4	5	6	7	8	eigen-vector
functional objectives	sensitivity	1	1	5	1/9	1/3	7	1/7	5	1	0,111
	gas diffusion	2	1/5	1	1/9	1/5	3	1/3	5	1/3	0,058
malfunctions	false warning	3	9	9	1	1	7	1	5	7	0,292
	system failure	4	3	5	1	1	7	1	3	3	0,182
	tyre wear	5	1/7	1/3	1/7	1/7	1	1/5	1	7	0,061
boundary conditions	small number of parts	6	7	3	1	1	5	1	3	1	0,189
	development costs	7	1/5	1/5	1/5	1/3	1	1/3	1	3	0,046
	designed for upgrade	8	1	3	1/7	1/3	1/7	1	1	1	0,062
										λ_{max}	11,165
										C.I.	0,452
										C.R.	0,321

Size of matrix	1	2	3	4	5	6	7	8	9
Random consistency	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45

Intesity of relative importance

Equal importance	1	
Moderate importance of one over another	3	
Essential or strong importance	5	
Demonstrated importance	7	
Extreme importance	9	
Intermediate values between the two adjacent judgements	2	4,6,8

Legend
eigenvector: highest value corresponds with most important objective
C.I.: Consistency index
C.R.: Consistency ratio
λ_{max}: highest eigenvalue of the matrix

Figure 3. Weighting of the objectives by bilateral evaluation and consistency check of the system of objectives

Within the matrix the reciprocal relation between the objectives is entered automatically. Therefore the corresponding cell between the detection of gas diffusion and the sensitivity is filled in with 1/5 which holds the meaning of essential less important. The diagonal elements obtain the number 1. The intensity of relative importance reflects the scale of the used values within the matrix elements. Through this scale and the reciprocal relation between two elements a special matrix is obtained namely a positive reciprocal matrix. This type of matrix has two meaningful characteristics [8]: The elements of the normalised principal eigenvector corresponds to the weights of the objectives. Calculating (λ_{max} - n)/(n-1) the maximum eigenvalue represents an index for the consistency of the assessment. The calculated value is named Consistency Index (C.I.): An inconsistency may occur if the objectives are miss-valued. Therefore an index measuring the consistency is very helpful to check the utility of the matrix. Dividing the C.I. through the median C.I. of randomly assessed matrices a normalized index called Consistency Ratio (C.R.) is obtained. The C.R. should be less than 20% to designate an acceptable consistent matrix.

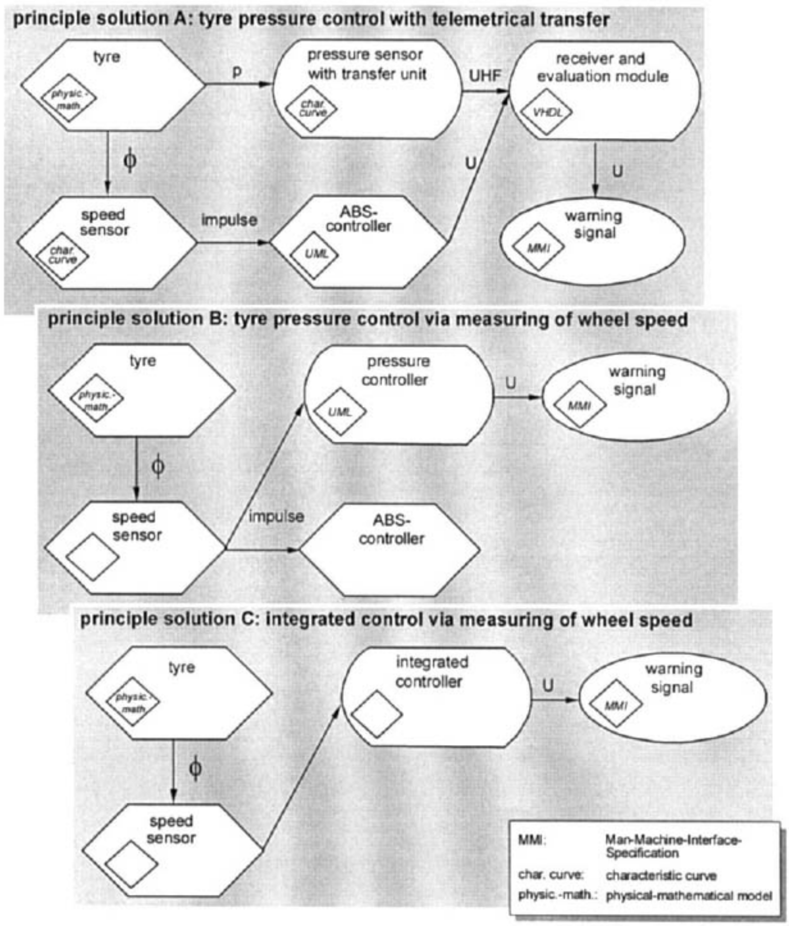


Figure 4. Modelling of three alternative principle solutions for tyre pressure warn systems

Within the next step the alternative principle solutions are modelled (Figure 4). Physical relations can be described and considered later for the evaluation.

For every objective a new matrix is built where the principle solutions are valued for the fulfillment of this particular objective (Figure 5). Applying the same procedure as described above the best solution is obtained by calculating eigenvectors and eigenvalues. The consistency can be checked in the same manner.

sensitivity

	PS A	PS B	PS C	EV
principle solution A	1	9	9	0,818
principle solution B	1/9	1	1	0,091
principle solution C	1/9	1	1	0,091
	1,222	11,000	11,000	
			C.I.	0,000
			C.R.	0,000

gas diffusion

	PS A	PS B	PS C	EV
principle solution A	1	9	9	0,818
principle solution B	1/9	1	1	0,091
principle solution C	1/9	1	1	0,091
	1,222	11,000	11,000	
			C.I.	0,000
			C.R.	0,000

false warning

	PS A	PS B	PS C	EV
principle solution A	1	3	3	0,584
principle solution B	1/3	1	1/3	0,135
principle solution C	1/3	3	1	0,281
	1,667	7,000	4,333	
			C.I.	0,068
			C.R.	0,117

system failure

	PS A	PS B	PS C	EV
principle solution A	1	3	3	0,567
principle solution B	1/3	1	1/5	0,110
principle solution C	1/3	5	1	0,323
	1,667	9,000	4,200	
			C.I.	0,147
			C.R.	0,254

wear of tyre

	PS A	PS B	PS C	EV
principle solution A	1	3	3	0,600
principle solution B	1/3	1	1	0,200
principle solution C	1/3	1	1	0,200
	1,667	5,000	5,000	
			C.I.	0,000
			C.R.	0,000

small number of parts

	PS A	PS B	PS C	EV
principle solution A	1	1/9	1/9	0,044
principle solution B	9	1	1/7	0,205
principle solution C	9	7	1	0,751
	19,000	8,111	1,254	
			C.I.	0,218
			C.R.	0,376

Figure 5. Evaluation of the alternative principle solutions considering the particular objectives

As the last step the fulfillment of the particular objectives can be aggregated to an overall value; it describes the utility of a principle solution based on the complete system of objectives (Figure 6).

	O 1	O 2	O 3	O 4	O 5	O 6	O 7	O 8	overall-value
P of O	0,111	0,058	0,292	0,182	0,061	0,189	0,046	0,062	
PS A	0,818	0,818	0,584	0,567	0,600	0,044	0,048	0,058	0,462
PS B	0,091	0,091	0,135	0,110	0,200	0,205	0,761	0,735	0,206
PS C	0,091	0,091	0,281	0,323	0,200	0,751	0,191	0,207	0,332

PS: principle solution
O: objective
P: priority as components of eigenvector

Figure 6. Aggregation of the particular priorities to an overall value for every principle solution

The application of this procedure ensures an integrated evaluation incorporating the complex dependencies of the system of objectives. Therefore this method supports perfectly the well known dilemma of valuing product concepts within the product development.

In contrast to the value-oriented-assessment-method that may considered as state-of-the-art in the conceptual evaluation [7] the incorporation of the AHP offers a remarkable benefit. The objectives are ordered by means of the ranking in pairs, the calculation of the eigenvector and the determination of the consistency ratio; the different importance of the objectives is specified consistently. The ranking of the fulfilment of the particular objectives by the alternative solutions is determined by the same functionality. The eigenvalue of the assessment matrices resp. the consistency ratio indicates the correctness of the ranking in pairs. A defective assessment may caused by a vicious circle. The solution A is higher valued than B. If B is higher valued than C and C has a higher priority than A a vicious circle is constructed. The value-oriented-assessment –method does not provide a consistency check.

6 Conclusion

Integrated in a reference model for the development of mechatronic systems a graphical method for the cross-domain modelling of principle solutions was presented. The total function of the system is modelled by working principles and solution elements. The system behavior is described by the established domain-specific methods. In this way a cross-domain specification of the principle solution is possible considering the possibilities of different engineering disciplines in an equal way. Furthermore a smooth transfer to the embodiment design is ensured.

The evaluation approach based on the Analytic Hierarchy Process of Saaty takes the consistency of the system of objectives and of the evaluation of alternative solutions into account. It ensures a better and securer evaluation during the early stage of product conceptualisation.

References

[1] McCormick, N.J.: Reliability and risk analysis. Academic Press, New York, 1981

[2] Suh, N.P.: The principles of design, Oxford Univ. Pr., New York et al., 1990

[3] Bröhl, A.-P. (Hrsg.): Das V-Modell – der Standard für die Softwareentwicklung. 2. Aufl., Oldenbourg, München, 1995

[4] Flath, M.; Kespohl, H.; Möhringer, S.; Oberschelp, O.: Entwicklung mechatronischer Systeme. In: Gausemeier, J./Lückel, J.: Entwicklungsumgebungen Mechatronik - Methoden und Werkzeuge zur Entwicklung mechatronischer Systeme. HNI-Verlagsschriftenreihe, Bd. 80, Paderborn, 2000

[5] Kallmeyer, F.; Gausemeier, J.: Modellierung prinzipieller Lösungen mechatronischer Systeme. In: Konstruktion. Jahrgang 51, Heft 11/12, 1999, S. 15-21

[6] Gausemeier, J.; Flath, M.; Riepe, B.: Strategic Planning and Conceptual Design of Mechatronic Products. In: Wallaschek, J. [eds]: Mechatronics and advanced motion control, proceedings of 3rd International Heinz Nixdorf Symposium, Paderborn, 1999

[7] Pahl, G.; Beitz, W.: Engineering Design, Springer, London, 1996

[8] Saaty, T.L.: The analytic hierarchy process: planning, priority setting, resource allocation. McGraw-Hill, New York et al., 1980

[9] Gumpp, A.; Schuchhardt, B.; : WARNAIR - a Tyre Deflation Warning System for PCR Tyres. In: VDI-Gesellschaft Fahrzeug- und Verkehrstechnik (Hrsg.): Reifen, Fahrwerk, Fahrbahn. Tagung Hannover, 21-22 Oktober 1999, VDI-Verlag, Düsseldorf, 1999

Prof. Dr.-Ing. Jürgen Gausemeier
Computer Integrated Manufacturing, Heinz Nixdorf Institut, Universität Paderborn, Fürstenallee 11, 33102 Paderborn, Tel.: +49-5251-606267, Fax: +49-5251-606268, e-mail: {gausemeier|flath|moehr}@hni.uni-paderborn.de, http://wwwhni.upb.de/rip/

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

USING THE SKELETON MODEL FOR PRELIMINARY GEOMETRICAL SYNTHESIS OF THREE-DIMENSIONAL CINEMATIC CHAINS

J-C Wahl, M Sartor, and J-C Fauroux

Keywords: Mechanism synthesis, computer aided design, geometric model, optimisation techniques.

1 Introduction

The first step in the design process is cinematic synthesis which aims to create a mechanism to yield a desired set of motion characteristics. A frequent design requirement is to cause an output member to rotate, oscillate, or reciprocate according to a specified function of time or of the input motion. This is called function generation. Moreover, if cinematic structures are thought of as ordered sets of constructive primitives, the resulting mechanisms can be called cinematic chains or SISO (single input single output mechanisms).

The literature provides several general computational methods for the synthesis of cinematic chains. For example, the abstract representation of geared cinematic structures was investigated with the aid of graph theory first by Buchsbaum and Freudenstein [1] and then by other authors [2-3]. More recently, researchers in computer science have proposed several methods in qualitative physics and constraint programming for the synthesis of this kind of mechanism [4-9].

Within the function generation field, the synthesis problem specification involves describing input and output motion, and a set of constraints, of which some are geometrical, e.g. the required positions and orientations of input and output members.

Most synthesis methods ignore geometrical constraints within their reasoning procedure, dealing only with structural and topological considerations. The approach suggested by Kota [6], which uses a matrix representation to model its building blocks, is one of the few which manage orientation constraints from the first synthesis process step. Note, however that it works only with the 3 orthogonal orientations (**X**,**Y**,**Z**) of the reference frame.

The first synthesis step must therefore be generally followed by the application of geometrical synthesis work to verify if the synthesized chain is able to fulfil the entire specification. For example, Chakrabarti and Bligh [8-9] start by producing a set of solution concepts for a given design problem, using kind synthesis procedures, and then continue by checking that a candidate solution concept satisfies the geometrical requirements, using a constraint propagation procedure. Working within orthogonal restrictions, they have shown that their approach makes it possible to process orientation and sense constraints; the management of position constraints is kept for the later and more detailed phases of design.

Our proposed method also consists in a multi-step solution to the cinematic chain synthesis problem:

- Step 1: from a data base including all the most common elementary mechanical blocks (EMB), a structural synthesis process produces all the global structures, as ordered sets of EMBs, which are likely to fulfil the requirements. It works in three phases: 1. Enumerating all the possible combinations, 2. Eliminating inappropriate solutions from a set of rules (some of which are geometrical, but only qualitatively), 3. Sorting the remaining solutions by order of decreasing interest. At the end of this first step, there is no guarantee that a suggested solution will be able to satisfy all the precise geometrical constraints of the specification.
- Step 2: a candidate structure having been selected, this preliminary task of geometrical synthesis is intended to find the associated 3D closed chain running from the input to the output position and respecting the main structural characteristics of each constitutive EMB. If it succeeds, an initial geometrical model of the structure is obtained.
- Step 3: this last task consists in a full synthesis which completes the one carried out previously by taking into account additional variables such as the main component dimensions (shafts and wheel diameters…) and considering conventional design criteria (contact stress, fatigue life, proportion ratios…).

This paper focuses on the second step. A model which is able to represent and position in space the main geometrical elements of any 3D speed reducer structure has already been presented in [10]. This model is based on the concept of mechanism skeleton. A CAD tool has been developed where about twenty EMBs are considered, such as different cylindrical gearings, crossed-axis helical gearing, worm gear... The main interest of this approach lies in its ability to manage accurate geometrical constraints. 3D problems with any input and output orientations may be tackled. Orientation and position requirements are considered together within an unique analytical formulation. We will show here that the skeleton technique can be extended to other types of 3D cinematic chains, especially those which also include more complex constructive primitives such as slider-crank, eccentric, rack-and-pinion, cam …

2 Problem setting

The starting point of the preliminary geometrical synthesis step is a given candidate structure, made up of a set of serially connected EMBs. The geometrical elements which constitute the specification sheet of the synthesis problem are: the parallelepipedic envelope inside which every part of the mechanism should be inscribed, the position (O_0) and orientation ($\mathbf{Z_0}$) of the input member and the position (O_S) and orientation ($\mathbf{Z_S}$) of the output member.

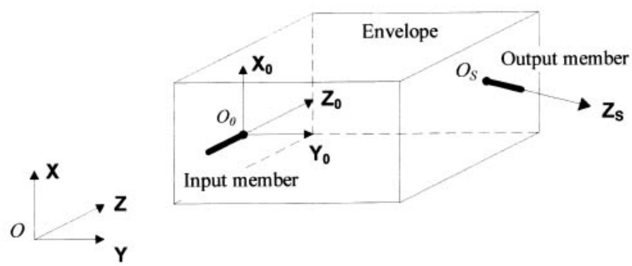
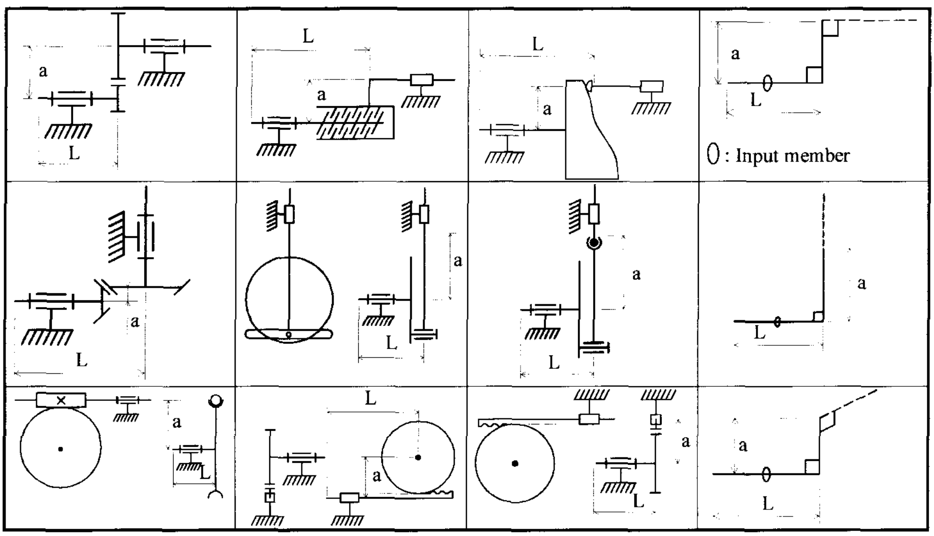


Figure 1. Specifications representation

3 Skeleton principle

The synthesis considered here is a preliminary work intended to make sure, before going any further, that the solution under study can offer a space configuration which satisfies the specifications. We have chosen to base this work on the most simplified geometrical model possible, in order to reduce calculations and thus the checking task to a minimum. We have thus eliminated all volumes from the parts and have retained only the geometric elements which play a role in the definition of the position and the orientation of the output member related to that of input. Each EMB is then represented schematically by a minimal model made up of lines which we call the "skeleton". Table 1 shows the skeleton associated with some components from our EMB database: external cylindrical gear pair, screw mechanism, cam-translating follower (line 1), bevel-gear, eccentric, slider-crank (line 2), worm-gear, rack-and-pinion (line 3). All the mechanisms which combine rotation and translation are conventionally represented in their most retracted position. Table 1 illustrates that primary mechanisms belonging to different classes can share a same skeleton. At the present time, our EMB database contains 39 mechanisms but only 10 different skeletons are necessary to model them all.

Table 1. Some EMB with associated skeletons



4 Geometrical model

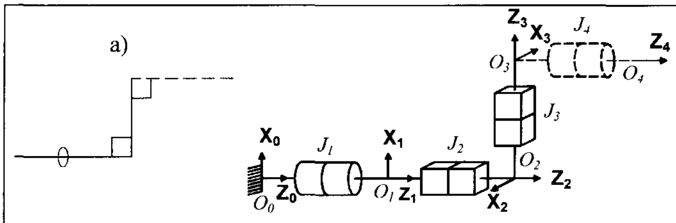
Certain dimensions of a mechanism, and thus of its skeleton, may be changed without compromising the correct working order of the mechanism. At the initial design stage, the structure is not yet sized, so all the dimension and orientation parameters likely to be modified can be considered as problem variables. These variables are either lengths, or angles. These degrees of freedom can be shown schematically by prismatic and rotational joints and in this way our skeleton can be transformed into a kind of deformable structure. The mDH notations [11], well-known in the robotics field, may be used to model the space configurations of this structure. Then, the initial geometric synthesis problem can be seen as a closing chain problem, or as a robot geometrical inverse model research.

4.1 The skeleton geometrical models

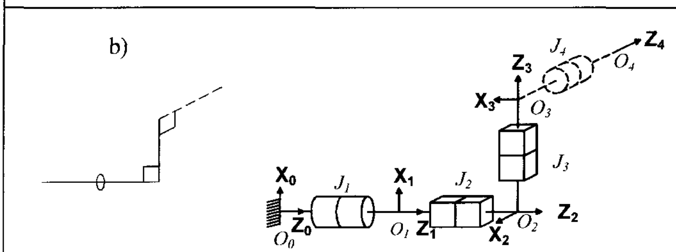
The skeleton of any EMB is now considered as a series of links, the link $(j\text{-}1)$ being connected to the link j by the joint J_j which is either a prismatic joint (length variable), or a rotational joint (angular variable). The transformation matrix allowing the change from the coordinate frame $R_{(j-1)}$, fixed with respect to the link $(j\text{-}1)$, to the frame R_j is:

$$[T]_{j-1}^{j} = \begin{bmatrix} \cos\theta_j & -\sin\theta_j & 0 & d_j \\ \cos\alpha_j \sin\theta_j & \cos\alpha_j \cos\theta_j & -\sin\alpha_j & -r_j \sin\alpha_j \\ \sin\alpha_j \sin\theta_j & \sin\alpha_j \cos\theta_j & \cos\alpha_j & r_j \cos\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where α_j , d_j , θ_j et r_j are the 4 mDH parameters. Figure 2 gives the geometrical model of two different skeletons, the one related to mechanisms with parallel members on opposite sides (fig. 2a) and the other related to mechanisms with concurrent and perpendicular members (fig. 2b). The corresponding mDH parameters are illustrated to the right of the figure, where q_j represents the variable introduced by the joint J_j.



	J_1	J_2	J_3	J_4
σ_j	0	1	1	
α_j		0	$+\Pi/2$	$+\Pi/2$
d_j		0	0	0
θ_j	q_1	$+\Pi/2$	$+\Pi$	
r_j	0	q_2	q_3	



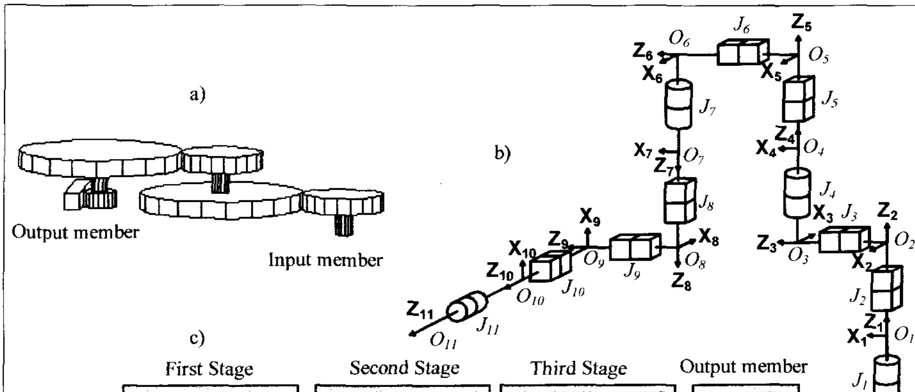
	J_1	J_2	J_3	J_4
σ_j	0	1	1	
α_j		0	$+\Pi/2$	$+\Pi/2$
d_j		0	0	0
θ_j	q_1	$+\Pi/2$	$-\Pi/2$	
r_j	0	q_2	q_3	

Figure 2. Two skeleton geometrical models

4.2 The whole mechanism geometrical model

A geometrical model is associated with each skeleton and a skeleton with each EMB, and as the entire mechanism under study is made of a set of serially connected EMBs, it is easy to obtain the geometrical model of this entire mechanism by just placing the different skeleton geometrical models of the constitutive EMB side by side. Two columns must be added at the end of the whole parameter table to represent the possible variation in the length of the last output member and the possible rotation of the structure about this output member. Note that, using the transformation matrix association, it is possible to build automatically the geometrical model of any 3D SISO mechanism.

Figure 3a shows an optical pick-up mechanism (used to move a compact disk reading head lens) made of three successive EMBs (2 gear pairs and 1 rack-and-pinion). Figure 3b gives its global geometrical model and fig. 3c the related parameters.



	J_1	J_2	J_3	J_4	J_5	J_6	J_7	J_8	J_9	J_{10}	J_{11}
σ_i	0	1	1	0	1	1	0	1	1	1	0
α_i	0	0	$+\Pi/2$	$+\Pi/2$	0	$+\Pi/2$	$+\Pi/2$	0	$+\Pi/2$	$+\Pi/2$	0
d_j	0	0	0	0	0	0	0	0	0	0	0
θ_i	q_1	$+\Pi/2$	$+\Pi$	q_4	$+\Pi/2$	0	q_7	$+\Pi/2$	$+\Pi/2$	0	q_{11}
r_j	0	q_2	q_3	0	q_5	q_6	0	q_8	q_9	q_{10}	0

Figure 3. The whole geometrical model of a pick-up mechanism with $q_2 = q_{11} = 0$ and $q_4 = q_7 = +\Pi/2$

5 Expression of the synthesis problem

The search for a space configuration of the candidate structure is expressed as the search for q_j values which satisfy the following constraints:

- the input member being positioned according to the specification (O_0, $\mathbf{Z_0}$), the output member (O_{NJ}, $\mathbf{Z_{NJ}}$ where NJ is the total number of joints) must take the position and orientation required (O_S, $\mathbf{Z_S}$).
- the entire skeleton must lie inside the specified envelope.

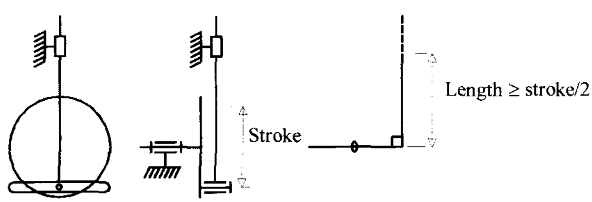


Figure 4. An example of the relationship between the specification and variable bounds

- each q_j value must remain inside its variation domain. The two limits of this domain are defined either from technological considerations (for example, the inferior limit on the number of teeth of a toothed wheel induces the minimum limit on the distance between gear shaft axes), or from an interpretation of the specification sheet. Figure 4 shows that, for an eccentric, the minimal value of a skeleton variable may be directly dependent on a stroke required in the specification.

This kind of synthesis problem is generally redundant as the number of solutions is often infinite, the variable number being higher than the closure equation number. An optimisation criterion has been chosen to find a more suitable solution. As designers often prefer compact mechanisms, we have decided to minimize the overall length of the skeleton. This proves that the result thus obtained provides a good starting point for the next step of our design process (see section 1, step 3) in which technological constraints are added to control the sizes assigned to the main parts.

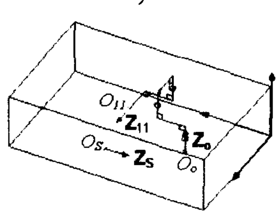
6 Applications

6.1 Optical pick-up mechanism.

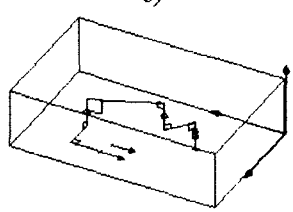
The study of the mechanism already introduced in section 4.2 can be summarised as follows: Table 2a illustrates the initial space configuration of the structure. Default values have been given at the outset to the variables q_1 to q_{11}, leading to an incorrect orientation ($\mathbf{Z}_{11}$) and position (O_{11}) of the output member. The optimisation algorithm manipulates the structure in order to satisfy first the orientation constraints (2b), then the position constraints (2c) and lastly to minimise the skeleton length (2d).

Table 2. Steps of preliminary geometrical synthesis

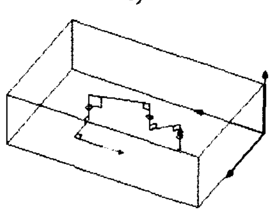
a)



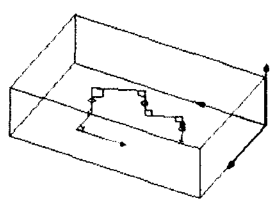
b)



c)



d)



	Initial step	Step 88	Step 662	Final step (1713)
q_1	5 mm	5 mm	5 mm	5 mm
q_2	0 rad	1.07 rad	0.76 rad	0.45 rad
q_3	5 mm	5 mm	5.03 mm	6.04 mm
q_4	5 mm	5 mm	5 mm	5 mm
q_5	0 rad	1.16 rad	1.06 rad	1.51 rad
q_6	5 mm	11.14 mm	10.19 mm	8.73 mm
q_7	5 mm	5 mm	5 mm	5 mm
q_8	0 rad	0.66 rad	0.25 rad	0.39 rad
q_9	5 mm	5 mm	5.01 mm	5 mm
q_{10}	5 mm	5 mm	5 mm	5 mm
q_{11}	0 rad	0 rad	0 rad	0 rad

6.2 Jigsaw mechanism

The main specifications for a jigsaw mechanism are the following (coordinates given in the general frame *Rg*): input member = rotation, O_0=(30,20,0), $\mathbf{Z_0}$=(0,0,1); output member = alternate translation, stroke = 25 mm, O_S=(30,90,60) and $\mathbf{Z_S}$=(0,0.87,0.5). The left side of Fig. 5 presents 3 candidate structures suggested by the first structural synthesis process. Among the numerous solutions proposed, we have deliberately kept here three combinations of two stages, the first stage varying one with the other: bevel-gear first configuration and slider-crank (fig. 5a), bevel-gear second configuration and slider-crank (fig. 5b), worm-gear and slider-crank (fig. 5c). Note that, by using only two stages, it is difficult to satisfy the specification where the input and output members are neither parallel nor orthogonal. In this case, it is useful to evaluate these solutions from a geometrical viewpoint to verify whether they are really suited to the requirements.

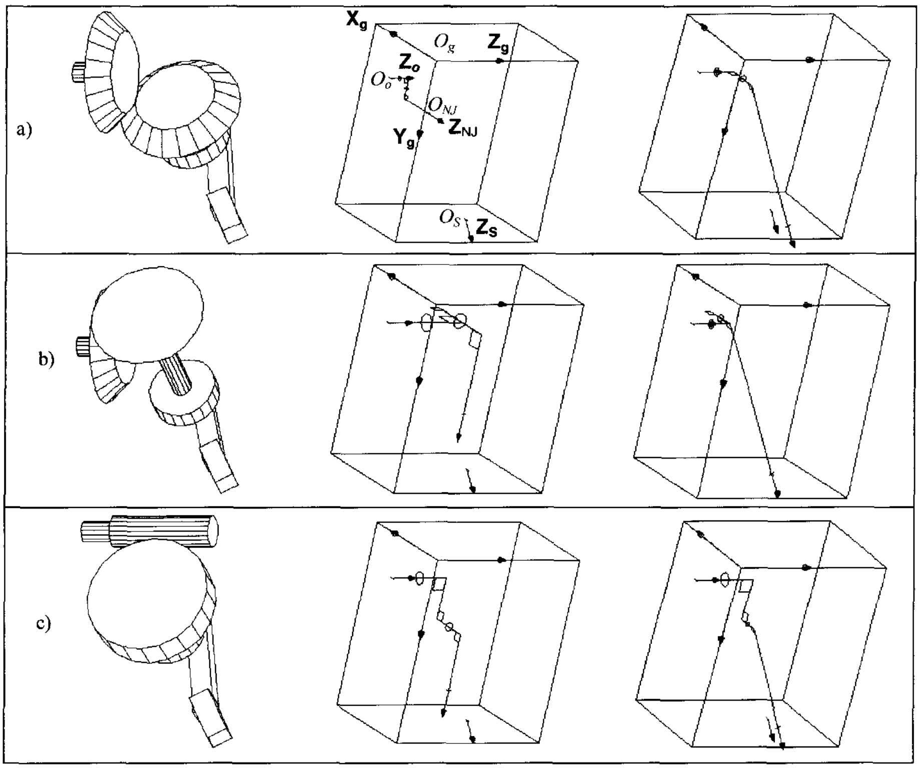


Figure 5. Preliminary geometrical synthesis of 3 structures candidate for a jigsaw mechanism

The initial and final space configuration of each combination are shown in fig. 5. They show that only the second combination is the able to satisfy both the orientation and position constraints.

7 Conclusion

The skeleton principle which consists in representing the main features of mechanism architectures by filar structures at the early stage of design is presented. The possibility of extending the skeleton concept to constructive primitives such as slider-crank, eccentric, rack-and-pinion or cam is considered. Using the well-known mDH notations, an assembly method is proposed, allowing for the automatic construction of the geometrical model of any 3D cinematic chains. This model is used within a closed chain synthesis process to help designers check whether a candidate structure can satisfy both orientation and position constraints.

References

[1] Buchsbaum F., Freudenstein F., "Synthesis of Kinematic Structure of Geared Kinematic Chains and other Mechanisms", Journal of Mechanisms, 5, 1970, pp357-392.

[2] Ravisankar R., Mruthyunjaya T.S., "Computerized synthesis of the structure of geared kinematic chains", Mechanism and Machine Theory, 20, 1985, pp367-387.

[3] Tsai L.W, "An Application of the Linkage Characteristic Polynomial to the Topological Synthesis of Epicyclic Gear Trains", Journal of Mechanisms, Transmissions and Automation in Design, 109, 1987, pp329-336.

[4] Hoeltzel D.A., Chieng W.H., "Knowledge-based approaches for the creative synthesis of mechanisms", Computer-aided design, 22, 1990, pp57-67.

[5] Hoover S.P., Rinderle J.R., "A Synthesis Strategy for Mechanical Devices", Research in Engineering Design, 1, 1989, pp87-103.

[6] Kota S., Chiou S.J., "Conceptual Design of Mechanisms Based on Computational Synthesis and Simulation of Kinematic Building Blocks", Research in Engineering Design, 4, 1992, pp75-87.

[7] Subramanian D., Wang C.S., "Kinematic Synthesis with Configuration Spaces", International Workshop on Qualitative Reasoning about physical systems:QR-93, Orcas Islands, USA, 1993, pp228-239.

[8] Chakrabarti A., Bligh T.P., "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part II: Kind Synthesis", Research in Engineering Design, 8, 1996, pp52-62.

[9] Chakrabarti A., Bligh T.P., "An Approach to Functional Synthesis of Solutions in Mechanical Conceptual Design. Part III: Spatial Configuration", Research in Engineering Design, 2, 1996, pp116-124.

[10] Fauroux J.C., Sartor M., Paredes M., "Using the skeleton method to define a preliminary geometrical model for three-dimensional speed reducers", Engineering with computers, 16 Issue 2, 2000, pp117-130.

[11] Khalil W., Kleinfinger J.F., 1986, "With new geometric notation for open and closed loop robots", Proc. ICRA 86, 1986, pp1174-1179.

Jean-Christophe Wahl
Laboratoire de Génie Mécanique de Toulouse, INSA, 135 avenue de Rangueil, 31077 Toulouse Cedex 04, FRANCE, Tel +33 (0) 561 559 718, Fax +33 (0) 561 559 700, E-mail: jean-christophe.wahl@insa-tlse.fr.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A PRACTICAL METHODOLOGY FOR INTEGRATING SOFTWARE DEVELOPMENT AND EMPIRICAL TECHNIQUES IN DESIGN

P M Langdon, R H Bracewell, L T M Blessing, K M Wallace, and P J Clarkson

Keywords: empirical study; descriptive models of the design process; design philosophy; user evaluation

1 Introduction

A common theme in engineering design research has been the development of models of aspects of the design process using a wide range of research methods, combined with implementation in computer software for the purposes of demonstration and evaluation. However, the methodology employed has frequently been *ad hoc*, reflecting the state of engineering design research as a field. We propose to develop a practical approach to design research methodology, with specific attention to linking (1) integration of research software development and (2) empirical techniques for research evaluation. We describe the background of this approach and the progress of pilot work in the two areas, from which we draw conclusions about the preliminary evaluation of the approach and its capability for improving practical guidelines for design research.

2 Background

A majority of design research, to date, has focused on developing, and in some cases testing, new design methods (see, for example [1]). Little of this research has adopted a rigorous research methodology, which is not surprising, bearing in mind the complexity of the engineering design process and hence design research. Subjectivity in the social and cultural approaches to design limit the generality of any one methodology. However, there are many excellent ideas, theories and methods being developed and it is suggested they are not having the required impact for the following reasons:

- There is no consistent and agreed design research methodology, so research results are fragmented and the resulting design methods not validated [2].
- The difficulty of producing convincing software to validate and demonstrate new methods.

The approach described here aims to continue work previously undertaken [2][3], bringing together methods into a consistent design research methodology, and to develop a flexible and comprehensive software platform on which to build and test demonstrators. The long-term hope is that this will lead to more convincing design methods that will be adopted by industry and thus improve future competitiveness, insofar as those methods can influence product success (Figure 1).

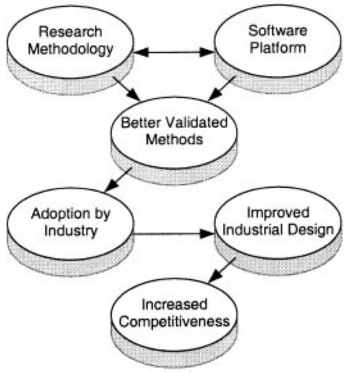


Figure 1. The model for the proposed approach

3 The Need

There is a need for the design research community to adopt a more effective approach to design research in order to maintain the current high standards and to improve the accessibility of research results. In particular, there is a need to ensure that more research delivers journal papers, but also and effective validated tools to industry. The need for integrated research methodology has been widely acknowledged in industry and the academic design research community in the US and the UK[4]. In 1996 the National Science Foundation of America held a strategic planning workshop on *Research Opportunities in Engineering Design*. This workshop identified a set of important research needs, such as:

- create seamless integration between analysis tools,
- create seamless integration between design and analysis.

A recent UK DTI/EPSRC (1999) workshop on *Future Issues for Design Research* brought together industry, funding bodies and academics [5]. This workshop also identified a number of key problem areas and made recommendations that included:

- Setting up a coherent multi-disciplinary framework for design research activity
- Better integration and evaluation of the diverse design support tools and methods.
- The designing of a general experimental and validation methodology for both design research and support tools.

As a result of the perceived need for integrated empirical and computing methodology, we propose a framework for general methodological support that consists of two linked components. One supports empirical and experimental design, statistical and social science techniques, and the other supports the need for engineering design software development and integration of software tools. We aim to link these two components within a common methodological framework. The immediate focus of evaluation of the approach will be directed towards the two areas, using existing research projects as case studies.

4 Theoretical background

Research into the application of design methods is considered to be the domain of design methodology. Very little research has been concerned with evaluating design methodologies and prescribing improvements to them [6] or with the establishment of usable scientific theories of the design process [7]. Design research methodology has been discussed, within the context of design methods, in the UK [8],[9], Germany [10], the Netherlands [11], Switzerland [12], and Japan [13]. For example, there have been calls for more explicit adherence to hypothesis testing and for research to be executed with a clearer purpose [14]. It has been argued that research methodologies are socially constructed and that a study of how engineering design research is carried out within specific design contexts is necessary to improve research practice and results [15],[2]. Duffy and Andreasen [1] reviewed the variable contributions to the field and proposed a more rigorous general methodology based on a scientific paradigm.

There has been growing interest in recent years, particularly in the USA, in the design of high-level software environments to support different areas of design research. SEED [16] supports and integrates research into computational tools for the early phases of building design. N-dim [17] provides a framework and a methodology for rapid prototyping, incremental hardening and deployment of agile design information systems.

Based on the results of descriptive studies, methods and tools have been developed to support design, that were often evaluated with real designers. A design research methodology has been developed by Blessing, Chakrabarti and Wallace [2] to support this research and to piece together the different research approaches in design.

5 Empirical Methodology

The first element is the Design Research Methodology [2], from which the four stage process has been adopted. In the first stage, *Criteria formulation,* measurable criteria are identified, such as “reduced time to market” together with a network of causal influences linking back to success criteria such as “increased profit”. The second stage, *Descriptive Study I,* analyses the existing design process or product aiming to discover relations between the measurable criteria and the design process or product to identify where application of a design method could lead to improvements. In the third stage, *Prescriptive Study,* insights gained in *Descriptive Study I* are used to create an intervention for an improved design process that could result from using the new design support. This intervention creates a starting point for specifying and implementing design support. Finally in *Descriptive Study II* the design support is tested experimentally to determine whether it works as intended and whether it actually impacts the measurable and thereby the success criteria.

The applications of the empirical methods part of the approach range from actually guiding the design and execution of experiments for a specific research need, through to simply providing advice on general empirical goals. The approach aims to give information on the applicability and use of methods involved in experimental design, analysis, data collection or presentation. Empirical methods can be applied at the descriptive level of the methodology or at the evaluative stage. For example, it may be necessary to analyse the existing situation in order to arrive at success criteria , both general and measurable. Various empirical techniques may also be employed to arrive at a suitable descriptions in the description stages. These can be data exploration techniques as well as data analysis techniques and may include, for example, questionnaires and tests; observations; interviews; surveys; cluster and factor analysis. Finally, the design, execution and analysis of appropriate experiments that can evaluate the impact of a design support on measurable success criteria is central to the proposed methodology.

6 Case Study I: Empirical Methods

This case study explored whether a well-known, social science methodological techniques could be usefully applied to an engineering Design Research problem in a hitherto unexplored way. The selection of materials is critical to design and manufacture and this is a process that is assisted by the structured organisation of alternatives. Hence, a conventional clustering technique was applied to a large set of materials on the basis of engineering properties such as their parameters or aesthetic properties. Forty materials were grouped on the basis of eight thermal and mechanical properties and the resulting groupings compared to conventional materials science classification. A Euclidean distance measure was calculated between each material and every other on the basis of a set of features and quantitative parameters whose values were tabulated for every material. The materials were then clustered using a hierarchical, agglomerative method yielding a hierarchy comparable to existing materials science taxonomies. The clustering was implemented using SYSTAT; a readily available standard statistical package. The results showed that engineering material properties could be grouped by the method in ways that were deemed by material scientist to reflect traditional groupings (Figure 2). For example, the clustering of materials on the basis of conventional engineering parameters produced similar hierarchies to science based groupings. Interestingly, some materials, such as aluminium foams were grouped outside of the metals and epoxy/carbon fibre composites were grouped with metals rather than other polymer composites.

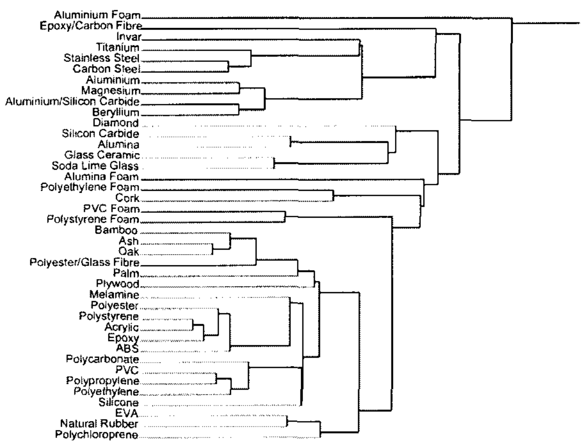


Figure 2. The Materials Clusteriing

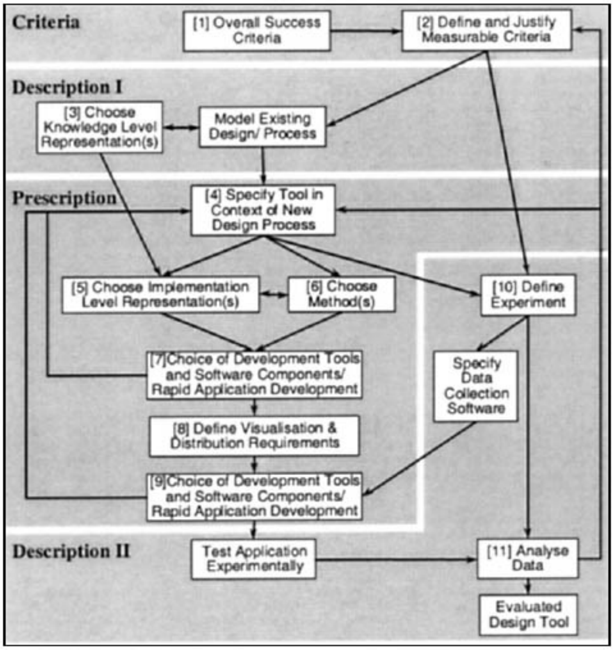
The technique was tested for reliability through the use of a range of grouping methods and distance metrics with the same materials set. It was found that the analysis was robust over methods for this data set and for the other sets reported. This data exploration technique has been applied in a trial [18] and shown promise for extending the approach on the basis of further sets of material, product or design attributes. It was concluded that new material or process groupings resulting from the use of the method might give rise to innovative combinations and that there is scope for extending this approach on the basis of different sets of product or design attributes. This could lead to the development of a software representation of various cluster trees and the method has already been disseminated by submission to an academic journal. [18].

7 Software development

Software implementation issues are addressed in the approach by incorporating two techniques whose technologies are described in more detail in [19 ibid.]. These are: (1) a software independent application development methodology that divides the process into six activities: task definition; choice of representations; choice of methods; definition of distribution and visualisation strategies; theoretical and experimental validation [20]. (2) The software design support required to create robust, functional prototypes, and integrate them into conventional and experimental design systems. (3) The product platform concept [21] where the "products" are modular, prototype tools that are sufficiently functional and robust for evaluation by industry. Using the product platform approach emphasises that it is not the choice of individual functional solutions that is vital but rather that importance should be placed on definition of interface specifications to form a flexible but coherent architecture. Facilitating integration of research tools can remove the practical obstacles that frequently exist in evaluating separate but complementary design tools in combination. (4) A further insight was that an integrated software platform should also provided ideal facilities for automated data collection and analysis during the empirical evaluation of prototype design tools[22]. These considerations are incorporated in the methodology in the "Define Experiment and Specify Data Collection Software" tasks (Figure 3).

8 Case Study II: Software Integration

A pilot industrial case study using the methodology was carried out using work creating and evaluating a computational tool to "sign-post" the helicopter rotor-blade design process [23]. The overall success criteria were defined, the principle one being increased profitability. From this measurable criteria were defined including, time taken to complete process, judged design quality; judged confidence in the resulting solution, and judged gain in experience. The computational "sign-posting" tool exploited knowledge level representations of task parameters, confidence matrices, task clustering and existing text guidelines. A prescriptive intervention was envisaged that would leave the design process tasks unchanged but would optimise the order of execution, reducing the time taken. Consequently, an implementation was made whose representations were based on AI constructs, such as frames and production rules. Standard knowledge-based system methods such as forward chaining, were implemented within the software domain of Lucid Lisp, the Goldworks III IKBS shell and these were linked to existing Fortran 77 rotor blade analysis programs. The Visualisation Requirements were for an interactive GUI tool showing a "Traffic Lights" display for tasks within the context of a process map and capable of generating interactive dynamic graph plots and displaying textual guidelines. To test the effectiveness of the intervention four experienced subjects were compared with four novices in their performance on a task to refine blade mass distribution while minimising pilot seat vibration. Analysis of the results



suggested that the "sign-posting" method increased solution quality, and novice expertise but had little effect on the designers' confidence in the result.

Figure 3. The Software Methodology in Practice

An evaluation of the method was made possible in this case study by integration of AI tools, the GUI, and empirical method

9 Discussion

The two case studies outlined illustrate respectively how an empirically based methodology and an integrated software development environment have the capability for improving practical approaches to design research. In both cases a piece of research would not have been attempted without the practical assistance of the integrated methodology.

The aim of the next stage is to further develop, evaluate and disseminate the framework for an integrated methodology for Engineering Design Research within the context of industrial engineering design. To gain more external validity, it is proposed to incorporate a survey of current research that will be translated into design methods, focusing specifically on integrating the use of experimental and empirical techniques, and design support tool development. This will also enable a methodical assessment of the practical use of design research methods, in general, and the development of guidelines for a practical integrated methodology that can be applied to industrially linked problems. Initially, it is intended to assess current design practice and identify success criteria for the approach. This will academic collaborators. It is important that the methodology and support tools should be disseminated to the academic and industrial design research community and it is intended that it should yield regularly updated internet-accessible, tools, information and design guidance for conducting integrated engineering design software research. Evidence of the impact of using the design research methodology will be derived from data about the relevance, quality and usefulness of developed design techniques and methods. The project will ultimately yield practical design research methods that will be more useful, usable and effective, to both the academic community and industry. This should lead to the development of more effective methods for engineering design.

10 Conclusions

The basis of a practical approach to design research methodology that integrates both software development and empirical methods and evaluation was proposed. This is based on developing and integrating existing technologies along with methodology and practices previously developed by the authors. Two example pilot case studies suggest that the practical use of an integrated approach has the capability to give quick and effective aid to those carrying out specific types of design research. As a result, further research and development of the methodology is justified on a case study basis in an industrial context to evaluate the approach.

References

[1] Duffy, A.H.B., Andreasen, M.M. Enhancing the Evolution of Design Science. 10th International Conference on Engineering Design (ICED'95), Praha, Heurista, (1995).

[2] Blessing, L. T. M., A. Chakrabarti and Wallace, K. (1998). An Overview of Descriptive Studies in Relation to a General Design Research Methodology. Designers - The Key to Successful Product Development. E. Frankenberger, P. Badke-Schaub and H. Birkhofer, Springer: 42-56.

[3] Langdon, P.M., Bracewell, R.H., Duffy, M.H., Sharpe, J.E.E.. An Integrated Design Platform for the Conceptual Design of Complex Mechatronic Systems. Second International Conference on Mechatronics and Machine Vision in Practice (M2VIP'95), Hong Kong, City University of Hong Kong, (1995).

[4] Shah, J J., Bonner, D, Brown, D, Finger, Research Opportunities in Engineering Design: NSF Strategic Planning, Workshop Final Report, NSF, Arizona, USA, (1995).

[5] Culley, S J, Future Issues In Design Research (FIDR) Report, Workshop, Bath EPSRC/DTI, ISBN 1 85790 061 8 (1999).

[6] Reich, Y. (1995). "The Study of Design Research Methodology." Journal of Mechanical Design 117: 211-214.

[7] Smithers, T. (1996). On Knowledge Level Theories of Design Process. Artificial Intelligence in Design '96, Kluwer.

[8] Duffy, A.H.B., O'Donnell, F.J. A Design Research Approach. Critical Enthusiasm, contributions to design science: a "festschrift" for Mogens Myrup Andreasen, Mortensen N.H., Sigurjonsson J. eds, Norwegian Research Council, (1999).

[9] Cross, N. and A. C. Cross (1996). Observations of Teamwork and Social Processes in Design. Analysing Design Activity. N. Cross et al. Eds. Chichester, Wiley: 291-317.

[10] Ehrlenspiel, K. and N. D. Dylla (1989). Experimental Investigation of the Design Process. International Conference on Engineering Design (ICED), Harrogate, UK, WDK 18.

[11] Eekels, J. and N. F. M. Roozenburg (1999). The Geography of Engineering Design Science.

[12] Schregenberger, J.W. The Further Development of Design Methodologies. Designers - The Key to Successful Product Development. E. Frankenberger, et al. Eds., Springer: p. 57-67, (1998).

[13] Hew, K.P. et al. A Fundamental Framework for the Research into Modelling of Synthesis. International Conference on Engineering Design (ICED), Munich, Germany, (1999).

[14] Antonsson, E. K. (1987). "Development and Testing of Hypotheses in Engineering Design Research." Journal of Mechanisms, Transmissions, and Automation in Design 109: 153-154.

[15] Reich, Y. (1994). "Layered models of research methodologies." AI EDAM Journal 8: 263-274.

[16] Flemming, U. and Woodbury, R., (1995), "Software Environment to Support Early Phases in Building Design (SEED): Overview", Journal of Architectural Engineering, v1, n4, Dec 1995, pp147-152.

[17] Reich, et al. Building Agility for Developing Agile Design Information Systems, Research in Engineering Design, 11, p. 67-83 (1999).

[18] Johnson, K.W., Langdon, P.M., Ashby, M.F. *Grouping materials and processes for the* designer: an application for cluster analysis. Cambridge University Engineering Department, Technical Report CUED/C-EDC/TR102 (July 2000). ISSN 0963-5432.

[19] Bracewell, R. H., Shea, K., Langdon, P.M., Blessing, L.T.M., Clarkson, P.J., A Proposed Computational Design Tool Research Methodology, Ibid., (2001).

[20] Raphael, B., Shea, K., Smith, I.F.C., A task and software independent CAE course, AICIVIL-COMP99: The fifth international conference on the applications of AI to Civil and Structural Engineering, Oxford, England, (1999).

[21] Meyer, M. H. and A. P. Lehnerd (1997), *The Power of Product Platforms*, The Free Press, New York, ISBN 0-684-82580-5.

[22] Langdon, P.M. and Cheung, H.F. A Hypertext Documentation System for Schemebuilder: The 'Mechatronics' Book, in Mechatronics : The Basis for New Industrial Development. M. Acar, J. Makra and E. Penney, Eds., Computational Mechanics Publications, Southampton, 1994

[23] Clarkson, P.J., Hamilton, J.R., (1999) 'Signposting' The Design Process, International Conference on Engineering Design (ICED), Munich, Germany, WDK, 107-112

Dr. Patrick Langdon

Engineering Design Centre, Department of Engineering, Cambridge University, Trumpington Street, Cambridge CB2 1TP, UK, Phone: +44 1223 766961, Fax: +44 1223766963, Email: pml24@eng.cam.ac.uk, URL: http://www-edc.eng.cam.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

SHAPE MANIPULATION OF DOMAIN DISTRIBUTED VAGUE MODELS

Z Rusák, I Horváth, J Jansson, J S M Vergeest, and G Kuczogi

Keywords: Vague model, particle, particle cloud, volumetric operators, physical operators

1 Introduction

Virtual claying techniques aim at interactive local and/or global deformation of virtual models by virtual tools. Various implementations of virtual clay models are known. The deformation computation for concrete implementations of virtual claying can be both geometry-based and physically-based. Geometry-based deformation manipulates the shape of the model by modifying geometric parameters (e.g. control points, implicitly defined potential fields [9]) or by applying mathematical transformation (e.g. free form deformation [1]). The geometric model of the virtual object is represented as a surface model or a solid model, which fulfils the continuity and validity conditions. The geometry-based shape manipulation makes it possible for the user to control the deformation, but as an expense, it requires a large amount of inputs from the user. In a physically-based shape manipulation technique, the changes of the shape due to external and internal effects are computed based on physical laws. It means that the deformation of the shape is controlled by the elastic and plastic properties of the supposed material [8]. Computation of the changes of the shape relies on idealisation of the object either by region discretization (e.g. finite elements) or by substitution with an ordered set of discrete elements (e.g. particle systems). Shape manipulation by physically-based virtual claying is known to be a powerful means to support shape conceptualisation and behavioural simulation.

Another interactive shape manipulation technique is sculpting. Virtual object sculpting composes objects from materialistic constituents, and interactively modifies their shapes by adding or removing materials. Galyen et al. developed a sculpting tool controlled by a 3D input device [2]. A virtual object is represented by voxels in the modelling space. The sculpting tools implemented in the system can locally modify the voxel structure. More sophisticated sculpting techniques allow the user to compose objects from virtually existing shapes and to simulate cut and paste processes [2]. These sculpting techniques can be applied to discrete geometric models, but cannot be applied to incomplete geometries. Furthermore, they are better suited to deformation of individual shapes than a cluster of shapes. This motivated the authors to develop a new technique of shape manipulation that combines the principles of virtual claying and sculpting with cluster oriented shape representation, allowing incompleteness.

1.1 The objective of the research and the paper

The Integrated Concept Advancement research program focuses on the methodological development and pilot implementation of a comprehensive computer aided conceptual design system. The core of this system is a novel computer internal artefact representation, which relies on discrete, domain distribution modelling of shapes. The Ph.D. research behind this paper is engaged both with the theoretical fundamentals and with the implementation of vague geometric modelling. The basic modelling entities of the vague modelling are particles, particle clouds, and particle systems. Interested readers are referred to [6], in which the mathematical notions of vague modelling are defined. The ultimate goal is to apply discrete vague modelling to mimic working with true materials. Therefore, we consider it an advanced implementation of virtual sculpting. To this end, we had to develop a concept for volumetric shape manipulation for a domain distributed discrete geometry representation. The primary functional requirements for the technique are (a) composition and modification of shapes and (b) deformation of the shape of point sets individually and in their interaction. The first option is termed volumetric manipulation, the second one is physical manipulation. The aim of this paper is to report on the developed theories and methods for shape manipulation that gives the basis of a designer-friendly shape conceptualisation system.

2 Concise survey of the process of vague modelling

Below, we depict the vague modelling process, with the aim to introduce the basic modelling entities. The main steps of the vague modelling can be seen in Figure 1. The process comprises a set of transformations that convert the input point-set to instances of particle systems. The input point-set is supposed to be dense enough, but not necessarily ordered, 3D point-sets generated by various input techniques e.g. hand movement detection, or 3D scanning. In the course of pre-processing, sufficiently regular particle clouds are generated from the point-sets in four main steps:

- Finding the close neighbour points of each point to obtain the topology of the model. The close neighbour points are found by using the binary space partitioning technique.
- Filtering the measurement errors out based on the variation of distance between close neighbour points. This procedure removes points from the set having close neighbour points in extremely large distance compared to the average distance of close neighbour points of the set. It results in a quasi-regular point-set.
- Calculation of the surface normals by the application of weighted α-shapes. In order to describe the surface normal vectors in each point, first we apply a triangulation technique, called α-shape. From the triangular representation, the surface normal vectors of the points are computed from the orientation of the neighbor triangles of the concerned vertex. The whole method results in an instance particle cloud with explicit neighbourhood relationship, and the approximated surface normals are described for each point.
- Positioning of the particle clouds in a reference frame and merging the component point clouds to generate the vague model.

Vague modelling aims at an explicit representation of the distribution domain of a model. The

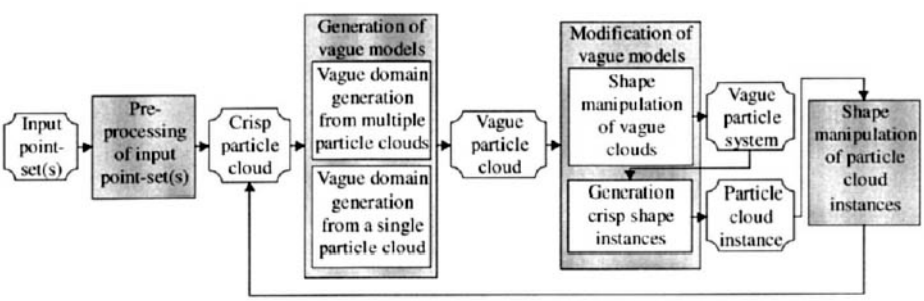


Figure 1. The process of vague modelling

procedure of generation of vague models involves (i) finding the maximum and minimum extents for the united point sets, (ii) calculation of the principal distribution trajectories, (iii) quantification of the metric occurrences, (iv) generation of the particles of the vague model. We defined two forms of deriving the distribution domain of a vague model. They are called initial domain definition and domain expansion, respectively. When the distribution domain of a particle system has to be derived from more than one discrete particle clouds, the resultant borders of the distribution domain are generated as the maximal and minimal coverings of the superimposed particle clouds. When the vague model has to be generated from one single particle cloud, the scattering of the points on the boundaries describes the distribution domain. Obviously, different operations need to be applied for the two cases, but the actual result is a discrete, vague particle system.

Two groups of operators are of significance for manipulating discrete vague models. The first group enables us to manipulate the shape of a particle cloud/system in the phase of building the vague model. By manipulation of their shapes, on the one hand, we can modify the geometric and morphological properties of the component particle clouds. On the other hand, shape manipulation can also be applied to modify the shape of the resultant composition of particle clouds, called the particle system. By means of the second group, particle cloud instances can be extracted from a vague model. Between its minimal and maximal coverings, a vague model contains infinite number of shape instances. Rule-based instancing provides an intuitive way of shape modification. The shape instances are represented by specific particle clouds. Individual shape instances are derived in the distribution domain based on shape formation rules, specific to a given application field. A rule-based instance generation can overcome the limits of conventional parameterisation techniques, but in certain cases formal expression of the shape forming rules is not a trivial task [7]. The particle cloud instances represent crisp discrete shapes, whose geometry can be further modified by physical deformation. These instance models can be used as input geometry for FEM analysis. If needed, particle cloud instances can be reused in the vague modelling process. Alternatively, by applying reverse engineering techniques, they can be converted to continuous geometric models, typically used by commercial systems.

3 Functionality needed for shape manipulation of vague models

Shape manipulation by sculpting relies on three principles: (a) adding material, (b) removing material, and (c) deforming the shape. In order to be able to computerise these principles, we have developed a set of operators. An operator is a mathematical function defined to change certain properties of the modelled object. Figure 2 shows the set of operators implemented in the vague sculpting tool.

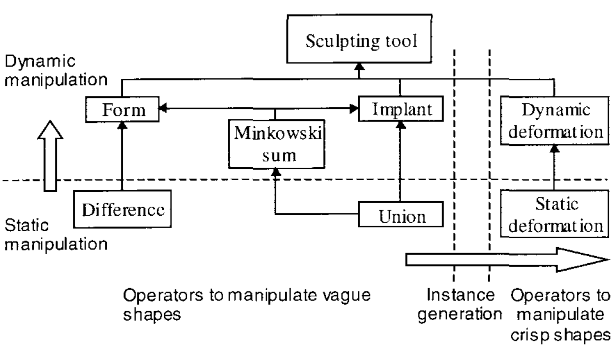


Figure 2. Operators implemented in the sculpting tool

From the point of view of human-computer communication, we can distinguish static and dynamic manipulations. An operator is called static, if all the parameters of the mathematical function of the operator are defined before computing the related mathematical functions. For instance, a regular Boolean union operation on two closed solids is a static operation, since it requires us to specify the two shapes to be combined and the positions of these shapes in advance. An operator is called dynamic, if the user can manipulate certain parameters of the mathematical function during computation. For instance, cutting is a dynamic operation, since the user can modify the path of the cutting tool in order to change the geometry of the model.

Operators implemented in the sculpting tool can manipulate vague or crisp discrete shapes (see Figure 2). The goal of manipulating vague shapes is to create compound elements (particle systems) from a set of elementary geometric entities (particle clouds), or to modify the shape of particle systems by adding/removing parts from the geometry. To achieve these goals, we defined two static operators, *union* and *difference*. They are basically the application of Boolean union and difference to a vague discrete model. The functionality of these operators is to provide the user an effective manipulation tool, which is able to combine two or more particle clouds. The union and difference operators can be used for placing particle clouds of feature type on a vague geometry. The *Minkowski sum* defined on a vague model is an auxiliary operator of the sculpting tool. It is a powerful means for shape generation by sweeping 3D solid models [4]. It sequentially unifies the geometry of an object along a trajectory. The trajectory of the operator is not necessarily defined before computation, therefore the Minkowski sum is a dynamic operator. We defined the *form* operator as a combination of Minkowski sum operator and difference operator. The purpose of the form operator is to simulate cutting technologies, e.g. milling, turning. Similarly, it is purposeful to define an implant operator by grouping the Minkowski sum and union operators together. The implant operator allows the user to dynamically specify a 3D geometry by application of Minkowski sum, and merge the result with the existing geometry. This manipulation is very similar to the mechanics of a 3D-printing device. Since all vague operators manipulate the volume, we call them *volumetric operators*.

Simulation of deformation of crisp geometries due to external or internal effects is useful both in finalising the shape of objects, or evaluation of the physical behaviour of objects. The theories behind crisp shape manipulation are the physical laws (e.g. law of conservation of mass [4]), therefore, we call these operators *physical operators*. Physically-based deformation of an object can be achieved by applying external forces and/or internal forces. In this paper we deal with these mechanical deformations of shapes that are caused by external forces. The reason is that our aim has been to develop a virtual sculpting tool. In the real life deformation of an object can be viewed from two aspect: (a) focussing on the result of a deformation process, (b) focussing on the course of deformation. The target system has to be capable to handle both. To this end, we formalised the first case as static deformation, and the second case as dynamic deformation. To realise these manipulations in our system, we have developed dedicated operators that take a shape instance as input. The static operators causes time independent changes of a shape, while the dynamic operators generate time dependent deformation changes. Static deformation requires from the user to specify the applied external forces before computation. Dynamic deformation assists the user to use virtual objects (e.g. tools, his hand) to manipulate the design on the modelling space.

4 Positioning of particle clouds

Implementation of the volumetric operators needs (a) to position the particle clouds in a global reference frame, and (b) to manipulate the shapes by modifying the metric occurrences

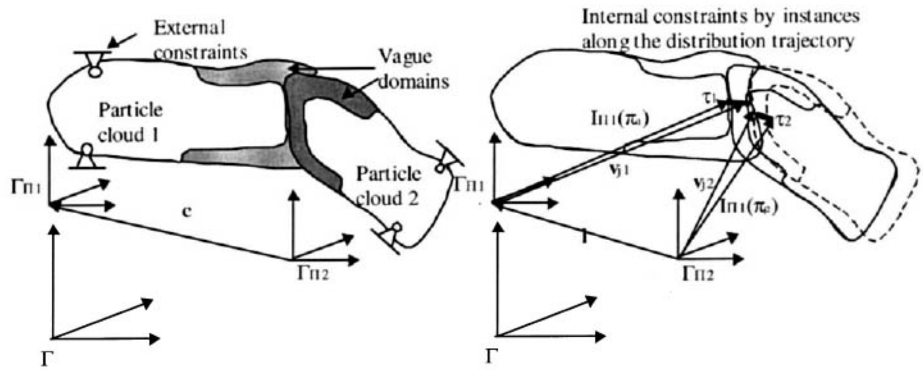


Figure 3. Fixed and flexible positioning of particle clouds

or by removing/generating new particles. There are two options for positioning depending on whether we have external or internal constraints in arranging the particle clouds in the modelling space. External constraints immovably position a particle cloud Π_1 relative to Π_2. It means that once the particle clouds are positioned, their reference frames are not influenced any more by the instance generation. To give an example, external constraints can be applied to position an arbour relative to a bearing by the central axis of each element. In this way, we can model the tolerancing of these elements. The mathematical formulation of fix positioning is defined as follows. Particle cloud Π_1 is *fixed positioned* relative to particle cloud Π_2, if the local reference frames $\Gamma_{\Pi 1}$ and $\Gamma_{\Pi 2}$ of Π_1 and Π_2, respectively, are defined so as $\Gamma_{\Pi 1} = \Gamma_{\Pi 2}+\mathbf{c}$, where $\mathbf{c}$ = const. is the vector marking from the origin of $\Gamma_{\Pi 2}$ to $\Gamma_{\Pi 1}$ in the global reference frame Γ.

Internal constraints reposition the local reference frames of Π_1 and Π_2 based on the selected instance points. This type of positioning is applied, for example, to place a local feature on a global shape with the condition that the interacting surfaces of each element have to touch each other for all instance shapes. Flexible positioning is defined as follows: A particle cloud Π_1 is *flexibly positioned* relative to Π_2, if the local reference frames $\Gamma_{\Pi 1}$ and $\Gamma_{\Pi 2}$ of Π_1 and Π_2, respectively, are defined so as: $\Gamma_{\Pi 1'} = \Gamma_{\Pi 1} + \frac{1}{N}\sum_{i=1}^{N} I_{\Pi 1}(\pi_i) + \frac{1}{N}\sum_{j=1}^{N} I_{\Pi 2}(\pi_j)$

This mathematical function $\Gamma_{\Pi 1'} = f(\Gamma_{\Pi 1}, I_{\Pi 1}(\pi_i), I_{\Pi 2}(\pi_j))$ is defined by the original location of $\Gamma_{\Pi 1}$, the instancing functions $I_{\Pi 1}(\pi_i)=\alpha_i \tau_i$ of Π_1 at particle π_i, $\pi_i \in \Pi_1$, $\pi_i \in R_1$ and $I_{\Pi 2}(\pi_j)=\alpha_j \tau_j$ of Π_2 at particle π_j, $\pi_j \in \Pi_2$ $\pi_j \in R_2$. R_1 and R_2 are the influenced regions specified on Π_1 and Π_2 either by the user or by computing the volumetric operation. α_i and α_j is specified by shape formation rules applied in the instancing function. τ_i and τ_j is the distribution trajectories at π_i and π_j. Figure 3 shows an example for fixed and flexible positioning of two particle clouds.

5 Static volumetric operations on vague geometry

The mathematical apparatus of volumetric operators is defined based on set theory. Basically, each particle cloud is defined by two sets of particles, boundary and internal ones. *Boundary particles* $\Pi_B\{\pi_{B1} \ldots \pi_{BN}\}$ of a particle cloud Π are a subset of all particles which are the touching both the primary ϕ_P and secondary coverings ϕ_S, and located on the opposite side of ϕ_P and ϕ_S. *Internal particles* $\Pi_I\{\pi_{I1} \ldots \pi_{IN}\}$ of Π are a subset of all the particles not touching

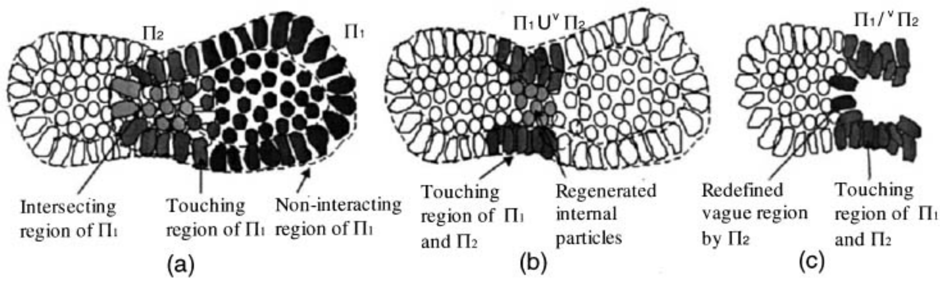


Figure 4. Regions of particle clouds in volumetric operations

ϕ_P, and located on the same side of ϕ_P and ϕ_S . Since the metric occurrence ε_I of an internal particle π_I of Π does not bear with vagueness information, it only represents a part of the volume and the material of the Π, the shape of ε_I is described by a spherically bounded half-space.

By relative positioning of two particle clouds we can identify three subsets for each particle clouds: (a) non-interacting, (b) touching, and (c) intersecting regions. *Non-interacting region* R_{NI} of particle cloud Π_1 on Π_2, is the set of the particles, which do not intersect with the particles of Π_2. *Intersecting region* R_{IS} of particle cloud Π_1 on Π_2, is the subset of particles $\{\pi_B\ \pi_I\}$ which intersect with the internal particles of Π_2. *Touching region* R_{IA} of particle cloud Π_1 on Π_2, is the subset of particles of Π_1, which intersect with the boundary particles of Π_2. Figure 4a shows the three regions of Π_1 as it intersects with Π_2. It follows from the definition of intersecting and touching regions, that there exist boundary particles, which can be part of both regions. In this case, the given boundary particle is assigned to the touching region.

The *union operator* $\cup^v$ defined on Π_1 and Π_2, so as $\Pi_1 \cup^v \Pi_2$ (a) positions Π_1 and Π_2, (b) describes the touching, non-interacting, and intersecting regions on both particle clouds, (c) combines the Π_1 and Π_2 to a particle system $\Pi_{12}\{\Pi_1 \cup^v \Pi_2\}$, (d) removes the boundary particles from the intersecting regions, (e) regenerates the internal particles in the intersecting regions with the averaged mass, metric occurrence and density of the internal particles of Π_1 and Π_2.

The *difference operator* $/^v$ defined on Π_1 and Π_2, so as $\Pi_1 /^v \Pi_2$ (a) positions two particle clouds Π_1 and Π_2, (b) describes the touching, non-interacting, and intersecting regions on both particle clouds, (c) combines the Π_1 and Π_2 to a particle system $\Pi_{12}\{\Pi_1 /^v \Pi_2\}$, (d) removes the particles of Π_1 from its intersection region and the internal particles of Π_2 from its touching regions, and the internal particles of Π_2, and the boundary particles from its non-interacting regions.

6 Dynamic volumetric manipulation

Dynamic volumetric manipulation of vague shapes is realised by the application of the Minkowski sum operator for a vague geometry. The Minkowski sum operator is a continuous sequence of union operators applied on a particle cloud along a trajectory T. We approximate the trajectory T with a polyline. Hence, the task reduces to finding the Minkowski sum of the tool with a straight line segment L. The Minkowski sum operator $\oplus^v$ on Π_1 along L is defined as follows: $\Pi_I \oplus^v L = \cup^v_{l \in L} (\Pi_l + l)$. The positioning of the union operation $\cup^v$ applied in the

Minkowski sum is restricted to fix positioning, since the positioning of the particle clouds is described by L. The result of a Minkowski operation is a newly generated particle cloud Π_{1M}. The dynamic form/implant operator is described as the combination of union/difference operator and the Minkowski sum operator. The form operator $\vee$ using Π_1 on Π_2 along L is defined as follows: $\Pi_1 \vee \Pi_2 = \Pi_2 \cup^v (\Pi_1 \oplus^v L)$. Similarly, the implant operator $\wedge$ using Π_1 on Π_2 along L is defined as follows: $\Pi_1 \wedge \Pi_2 = \Pi_2 /^v (\Pi_1 \oplus^v L)$. Both operators allows the user to specify the type of positioning between Π_{1M} and Π_2. In this way, the Π_{1M} is interpreted as a feature of manipulation, and it is repositioned regarding to the selected instance shape of Π_2.

7 Operators for physical deformation of objects

The model of deformation of crisp shapes, called general mechanics model, is based on inter-atomic interaction, which is viewed as a mass-spring model. The theory and implementation of the general mechanics model is discussed in [5]. Figure 5 shows an example for static and dynamic deformations. *Static physical deformation* (a) identifies the region on which the physical interaction is applied, (b) specifies the acting external forces, (c) specifies the time interval Δt in which the forces are effective, (d) discretises the time interval (e) calculates the new positions of particles according to the forward Euler method. Static physical deformation requires all the necessary data from the user in advance. Thus, the character of the interaction between the user and the shape is indirect. However, static physical deformation is a useful operator for behaviour of an object, when there is a need for modelling the effect of known mechanical loading. In order to allow virtual physical interaction between the user and the model, we defined the dynamic physical operator. *Dynamic physical deformation* (a) detects colliding regions, (b) calculates the external forces based on the mass and velocity of the particle clouds and the change of the time, (c) applies static physical deformation of particle clouds. Dynamic physical deformation is a virtual claying tool allowing the user direct interaction with the shape. It effectively supports shape forming.

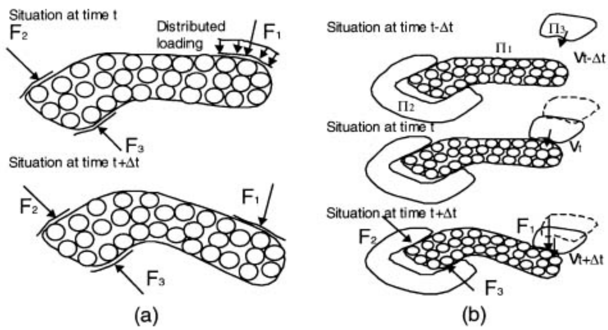


Figure 5. Static and dynamic physical deformations

Figure 6. Vague model of a cup

8 Application example

Figure 6 shows the vague model of a cup. First, the body of the cup and the sweeping tool (below the cup) are generated from multiple point-sets. The handle of the cup has been created by using Minkowski sum operator on the sweeping tool. Finally, the handle and the body are merged together by applying the static union operation.

9 Conclusions

In order to manipulate vague shapes, we have developed and implemented a collection of vague sculpting tools. The collection contains a set of volumetric and physical operators, which can be static or dynamic type from the point of view of human-computer interaction. Regarding to vague manipulation, we defined fix and flexible positioning of particle clouds, which provide the user more freedom in the design process during instance generation.

References

[1] Coquillart, S., "Extended free-form deformation: a sculpturing tool for 3D geometric modeling", *Computer Graphics*, 1990, Vol. 24, No. 4, pp. 187-196.

[2] Chandru, V., Mashes, N., Manivannan, M., Manohar, M., "Volume sculpting and keyframe animation systems", in *IEEE Computer Animation 2000 Proceedings*, 2000, pp. 134–139.

[3] Galyean, T. A., Hughes, J. F., "Sculpting: an interactive volumetric modeling technique", *Computer Graphics*, 1991, Vol. 25, No. 4, pp. 267-274.

[4] Ilies H., Shapiro V., "The Dual of Sweep", *Computer-Aided Design*, 1999, Vol. 31, No. 3 , pp 185--201.

[5] Jansson, J., Vergeest, J. S. M., "A general mechanics model for systems of deformable solids", *in Proceedings of the Third International Symposium On Tools and Methods of Competitive Engineering 2000,* Delft University Press, pp. 361-373.

[6] Rusák, Z., Horváth, I., Kuczogi, G., Vergeest, J. S. M., Jansson, J., "Discrete domain modeling for shape conceptualization", *in Proceedings of the 4th International Conference of Engineering Design and Automation* - EDA 2000, 2000, pp. 228-233.

[7] Rusák, Z., Horváth, I., Kuczogi, G., Vergeest, J.S.M., "Techniques for generating shape instances from domain distributed vague models", *Proceeding of UkrObraz2000*, 2000, pp. 29-32.

[8] Terzopoulos, D , Platt, J., Barr, A., Fleischer, K., "Elastically deformable models" *Computer Graphics*, 1987, Vol. 21, No. 4, pp. 205–214.

[9] Wyvill, G., McPheeters, C., Wyvill, G., "Animating soft objects", in *The Visual Computer, International Journal of Computer Graphics,* 1986, Vol. 2, pp. 227-234.

Zoltán Rusák
Faculty of Design, Engineering and Production, Delft University of Technology, Jaffalaan 9, 2628 BX, Delft, The Netherlands, Tel: +31 15 278 3788, Fax: +31 15 278 1839,
Email: z.rusak@io.tudelft.nl, URL: http://www.io.tudelft.nl/research/ica/index.html

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

ANALYSIS PRODUCT DEVLOPMENT TOOLS BY LOOKING AT THEIR LIFE-CYCLE

C S Araujo

Keywords: Integrated product development, design tools, selection of design tools, DFX.

1 Introduction

A large variety of Product Development tools (i.e., methods, techniques, concepts, methodologies, etc) are presently available - both in paper and computer based formats. These tools are always developed and promoted with the assumed aim of supporting the establishment, execution, and control of all kinds of PD related activities (both technical and management). However, it has been recently reported in the literature that, from all of the tools that are made available, only a very small fraction is ever implemented in practice (Araujo et al, 1996). Moreover, from those that are implemented, the delivered results are often reported as being lower than the expected (Barkan, 1994).

In order to be better developed, as a means to increase their chance of being (a) chosen and then (b) successfully implemented by practitioners, PD tools must first be better understood. The hypothesis bounding this work is that the existence of a comprehensive understanding of PD tools would provide all of those involved with PD tools (developers, promoters, marketers and users) with more appropriate information to support their activities. One possible way for analysing the many complexities related to PD tools is by means of looking at what happens to the tools along their life-cycle (Figure 1). For that purpose the traditional concept of product life-cycle has been borrowed. Indeed, just as the life-cycle concept is useful at disclosing important aspects of product development in practice, it could, it is proposed, disclose interesting issues related to the development of new PD tools by means of setting a frame through which every aspect of tool development can be considered. The research approach is descriptive in nature, in which a phenomenon was observed and structured using a frame or reference (in this case the concept of life-cycle).

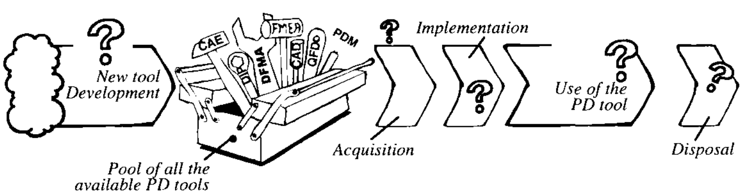


Figure 1. Phases in the life-cycle of product development tools: One way of understanding tools

The work is based on the information and conclusions gathered from a survey conducted in the United Kingdom on the use of PD tools in the industry (Araujo et al, 1996). The conclusions from this survey was then confronted with a series of interviews conducted with experienced PD consultants in Denmark, and also with the results from a focused survey involving 10 Danish manufacturing companies. The details of the reasoning (graphics, analyses, etc) behind of what is presented in the article are discussed in (Araujo, 2001).

2 Development of new PD tools

One of the key assumptions bounding the development of new PD tools is that they are designed having the goal of enhancing the practice of product development in industry. The starting point is thus the identification of a certain need for a certain kind of tool. A number of different sources provide the ideas for the new tools, and can initiate (trigger) and guide the process, including:

- The recognition of a real need in the context of PD, as identified by means of formal or informal research.
- A formal request from the organisation (or set of organisations) in need for a 'solution' for some new or old problem.
- New guidelines and/or regulations from government bodies, certification agencies, etc.
- The perceived potentialities offered by a new technology, or ideas from unrelated sources.
- The opportunity offered by the recognition that further improvement in an existing PD tool is necessary and/or possible.
- The opportunity offered by the identification of a potential new market for an existing (and not appropriate for that market) PD tool.
- The opportunity offered by the recognition of an under-developed theoretical idea or concept from academia.
- The opportunities identified by means of foreseen the directions changes and new conditions".

Similar to the general problem of product development, development of new tools can also be characterised by means of three dimensions:

- The *degree of novelty* in the content of the technologies/ideas involved in the new tool.
- The *market for the new tool*, considering the development of a new PD tool in order to satisfy either a new or class of practitioners.
- The degree of novelty of the problem that the new tool will solve, or the task that it will support (new or old problems/tasks).

Apart from its dependence on the dimensions shown above, the way in which the development itself is actually carried out, including the establishment of the idea for the new tool, is also dependent on who is undertaking the development task, that is, the author of the new tool. Generally speaking, there exist three major 'sources' of creators (developers) of new PD tools, namely, (a) Scientists and researchers, (b) Specialised tool developers, and (c) PD organizations. It is important to note, however, that most of the tools presently being used by industry have likely suffered inputs from all of these 'sources' (authors). Thus, the distinction of tool authorship is only relevant for academic purposes.

2.1 PD tools as a result of academic work

A considerable amount of what is presently available in terms of PD tools has its origin in the academic means. They can be both computer and paper-based tools. Paper-based tools will include *methods*, *techniques*, *procedures*, *guidelines*, etc, and are disclosed to the public by means of research papers, books, booklets, master and doctor thesis, etc. Computer based tools, on the other hand, include the full range of computer based systems delivered as the result of academic projects.

The actual track in which the development will proceed once the idea has been established is intrinsically related to the research methodology adopted by the researcher that, by its turn, is strongly related to the outstanding tradition in the academic arena where the new tool is being developed. In spite of the bias towards fulfilling the scientific goals and expectations of the project, the work of this specific class of developers is also characterised by the expectation that the result of their work, that is, the new tool, will have a practical value.

Expectation for practical value is increasingly becoming stronger, mainly pushed by the new evaluation standard set by most of the granting bodies (see for instance Andreasen, 1996). Having to satisfy these two different, and somehow conflicting, classes of requirements, the result is that academics rarely have the necessary time, and motivation, to focus on the appropriateness of the developed tool to the intended market. This is seen to be one possible reason for the lack of success of their tools in practice.

Indeed, in the last couple of years a number of different authors are realising that the effect of the tools delivered by researchers has not been convincing, and their acceptance by industry has been generally poor (e.g. (Gill, 1990). Part of the problem is said to be intrinsically related to the poor *quality* of the tools delivered by researchers, which are generally perceived by practitioners as being too theoretical in nature, too complicated to understand, presented in a 'strange' kind of language, difficult to implement, difficult to use, and also difficult to measure in terms of delivered benefits.

Further "refinement" is necessary, which provides with lots of opportunities for another kind of business: The business of re-engineering the 'academic tools' into applicable "solutions" for diverse industrial sectors, and various levels of hierarchy within companies.

2.2 Companies specialised in developing and marketing new PD tools

Mostly working at the borderline between academia and PD organisations, the other group of new tools developers are the companies specialised in the business of providing 'solutions' for PD organisations. An example of such companies is The Adept Group (1).

Convinced of the intrinsic value of a PD tool called NewProd, developed by (Cooper, 1995), this small consultant company decided to buy the rights for the tool, and then to 'translate' the NewProd model into a software package. The results have been good, and the company now commercialises this rather simple software package for US$ 50,000 (including site license and training in the model). However, all that is presented in the computer based version of the NewProd is also presented in Cooper's original book "Winning at New Products", marketed in the USA for US$ 25.00 (nothing less than two thousand times cheaper!).

2.3 Tools originating in the Industrial PD organisations

A less typical, but far from being less important, new tool development process is carried out internally by industrial organizations. In this case the process is mostly done informally, and is often the matter of a gradual process of developing and establishing new ways for carrying out old or new tasks or solving problems. This process is strongly influenced both by the dominant figures in the PD organisation of the company wherein the phenomenon takes place, and also by the cultural/social context specific to the company, which makes it complicated to establish any theoretical description for the phenomenon. In any case, the process was found to be composed by two steps, or phases.

The first phase refers to the original process of establishment of the new tool in the PD context. As said before, this is mostly done informally, and may take many years until a new concept, a new means of doing an old or new task, can be identified and established as a formal procedure for this company. The second phase in the process refers to the formalisation of the tool. In a sense, it is the result of this second phase that we can call a PD tool. The person undertaking this task can be either an internal staff in the organisation, or an external observer (researcher or consultant).

The exact way that this process is carried out certainly varies. Nonetheless, the following explanation, given by (Dimancescu & Dwenger, 1996) (who works as a consultant in the area of Product Development), appears to present a good description of what is generally involved: -"*Our process starts by identifying successful companies in selected industries according to profitability, growth of market share, or customers satisfaction ratings. This done, methods and techniques associated with the success are isolated as best practice.*"

PD Tools developed directly from observation of practice have a number of good and bad characteristics. Perhaps the strongest good characteristic of these tools emerges from the fact that practitioners have developed them from PD practice. One direct consequence from that is the language used to describe the tool. In the general case, it will be presented in an easy to understand language and format. That fact alone delivers immense benefits in terms of easy of implementing those tools, and consequently in a higher chance of success.

Another relevant aspect of this approach is probably a bit more perceptual than factual. Because these tools are often sold as 'the secret from the best', they tend to create a feeling of empowerment. It is consequently more likely that practitioners will have a more positive approach towards these tools, which might facilitate the process of implementation, and likely lead into better results (real or perceived). However, and contrary to what promoters tend to say, it is simply non-founded the hypothesis that what is a 'best practice' to a certain company will necessarily be also a good practice to another. A good example of this problem is the general lack of fitness of the many of the Japanese tools into the Western companies, especially those that demand cultural changes as a basis for its implementation. It strongly supports the hypothesis that there exists more in the successful implementation of a new PD tool in the context of use than in the tool itself.

3 Promotion of new PD tools

Promotion of tools refers to the general effort of making the potential users (practitioners and PD organisations) aware of their existence, benefits, and potentialities. Different tool developers use different means for disseminating their tools. From one side, academics will mostly deliver their new tools by means of research papers, books, and try to disseminate them using scientific conferences. On the other hand, specialised companies will more likely use more traditional channels to sell their tools. This will include workshops and practice-oriented conferences, specialised magazines and, in some specific cases, direct advertising by traditional means, such as bulk mail, internet, on-site visits, etc.

The emphasis of the 'marketing' campaign is also different. From one side academics will try to sell their products on the basis of its worthiness as scientifically proved in their experiments. Specialised companies, on the other hand, will likely try to persuade their potential customers based on the robustness of the tool, as proved by the so many successful companies that are already using it, or based on the technological aspects embedded in the tool. It goes almost without saying that the second method is much more efficient than the first one, as we can readily see by the higher success rate achieved by the later kind of tools (See, for example, Araujo et all, 1996). It should be noted that professional PD tool promoters can be very aggressive in their effort to sell their tools, always making very extensive use of all of the best available selling tools and strategies (see Araujo, 2001).

4 Acquisition of new PD tools

Acquisition of PD tools refers to the generic process in which, given the recognition of a need in the context of PD, decision makers (project leaders, designers, PD managers, etc.) will engage in a process which will lead to the decision regarding the selection of a certain PD tools, out of a collection of identified options. Acquisition of tools can be understood at least at two distinct levels. The first level refers to selection among different concepts of PD tool (e.g. choosing between introducing *QFD* or *DFM*.) The second level refers to selecting among different applications of a certain concept (e.g. choosing among acquiring a book about *QFD*, or buying a *QFD* software package). The process of acquisition of new tools can be quite complex, and take a long time, depending on the degree of involvement of the practitioner(s) in the process, the perception of the costs/resources involved in the introduction of the new tool, and on the degree of impact the introduction new tool is perceived to cause in the intended organisation. The process of acquisition of PD tools is the main topic of the research investigation conducted by this author (from which this article was brought) and is discussed in (Araujo, 2001). Once the decision-makers have agreed in introducing a specific new tool, the next step refers to the implementation of the tool in the practice.

5 Implementation of PD tools in the practice of PD organisations

Implementation of new PD tools is a complex process. In the last couple of years a number of authors have devoted attention to investigating this important topic (e.g., (Norell, 1996)). Indeed, no PD tool has the attribute of self-implementation. This has to be carried out by others, and will often involve the consumption of valuable resources from the organisation (money, time, etc). The process can be carried out in a number of different ways, with varied degrees of effectiveness. The factors account for the differences in the implementation practice will likely include:

- The perceived impact that the introduction of the new tool will cause in the organisation.

- The perceived degree of relevance of the new tool, measured by the degree that members in the organisation (especially management) believe the tool has the potential of delivering important and demanding benefits.
- The person (or group of persons) that carry out the implementation process.
- The perception of the intended users towards the new tool being introduced.

Depending on the specific situation and context, the implementation of the new tool can be the responsibility of different actors, including (a) internal staff, (b) the expected users of the tools, (c) hired specialists and (d) the vendor of the tool. The method applied in the implementation also varies widely from case to case. In general, various methods are common, including (1) individual learning, (2) workshops and (3) pilot projects.

6 Results achieved by companies introducing new PD Tools

From a company's viewpoint, the sole reason for introducing and using new tools is for boosting improvement in the organisation. The extent to which companies can really profit from introducing new tools has always been a very sensitive subject to academics working in the PD area. While promoters insist on the intrinsic value of their tools, recent publications have been keen to point a rather usual low level of benefits achieved by companies implementing new tools (Barkan, 1994). At the heart of the discussion, it is suggested, is the lack of a valid framework for measuring the actual practical value of any new PD tool.

Altogether, the fact is that new tools come with no warranty of what they will actually deliver, once they have been introduced in a certain PD organisation. The best one can do is to try to maximise the benefits, at the same time that they try to keep the side effects (drawbacks) at a minimum level (Cantamessa, 1997). In general, evidence presented in the literature indicates that companies implementing tools are collecting not only good results, but also many bad outputs. Some typical good and bad results achieved by companies introducing new PD tools are briefly discussed in the next two sections.

Good results (benefits)

The following are typical [perceived or real] good results reported by companies implementing new tools:

- Improvement in the quality of the execution of the tasks/problems supported by the tool.
- Improvement in the quality of the resultant products.
- Reduction of the time spent on handling the task or solving the problem (lead time).
- Reduction in the costs involved in the development.
- Control improvement.

Apart from those above, an interesting aspect of introduction of certain tools in practice is the delivery of a range of 'unexpected' good results. In this case, a certain tool, originally introduced with a certain purpose in mind will deliver not only positive results regarding that purpose, but also improvement in other dimensions of PD in practice. A good example of tools presenting such peculiarity is the *QFD*. Even though the tool has been originally developed with the basic functionality of "translating the user's desires into design requirements'' various lateral benefits have been consistently reported, such as the improvement of communication among team members, etc.

Drawbacks from tools introduction

Perhaps the most unfortunate characteristic of PD tools is their potential of delivering a range of side effects to the companies trying to implement them. The following is a list of typically reported drawbacks from tools introduction. This list was assembled during the course of the research project, and is based on interviews and conversations to PD consultants, practitioners and researchers in the field.

- Diverting the attention from the problem or task at hand, to the tool itself.
- Too much time consumed in applying the tool.
- Too little results for so much expectations and effort.
- Increasing the bureaucracy due to the increase in the amount of generated information.

Finally, the following problems where found to lead into sub results in tool utilization:

- Techniques and tools are applied at the wrong point in the development process.
- Successful application is dependent on an optimised execution of other related PD tasks.
- The new tool is promoted by a specific functional department, therefore increasing the functional barriers.
- People within the company have unrealistic expectations towards the tool.
- The technique or tool is not properly implemented.

7 Disposal of tools

Simply stated, the disposal of tools corresponds to the stage in which practitioners will quit using a certain tool. Product Development organisations are highly dynamic entities. What is the best tool one day may simple not fit at all into the organization, the next day. Bellow are listed some reasons that are suggested as being useful to explain the phenomenon of tool disposal:

- The implementation of a new tool, which replaces the old one.
- The introduction of a larger [in scope] tool (such as SAP, PDM, etc) cover the whole range of tasks and problems previously supported by one or more old tools.
- The old task or problem no longer exists (new technology, new process, etc).
- The person that used to sponsor the use of the tool has left the organization.
- The tool does not have a suitable interface with other tools in the organization.
- The tool has been blamed for certain bad results (as shown in the last section).

The act of disposing a tool is quite simple in most cases. The only thing that must be observed is that the users must be convinced that a new, replacing tool (or solution), is in fact being introduced, and that is better than the old one. Failure to do so will likely create dissatisfaction and political problems in the organization.

8 Discussion

In this paper a life cycle interpretation for PD tools was introduced. A survey on the use of PD tools, performed in the UK (Araujo et al, 1996) constituted an important basis for the reasoning about the life cycle aspects of the tools, together with analysis of the available literature, and the visit and discussions with practitioners and researchers. The use of the life-cycle concept shed light on interesting aspects of new PD tools, under a holistic umbrella. It delivered new insights and meaningful information that is believed to be useful both for those people involved in the development of new tools, and to those involved in the selection and use of tools in organisations.

The differences between the way that tools are developed and marketed by academics and professional companies were shown, and the consequences of the differences in the approaches were discussed. In most of the cases the tools developed in the academic means have a lower chance of wining, mainly given the conflicts of the goals involved in their development (scientific proof *Vs* practice worthiness). The tools developed by specialised company have a larger chance of succeeding thanks mainly to the format in which it is presented and sold (more friendly in the views of practitioners). Finally, the tools developed in the industry can be very successful. The reason, again, is their friendliness (language employed, framing situation, etc.)

References

[1] The Adept Group, Ltd. #110 Southpoint Blvd. Suite 230 (Jacksonville, Fl 32216, USA)

[2] Andreasen, M.M. (1996), 'Challenges in Nordic Design Research', *in* Kuuva, M. and Kleimola, M. (eds.), *NordDesign' 96*, Espoo, Finland.

[3] Araujo, C. S. et al., The Utilisation of Product Development Methods: A Survey of UK Industry, *Journal of Engineering Design*, 7(3): pp 265-77, (1996).

[4] Araujo, C. S. *Understanding Acquisition of Product Development Tools in Practice*, Ph.D. Thesis, Technical University of Denmark, Lyngby, Denmark, 2001.

[5] Barkan, P., Benefits and Limitations of Structured Methodologies in Product Design, In: *Management of Design: Engineering and Management Perspectives*, S. D. a. C. Eastman. Massachusetts, Kluwer Academic Publications, 1994.

[6] Cantamessa, M. Investigating Productivity in Engineering Design: A Theoretical and Empirical Perspective, *in* Duffy, A. (ed.), *The Design Productivity Debate*, Strathclyde - Scotland, Springer Verlag, (1997).

[7] Cooper, R.G. *Winning at New Products: Accelerating the Process from Idea to Launch* (2sd Edition), Addison-Wesley Publishing Company, Inc. 1995.

[8] Dimancescu, D. and Dwenger, K. *World Class Product Development* (1st Edition), New York, AMACOM - American Management Association (1996).

[9] Norell, M. Competitive Industrial Product Development Processes - a Multidisciplinary Knowledge Area, *in* Kleimola, M.K.a.M. (ed.), *NordDesign'96*, Espoo, Finland, Helsinki Univ. of Technology - Faculty of Mechanical Engineering, (1996).

Claudiano Sales Araujo
Technology Development Sector (GDT/DTE), Engineering Department, EMBRAER AS
Av. Brigadeiro Faria Lima 2170, ZIP 12227-901, São Jose dos Campos, SP, Brazil
Tel: +55 12345-3458 Fax: +1 309273-0663 Email:claudiano@terra2.com URL: http://i.am/claudiano

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

INTERACTIVE MODEL FOR POWER TRANSFORMER'S CORE DESIGN

D Žeželj, N Bojčetić, and D Marjanović

Keywords: Assembly modelling, computer-aided design, feature-based design

1 Introduction

Production process planning often fails due to the problems arising in the design office. The nature of causes is different but the consequences are more or less the same. Time and money are spent unproductively for redesign and development time is extended needlessly. Such problems have a strong negative impact on company's competivity, market share and revenues. At the same time, design office is a place where the significant amount of product data is generated and reused. Driven by such issues, the need for a computational design tools to support representation and managing engineering information becomes more crucial.

Production of complex investment products such as electrical power transformer is a time-consuming process, mostly consisting of several phases. The contracting and functional design phase take most of the time, in order to capture client's needs and/or specific regional standards. In traditional transformer development concept the embodiment design and manufacturing phases follow in sequential order. In order to shorten the total product development time, most of the phases are now running concurrently, with significant overlaps. The most critical situation arises from the fact that embodiment design phase starts before the project details are fully elaborated with the final contract specification. The design process usually starts with minimum data available. Occasionally, could be a rise of requests for minor or major changes to the basic concept might be requested after design is more or less done. Those changes are often unpredictable, initiating resets in the product development that may cause partial or complete redesign with time consuming revisions in corresponding documentation.

Usage of advanced 3D CAD-systems, which are necessary tools that support the modern product development, will shorten design process time and improve handling of even major changes. However, resolving those changes is often too complicated for novice designers and senior designer involvement is required. Further more, this kind of interventions could be time-consuming and not always totally successful, resulting in rebuilding or restructuring of the product CAD model. Based on this approach, there is an argument to build an application that will help the designer to handle not only simple but even rapid changes in the product design. This is particularly interesting for the products that may be classified as adaptive and variant designs, i.e. the designs where the product functionality, acting principles and structure are within well-known boundaries. This paper describes such an application created for the embodiment design of the power transformer core as the application subtask that will cover design of the whole transformer. Power transformer consists of over 1000 components grouped in 5 main subassemblies: tank, core, tank cap, oil-conservation system and equipment and accessories.

2 System structure

The needs and specifics of the company's traditional design process were established by a series of exhaustive unstructured interviews performed with the senior and novice designers. After the analysis, the system structure was proposed. Figure 1. depicts two major components of the developed system: the 3D product model of fully detailed transformer structure, including relations and constraints, based on the product hierarchy (assembly, subassembly, component), and the application as the second part of the system containing building rules. Such a concept enables designer to manipulate transformer model more easily, relieving him from concern of data and model integrity. It is also extremely important for the new designs derived from existing models by the parametric variations.

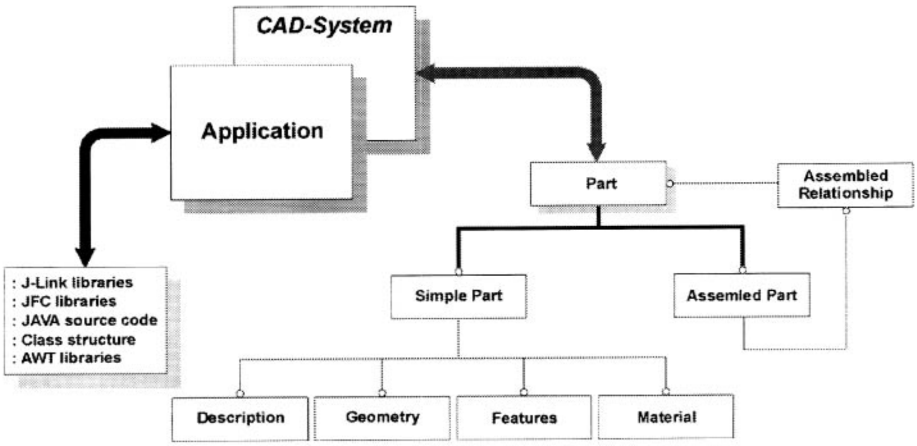


Figure 1. Proposed system structure

2.1 Conceptual CAD model structure

Power transformer core is a complex subassembly with a few variations. However, proposed model should cover only about 85% of possible design solutions in order to avoid loss of time handling special design cases. The goal was not to develop a "design automata" but rather a useful tool that will fulfil the most needs leaving the open door for manual model workouts. The model was built of parts that would optionally change according to design solution (laminations, clamp plates, etc.) and standard parts that are created as elements of common parts database (bolts, nuts, bandages, etc.).

Therefore, the structure of the core was captured through usage of the skeleton parts and feature propagation according to Top-Down [1] design paradigm and Feature Based Modelling [2]. The parametric 3D CAD model was created with extensive use of Family parts and Interchangeable subassemblies. Except feature propagation, which enforces basic design rules, other methods of parts bounding was avoided in order to leave the model in a more suitable state for manual intervention. For every constructive element of the core model main dimensions and design intent were captured and documented. Those data will be fundamental in later stages of system development as well as for the novice designers.

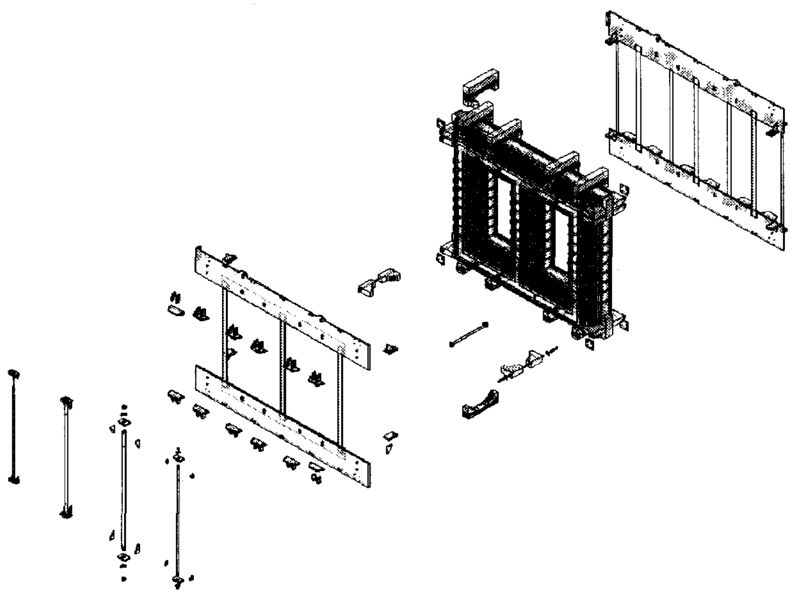


Figure 2. Exploded view of power transformer core
(both horizontal and vertical bolt tying)

2.2 Conceptual application structure

In attempt to fulfil all the goals mentioned, conceptual application structure was created. Because of specificity of CAD-System interface, brought to us as the J.Link libraries [6], JAVA [5] programming language was chosen for the application-building environment. Part data mentioned in earlier section were used for creation of the user interface and the inference engine rules [3][4], which enables control of model dimensions, model material and interaction with other construction parts. Figure 2. depicts the application structure. Three main phases may be seen: the Initialisation phase, the Redesign phase and Documentation. Only the last two phases are obvious to the user, although the first phase is performed on every start-up.

The application building bricks include the following methods:

- StartUp – establishing connection to CAD-System throughout J.Link libraries providing retrieving, changing and deleting data regarding CAD model,
- AppCheck – controlling data integrity on system start-up,
- AppDataAquire – collecting data from initialisation files as well as from CAD model,
- AppDataDisplay – graphical user interface with business rules included, handling events on user change and input,
- JDBC – link to external database or database server,

- ModelCheck – tracking model changes,
- AppUtilities – providing connection to external applications used for calculations,
- AppDataModify – setting up new values, suppressing and resuming parts to create different design solution,
- AppHelp – providing application help and the power transformer specific design knowledge in form of HTML pages.

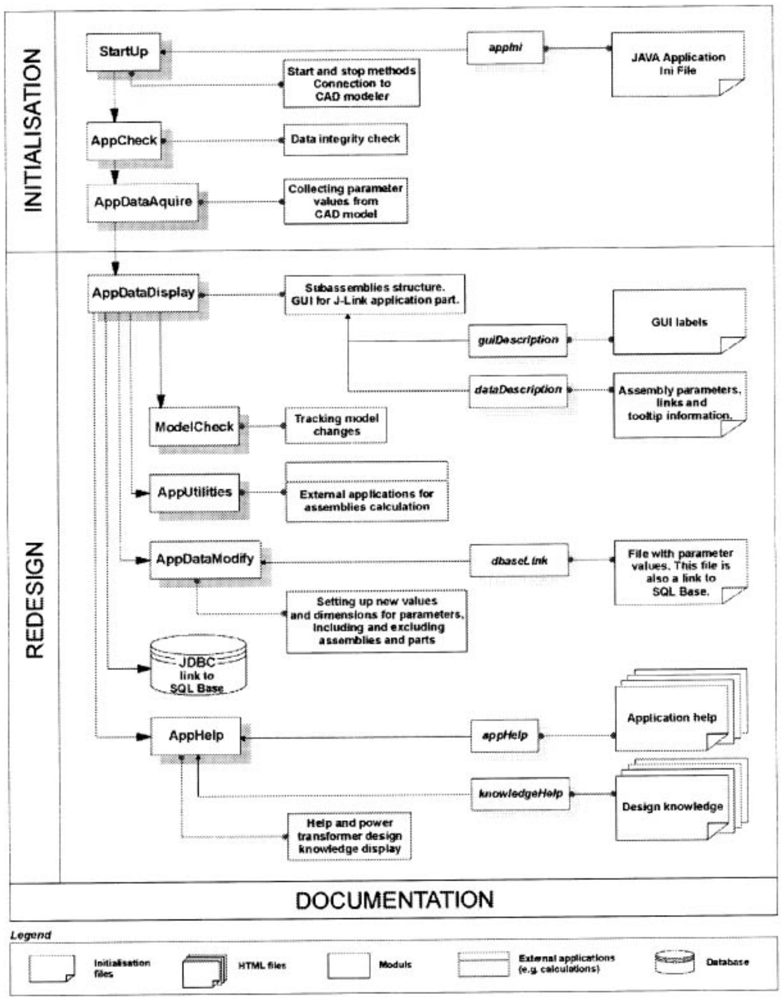


Figure 3. Conceptual application structure

Model data changes, internal constrains and relations are enforced according to the following diagram:

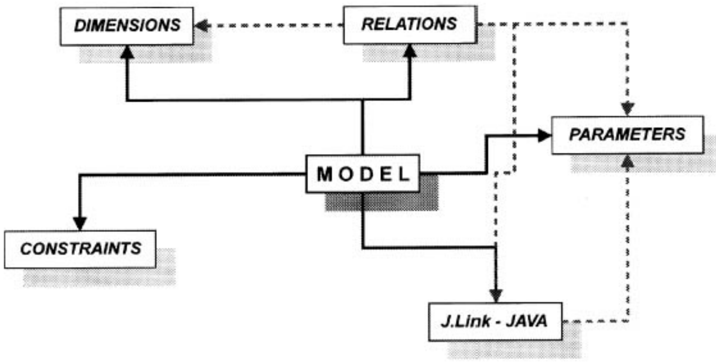


Figure 4. Data changes functionality

Constraints and relations inside parts are enforcing design static rules. Dimensions and parameters could both be changed via application or included in constrains or relations. All dimensions and parameters that should be changeable are listed in initialisation files.

3 Implementation

In this paragraph the description of the implementation issues will be discussed, from the program-structuring viewpoint. The product oriented discussion and the problem relevant for the data exchange in the web environment have been discussed in [7][8]. The Application structure proposed, as well as core model and data changes diagram, are represented in JAVA programming language through power transformer classes.

Each class is responsible for some tasks according to level of hierarchy. Power transformer main class handle start and stop routines, initial set-up reading and application control, detecting of subassembly that is currently active in CAD-system and opening of initialisation files. This class invokes GUI for active subassembly. PTAssembly is base class for derivation of subclasses for each main subassembly (PTAssemblyCore, PTAassemblyTank, etc.). Those derived classes start model check and data change routines, read corresponding data initialisation files and establish links to HTML pages for application and design knowledge.

Each main subassembly has specific data and building parts. Therefore, each of them must have class creating GUI object (DataDisplayCore, DataDisplayTank, etc.). Through those objects designer can communicate with CAD model, via J.Link classes, change data or whole subassemblies and suppress or resume components. As mentioned before, CAD model of the core was built so it could be handled manually. If such intervention occurs, when application starts, it reads the last entered data.

Common classes (lower right corner) are responsible for log files in which all data changes are recorded, system messages and the built in text editor. Such editor enables designer to record notes on fly during design. The principal goal of such editor is to record designers thoughts, or reasoning that led to a particular design solution. Such notes are afterward accessible through web interface.

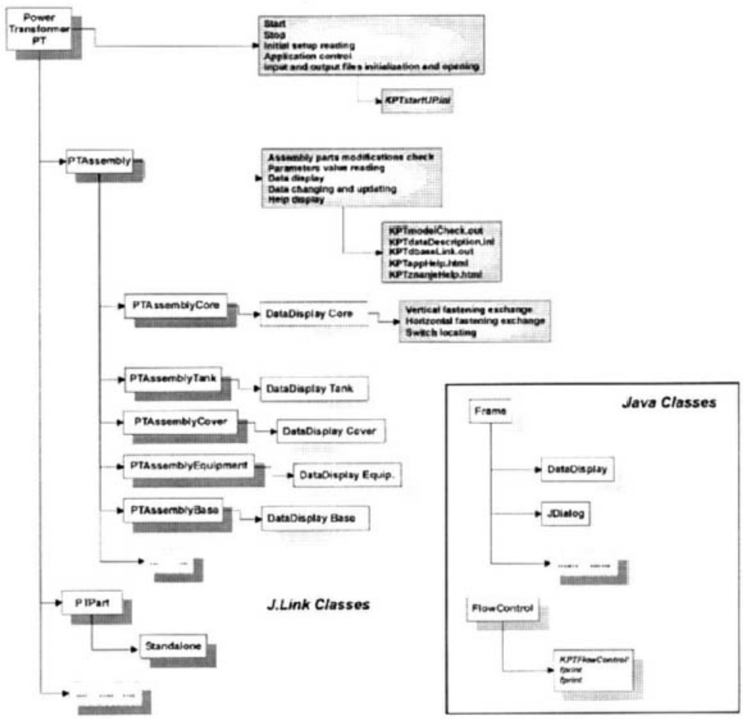


Figure 5. Power transformer's classes structure

4 Example

Upper and lover clamp plates are tied with inner and outer vertical bolts via girders welded on plates. Following pictures are presenting application usage in process of changing instance of inner vertical bolt girder.

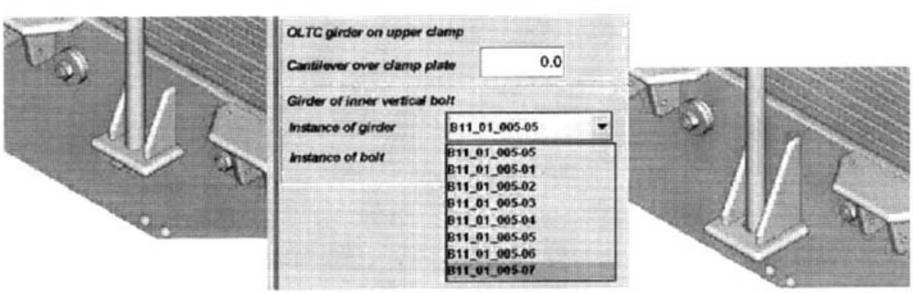


Figure 6. Changing instance of inner vertical bolt girder using combo box with all instances

Power transformer core with vertical bolt tying has 8 girders for inner and as much for outer bolts (4 girders per clamp plate). Application is configured so that the described action would change instances for all 8 girders of inner bolts at the same time. If this task is driven manually this would be the order of actions: change instances for all inner girders, recalculate and change position regarding size of new instance, recalculate and change vertical bolt length. As for many of CAD-systems, changes taken will automatically reflect on existing technical documentation, and any action that could be needed would be simple. Reviewing these changes more closely, this is the sequence of actions that takes place:

- Application is started over the 3D CAD model of the core. During the application initialisation data integrity check is conducted and values from the model collected.

- Parameters data are taken from the initialisation files and used in the GUI creation. Actual values are shown.

- Upon the users action, the application performed calculations for the new design solution regarding only those parameters that were not included in the model static relations and constrains.

- The application regenerated the model.

- The user stops the application.

Step 3 and 4 are representing users interaction with the model through the application GUI. Those steps could be performed as many times as needed until specific design solution is met.

5 Conclusion

This project was conducted in cooperation with Končar Power Transformers, Zagreb, Croatia. A small team of senior and novice designers was established to validate project results performing application tests and analysis in a realistic environment. All team members were familiar with basic 3D CAD concepts only. The novice designers had no previous experience about power transformer design.

The project implementation in the real environment has demonstrated dramatic improvements. At the very beginning of system usage core development time has been shortened by almost 60% and the model correctness or the model integrity was improved considerably.

After five months of system exploitation conceptual model was proven to be suitable for the task given. Time spent in design and detailing of the model of power transformer core was considerably shortened and accuracy of the model increased. Additional benefit of the project is extensive accumulation of design knowledge, which is now integrated in the template model of power transformer core.

Side effects include extensive preparation of data and knowledge gathering that was crucial for system creation.

The Project will be continued to cover other main subassemblies according to power transformer classes structure (fig. 5). Further work will be oriented towards spreading the application over LAN and achieving an on-line database connection.

References

[1] *Pro/Engineer 2000i² Manual Set*

[2] *J. J. Shah, M. Mäntylä, "Parametric and Feature-based CAD/CAM", John Wiley & Sons, Inc., 1995.*

[3] *K. Parsaye, M. Chignell, "Expert Systems for Experts", ", John Wiley & Sons, Inc., 1988.*

[4] P. R. Cohen, E. A. Feigenbaum, "The Handbook of Artificial Intelligence", Addison&Wesley, 1986.

[5] *JAVA Unleashed*

[6] *Pro/Engineer J.Link Manual Set*

[7] Pavlić, D., Marjanović, D., Štorga, M.: "The implementation of web-based technologies in engineering data management"; Proceedings of the 6th International Design Conference DESIGN 2000"; CTT, FSB, WDK; ed. Marjanović, D.; Dubrovnik; Croatia; 2000.

[8] Štorga, M., Pavković, N., Marjanović, D.: "Computer Aided Product Structure Design"; Proceedings of the 6th International Design Conference DESIGN 2000"; CTT, FSB, WDK; ed. Marjanović, D.; Dubrovnik; Croatia; 2000.

Dragan Žeželj, B.Sc.
University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture, of Design, Chair of Machine Elements, I. Lučića 5, 10000 Zagreb, Croatia, +385 1 6168 359, +385 1 6156 940, dragan.zezelj@fsb.hr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

A FEATURE-BASED SYSTEM FOR APPARATUS ENGINEERING CONCEPTS AND SOLUTIONS

M Zirkel, S Vajna, M Jandeleit, and K Strohmeier

Keywords: configuration modelling, constraint-based design, feature-based design, parametric modelling, product data management

1 Introduction

Due to the constantly rising costs and the constantly increasing flood of information the use of computer-aided tools has been established in the development of plants during the last years. Besides CAD systems, predominantly EDM/PDM systems (engineering data and product data management systems) are used. Application areas of computer aided systems during plant engineering are design of piping and connection diagrams, layout planning, the steel structure planning and the simulation of assembly procedures [1]. The emphasis is on the creation of a figurative representation (usually by a 3D-model) of the plant or larger equipment such as pressure vessels, heat exchangers and columns. These tools don't support the detailing in a design or technical point of view. The IT support and, in particular, the integration of suppliers are missing, for example within apparatus engineering.

The design of apparatuses is characterised by a high proportion of the adaptation and the variation of components, which are at a high standardisation level. Completely new designs are rather the exception. New designs or advancements are usually limited to the optimisation of existing components or the use of new materials. With the variety of processes in chemical engineering, almost any apparatus is custom-made. A series production of 3 or more equal apparatuses is very rare [1].

The target of the research project concentrates thereby on the completely computer aided product development and administration. Shorter turn-around times with increased product quality require the intensified application of computer aided systems. At present, the computer support in apparatus design is limited to the use of CAD systems for the description of geometry. However, these systems essentially support the technical designer in the design activity "detail", within the areas of drawing preparation and documentation. For special applications isolated solutions exist, which are in most cases not linked to the CAD applications. For the implementation of a global computer support for all design stages the development of apparatus engineering-specific analysis and structural design methods is necessary. The most promising approach is to base them on a feature-based product model.

2 Feature-based Product Model

The term "feature" was originally linked to manufacturing technology and described geometrical elements carrying information relevant to manufacturing. Today, the understanding of features covers simple geometrical elements, which are stored in a CAD system. On the other hand they are also information carriers, which allow grouping of data going beyond geometric relations. But contemporary commercial CAD systems are limited to modelling only the geometry of the product and its components, generating them from elementary entities (e.g. cylinder, bosses, and pads) [7]. These primitives have no direct link to their respective task within the actual product and the appropriate manufacturing processes. The connection between the geometry of a product and the important non-geometrical information is missing. The basic concept of a feature-based product model is the creation of an application-oriented view on a uniform information model. This model contains all product data during the entire life cycle. A result of the research work of FEMEX in 1996 was a general feature definition that is shown in Figure 1. This theoretical basis describes procedures of feature-based product modelling and a framework for the generation of an application-independent product model. It supports information integration and the optimisation of the product life cycle.

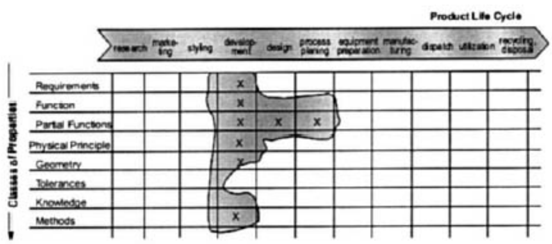


Figure 1. General Feature Definition of FEMEX [2]

The general feature definition describes a feature as an information unit (element) representing a region of interest within a product life cycle. It is described by an aggregation of properties of a product. The description contains the relevant properties including their values, their relations, and their constraints.

2.1 Data Modelling Language EXPRESS

An information modelling language is necessary for the definition of features and the corresponding feature-based product model. Within this project the language EXPRESS (Table 1) and its graphic representation EXPRESS-G are used. It is a specification language (no programming language) for the logical description of information models. EXPRESS is standardised in ISO/10303.

Table 1. Elements of EXPRESS

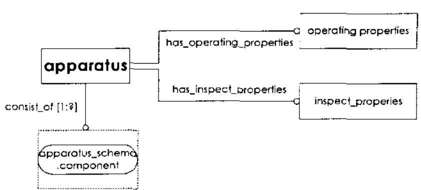
SCHEMA	a wrapper for collections of related information
ENTITY	a definition of an object
ATTRIBUTE	the property of an object
TYPE	a representation of value domains
RULE	for handling of full and partial constraints

It has characteristics that are object-oriented as well as ones, which are defined by the Entity Relationship method. EXPRESS was developed for the formal, consistent, and semantically unique description of product models. With technical progress, changing user demands and new implementation techniques can be supported. The standardised product model structures can be used as starting points for company specific variations. The customer is able to use all advantages of the standardisation. All advantages specified above can also be used independently of the standardisation for the development of arbitrary applications.

2.2 Feature-based Product Model for Apparatus Engineering

For the description of the general apparatus structure, four basic schemata were developed in EXPRESS: the apparatus schema, the connection schema, the general component schema, and the basic schema.

The **apparatus schema** (Figure 2) contains the basic types ‘pressure vessel’, ‘heat exchanger’, ‘column’ and their use. Additionally in these schemas, general characteristics such as operating conditions, test conditions, and the type and number of their components are described.

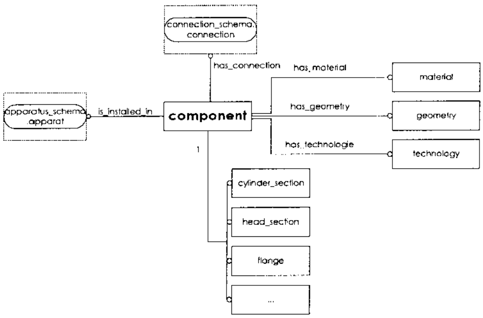



```
SCHEMA apparat_schema;

REFERENCE FROM projekt_schema (projekt);
REFERENCE FROM komponente_schema (komponente,CAD);
REFERENCE FROM verbindung_schema (verbindung);
REFERENCE FROM grund_schema (temperatur, druck, laenge, zahl, punkt, volumen,
bilanz, orientierung, bezeichnung);
REFERENCE FROM druckbehaelter_schema (druckbehaelter);
REFERENCE FROM kolonne_schema (kolonne);
REFERENCE FROM waermetauscher_schema (waermetauscher);

ENTITY apparat ABSTRACT SUPERTYPE OF
        (ONEOF (druckbehaelter, kolonne, waermetauscher));
        behaeltername : bezeichnung;
        hat_betriebsbedingungen : betriebsbedingungen;
        hat_pruefbedingungen : pruefbedingungen;
        hat_komponente : SET [1:?] OF komponente;
        hat_verbindung : SET [1:?] OF verbindung;
        gehoert_zu_projekt : projekt;
        hat_plazierung : plazierung;
        gesamtlaenge : laenge;
        nenndurchmesser : laenge;
        wandstaerke : laenge;
        nenninhalt : volumen;
        AutoCAD : CAD;
        SolidEdge : CAD;
END_ENTITY;

ENTITY betriebsbedingungen;
        betriebstemperatur : temperatur;
        umgebungstemperatur : temperatur;
        betriebsdruck : druck;
        umgebungsdruck : druck;
        stoffbilanz : bilanz;
        waermebilanz : bilanz;
END_ENTITY;
```

Figure 2. Extract of Apparatus Schema in EXPRESS-G and EXPRESS

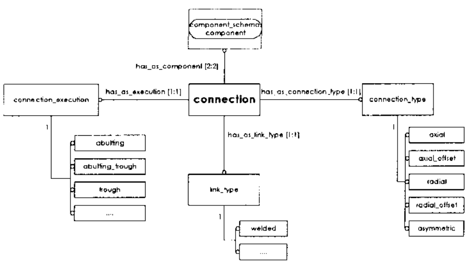
The **component schema** (Figure 3) contains all general information, which is needed for the description of the apparatus elements from the point of view of the designer. Each component needs information about material, technology, usage, and the apparatus they are built into. Additionally the component schema handles the information about the connection between the components. In particular it refers to the type schema, in which individual versions are defined such as connecting pieces, flanges, coats, etc. and their specific geometry sizes.

In apparatus engineering the connections of components are very important from a physical and also from a geometrical point of view. The characteristics of a connection are defined in the **connection schema** (Figure 4), which was introduced as abstract object "connection". This object enables both the organisation and the calculation of the different connections, without having to consider the whole apparatus. All recurring sizes such as dimensions, printing quantities etc. are defined in the **basic schema**. The basic schema is referred by each of the above-mentioned schemas.




```
SCHEMA komponente_schema;

REFERENCE FROM apparat_schema (apparat);

REFERENCE FROM grund_schema (laenge, punkt, bezeichnung, zahl, spannung);

REFERENCE FROM zylinder_schema (zylinder);

REFERENCE FROM kugel_schema (kugel);

REFERENCE FROM konus_schema (konus);

REFERENCE FROM kegel_schema (kegel);

REFERENCE FROM boden_schema (boden);

REFERENCE FROM platte_schema (platte);

REFERENCE FROM dichtung_schema (dichtung);

REFERENCE FROM balgkompensator_schema (balgkompensator);

REFERENCE FROM tragelement_schema (tragelement);

REFERENCE FROM ausschnitt_schema (ausschnitt);

REFERENCE FROM Flansch_schema (flansch_b8);

ENTITY komponente ABSTRACT SUPERTYPE OF
        (ONEOF (zylinder, kugel, konus, kegel, boden, platte, dichtung,
        balgkompensator, tragelement,flansch_b8));
        komponentenname : bezeichnung;
        hat_grundtyp : bezeichnung;
        hat_werkstoff : werkstoff;
        hat_technologie : technologie;
        ist_eingebaut_in : apparat;
        hat_verwendung : verwendung;
        hat_ausschnitt : OPTIONAL SET [1:?] OF ausschnitt;
        AnzahlEndflaechen : zahl;
        AutoCAD : CAD;
        SolidEdge : CAD;
END_ENTITY;
```

Figure 3. Extract of Component Schema in EXPRESS-G and EXPRESS




```
SCHEMA verbindung_schema;

REFERENCE FROM apparat_schema (apparat);

REFERENCE FROM komponente_schema (komponente);

REFERENCE FROM grund_schema (nummer, punkt, laenge, winkel, bezeichnung);

REFERENCE FROM ausschnitt_schema (ausschnitt);

ENTITY verbindung ABSTRACT SUPERTYPE OF
        (ONEOF (grundkoerper_verbindung, stutzenverbindung, anbau_verbindung,
einbau_verbindung));
        verbindungsname : bezeichnung;
        gehoert_zu : apparat;
        anschlussbauteil : komponente;
        anzuschliessendes_bauteil : komponente;
        endflaeche_anzuschliessendes_bauteil : nummer;
        hat_als_verbindungsart : verbindungsart;
        anschlussrichtung : ausrichtung;
END_ENTITY;

ENTITY grundkoerper_verbindung SUPERTYPE OF
        (ONEOF (koaxiale_grundkoerper_verbindung, axiale_grundkoerper_verbindung))
        SUBTYPE OF (verbindung);
        endflaeche_anschlussbauteil : nummer;
END_ENTITY;
```

Figure 4. Extract of Connection Schema in EXPRESS-G and EXPRESS

3 Architecture of Feature-based Systems

A feature-based system supports all phases of the product life cycle. The features are "containers" of information with references to these different phases. They store the properties of the product (also geometric information) as well as their relations. Properties and their relations are an essential part of feature-based models, because they are necessary for the description of the technical significance of a product. Constraints are included in the system, e. g. dependencies between parameter values and the spatial configuration of components.

Special editors allow the modification of the features and the corresponding product model. So it will be possible to add and to modify the features according to individual demands. The architecture of a feature-based system is characterised by independent modules that rely upon a common product model.

3.1 Architecture of the Feature-based System IKA

For a global computer support a methodical procedure is essential. For this reason the apparatus engineering process was analysed. A generally accepted methodology and the corresponding computer aided methodology were developed. Based on these procedures the appropriate design modules were derived and implementation concepts designed. Thus an iterative working procedure during the entire design process can be supported.

The feature-based product model described in 2.2 is the basis of the integrated design system IKA (**I**ntegriertes **K**onstruktionssystem für den **A**pparatebau - integrated design system for apparatus engineering). Its principle structure is represented in Figure 5.

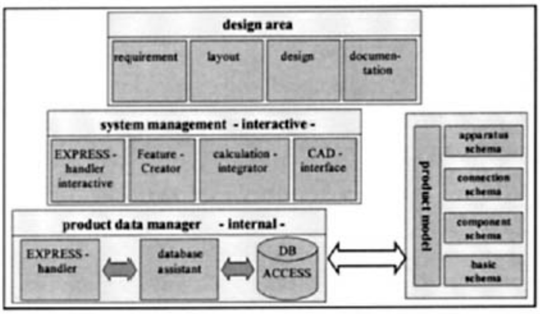


Figure 5. Principle structure of the design system IKA

In order to implement an open and easily expandable system, the design system was conceived as development environment. So it is possible to add design-relevant information to the product model as it becomes available during the design process. Thus the model becomes more and more complete as the process is progressing. The product model based on EXPRESS schemes is administrated by an internal product data manager, which can't be edited by the user. The implementation of the product model is performed via the development of a database, which is adapted dynamically to the modifications of the product model. The interactive system administration provides tools for the manipulation of the product model, the integration of special tools, and the link to CAD systems. The user may integrate company-specific modules using provided developed integration tools. The design space, which is generated dynamically, is customisable by a tool called feature-creator. This tool will be described later on. When the design system is customised by the system administration, the designer is guided trough the different design stages.

3.2 Integration Tools and Modules

For the development of the integrated design system IKA, tools were implemented that allow on one hand the connection of application and data models and on the other hand the extension of the system. For the implementation of an open and expandable system an EXPRESS handler was developed. It can edit and extend the different definitions and dependencies of the entities defined in the product model. Figure 6 shows the principle functionality of this tool. This tool also allows the individual configuration of the input and output dialogues during the different design stages. The appropriate data base fields will be automatically created or manipulated. The main system can be adapted without modifications of the source code by a dynamic manipulation of the product model.

Figure 6. EXPRESS handler

For the support of the early design stages IKA provides a dynamic 2D tool, which is shown in Figure 7. The toolbars are generated automatically using the definitions in the product model. The designer is able to create a tentative draft and to add information to the components.

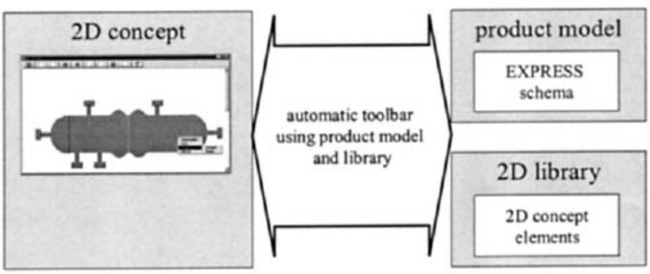


Figure 7. 2D conceptual design

Company-specific and commercial calculation modules can be integrated into the design system with the help of the calculation integrator represented in Figure 8. The modules are in form of dynamic link libraries, so-called DLL, which are dynamically loaded and unloaded during run-time. Only the parameters that are needed for the calculation and their sequence must be selected. If this selection is kept, the system-oriented integration is performed automatically.

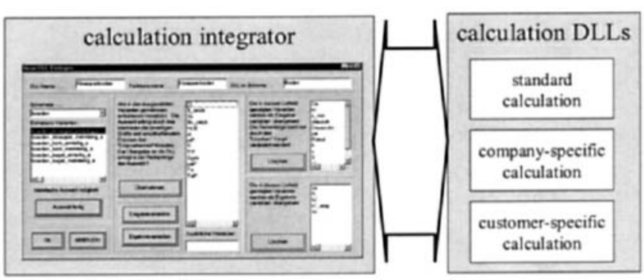


Figure 8. Calculation integrator

For the implementation of the CAD integration, commercial CAD systems are used, whereby the design system IKA controls the CAD system (Figure 9). The employment of the CAD system's functionalities offers all advantages for an extension of the design system. New components or new component versions can easily added or edited. Using the feature creator these variants can be added into IKA.

The apparatus components were created as parametric 3D-models, which contain the parameter designations according to the EXPRESS definitions. When a new version of a component is defined in IKA, the parameters are transferred to the CAD system via the system's application interface.

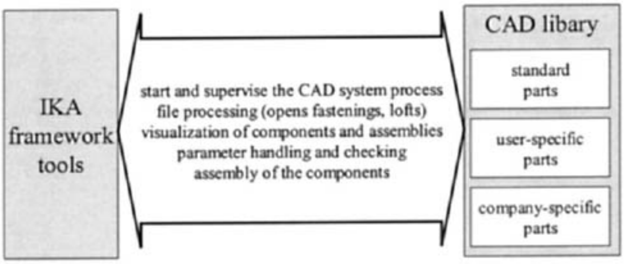


Figure 9. CAD-Interface

The generated CAD model can be checked visually. The same procedure is applied for the assembly of the components. Figure 10 in the summary shows some examples of apparatus components, different connecting types as well as assemblies that were generated by IKA.

4 Summary

The aim of the project is the optimisation of the apparatus design by the development of an integrated computer support on the basis of a feature-based product model for apparatus engineering. This includes the development of computer-aided methods for apparatus engineering and the corresponding tools and the integration of the partial solutions into commercial CAD systems. The result of the research work is the design system IKA (Figure 10), which enables an integrated feature-based computer support for apparatus engineering. The emphasis was the development of a generally accepted methodology and the supply of application-independent tools that enable the integration of company-specific modules into commercial CAD systems.

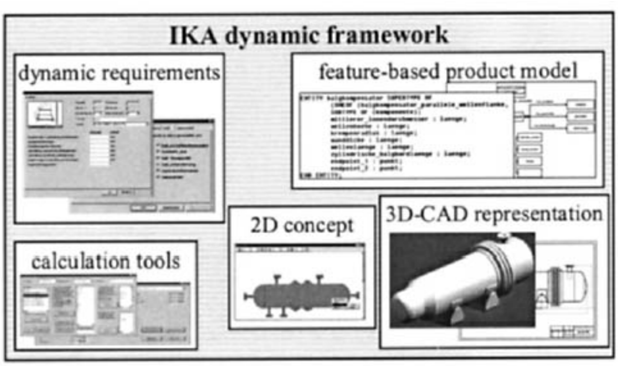


Figure 10. IKA dynamic framework

Due to the separation of application and data model the system can be adapted easily to user specific requirements without the need of re-programming the entire system. The technical designers get the possibility to customise the used environment with extended functionalities. An adjustment of the system is enabled by the application of features to application-oriented requirements. Thus the design result is optimised. With the help of the feature technology, all areas can use the information on development results and the data developed during the design process. That way possible errors and problems can be detected earlier then now, they can already be corrected before the manufacturing of the apparatus or their components starts.

References

[1] Dittrich, T., "Rechnergestützte Konstruktion im Apparate- und Anlagenbau", VDI Reihe 20 Nr. 179, VDI-Verlag, Düsseldorf 1995

[2] Vajna, S., Wegner, B., "Features - Information Carriers For The Product Creation Process", Proceedings American Society of Mechanical Engineers, Sacramento 1997

[3] Vajna, S., "Stand und Tendenzen der rechnerintegriertenAnlagenplanung", Tagungshandbuch 19. Konstruktions- Symposium der DECHEMA e. V., Frankfurt Main 1999

[4] Jandeleit, M.Strohmeier, K., " Integrierte Konstruktionssysteme im Anlagenbau – ein Konzept, Konstruktion 9/2000", Konstruktion 9/2000, Springer-VDI-Verlag 2000

[5] VDI 2218, " Feature-Technologie ", VDI-Handbuch Konstruktion, Beuth Verlag Berlin, 1999

[6] Bugow, R., "EXPRESS, das universelle Werkzeug", Produktdaten Journal, ProSTEP e.V., Juli 1997

[7] Shah, J., Mäntyla, M., "Parametric and Feature-Based CAD/CAM", John Wiley & Sons, Inc. 1995

Dipl.-Ing. M. Zirkel
Computer Applications in Mechanical Engineering, Institute of Mechanical Design, Otto-von-Guericke-University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany, Tel: + 49 - 391 - 67 – 18099, Fax: + 49 - 391 - 67 – 11167, E-mail: Manfred.Zirkel@mb.uni-magdeburg.de, URL: http://imk.uni-magdeburg.de/LMI/index.htm

Prof. Dr.-Ing. S. Vajna
Computer Applications in Mechanical Engineering, Institute of Mechanical Design, Otto-von-Guericke-University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg, Germany, Tel: + 49 - 391 - 67 – 18794, Fax: + 49 - 391 - 67 – 11167, E-mail: vajna@mb.uni-magdeburg.de, URL: http://imk.uni-magdeburg.de/LMI/index.htm

Dipl.-Ing. Markus Jandeleit
Institute of Pressure Vessels and Plant Design, Technical University Munich, Boltzmannstr. 15, 85748 Garching bei München, Germany, Tel: + 49 - 89 - 15697, Fax: + 49 - 89 - 15694, E-mail: m.jandeleit@laa.mw.tum.de, URL: http://www.laa.mw.tu-muenchen.de

Prof. Dr.-Ing. K. Strohmeier
Institute of Pressure Vessels and Plant Design, Technical University Munich, Boltzmannstr. 15, 85748 Garching bei München, Germany, Tel: + 49 - 89 - 15696, Fax: + 49 - 89 - 15694, URL: http://www.laa.mw.tu-muenchen.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

FEATURE EXTRACTION TOOLS ON POLYHEDRAL MODELS FOR MACHINING APPLICATIONS

M Labrousse and P Veron

Keywords: Design for manufacturing, form feature extraction, polyhedral models, reverse engineering

1 Introduction

For reverse engineering[1] applications, polyhedral models are widely used to describe object geometries from clouds of digitised points. The use of non contact optical measuring systems, or of co-ordinate measuring machines produces a large number of 3D points with an accuracy of up to 10 μm (see [1] for an overview of reverse engineering approaches). Unfortunately, in such reverse engineering processes, only geometrical information is obtained and technological information such as diameters of holes, machining tolerances and form features are lost. Consequently, the machining software generates tool paths which are not optimised according to the form of the part to be manufactured. The knowledge of the main form features of the object, from the machining point of view (like holes for example), should improve the quality of the trajectories, and reduce drastically the machining time needed by gathering together similar operations (drilling of holes with the same diameter, for example).

This paper describes new feature extraction tools working on a polyhedral model of a part. Then, the use of the features extracted to improve machining strategy is presented. The development of these tools are currently limited to the extraction of cylindrical features.

2 Extraction of cylindrical form features from the polyhedral model

2.1 Simplification of polyhedral models

The digitisation process produces a very large number of 3D points, particularly when non contact optical measuring techniques are used. The number of points can be more than 10 million, thus producing very large files difficult to process.

In order to obtain good results in less time, the polyhedral model build from a digitised cloud of 3D points is first simplified using algorithms previously developed by P. VERON [2] (Fig. 1). During the simplification, a geometrical criterion checks that the distance between the initial and the simplified polyhedron remains below a user defined value. When this value is chosen near the accuracy value of the sensor, the influence of measuring noise on the geometry is reduced.

[1] Reverse engineering consists of creating a CAD model from an existing physical object whose geometrical or technical information are unavailable in digital form.

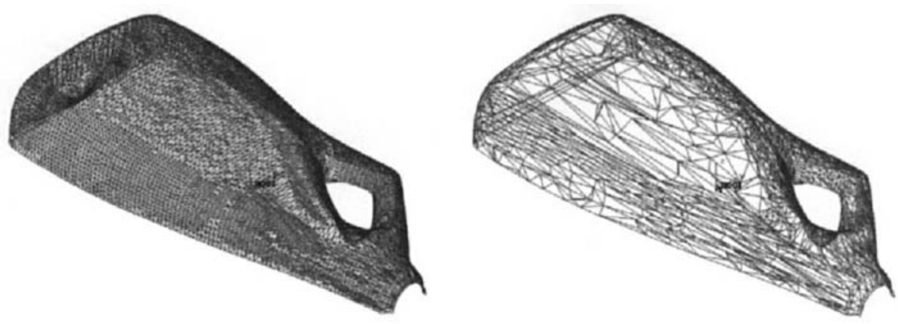


Figure 1: Simplification of the polyhedral model.

2.2 Identification of the "sharp lines" of the object shape

The approach proposed for extracting cylindrical form features begins with the identification of the "sharp lines" of the part (Fig. 2).

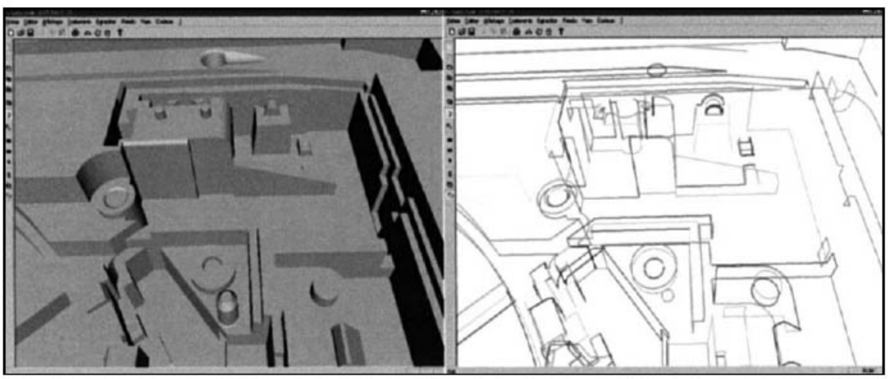


Figure 2: "Sharp lines" identification step.

To this end, because real surface curvature properties are not defined on a polyhedral model, specific discrete curvature approximation criteria have been developed [3].

It should be underlined that the computation is based on the properties of the triangle: the surface isn't locally approximated by parametric quadric surfaces as in [4]. The disadvantage of such surfaces is that they couldn't represent correctly tangency discontinuities, thus producing a harmful smoothing. For this reason, A. Meyer and P. Marin [5] propose a very particular surface with a tangency discontinuity, the "absoïd". Unfortunately, the computational time becomes then very long, making this method unusable on dense industrial digitised data.

The "sharp lines" algorithm developed is based on a propagation process of lines along points of high discrete curvature values (Fig. 3). The propagation stops when the curvature value is below a threshold specified by the user.

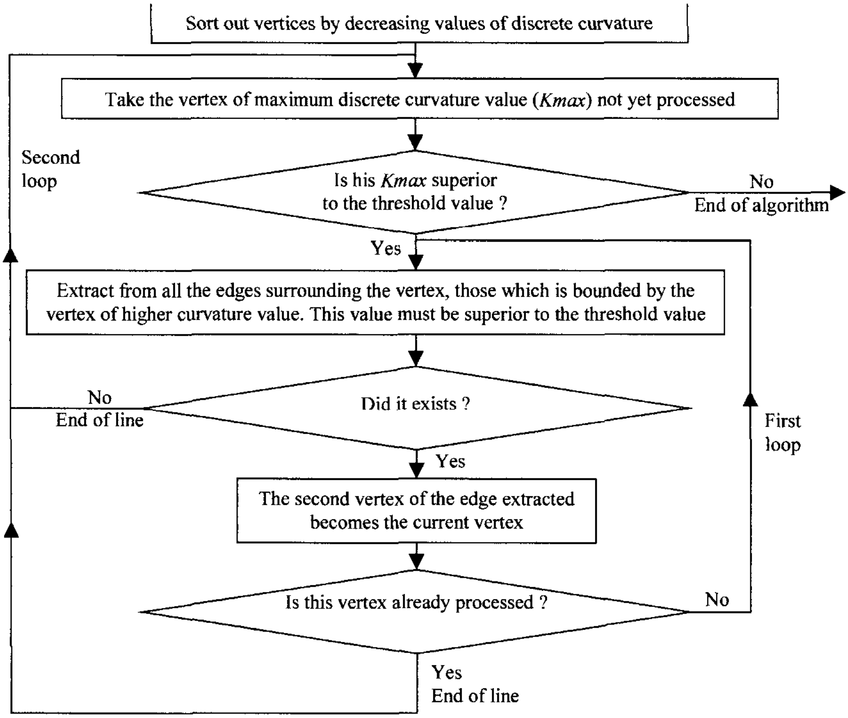


Figure 3: "sharp line" identification algorithm.

2.3 Association of elliptic lines in order to detect cylindrical areas

Remark : our algorithm uses directly three dimensional data. The computation of planes sections of the part as in [6] or in [7] is not needed.

After the "sharp lines" are extracted, a criterion is used to identify closed lines which are nearly elliptical. Only these lines are kept for the cylindrical feature extraction step. Indeed, these lines are supposed to be potential extremities of cylindrical areas, due to the fact that, for mechanical parts, holes are generally continuously, but not tangentially, connected to another area of the object. The lines which seems to belong to the same cylindrical area are then associated. To this end, for two candidate lines, the criterion used is based on the comparison of their perimeters, their relative orientation and their relative distance. When two lines are matched, the theoretical axis of the cylinder (a point and a vector) is computed, and its radius is approximated. This information will help to determine if a facet belongs to the cylinder being detected or not.

The next step consists of the detection of the facets which belong to the cylinder defined by the two lines matched. A propagation process is initialised from all the triangles included in the area between the two lines and containing an edge of the first one (Fig. 4.a). The front is then propagated in the direction of the second line by an iterative process (Fig. 4.b and c).

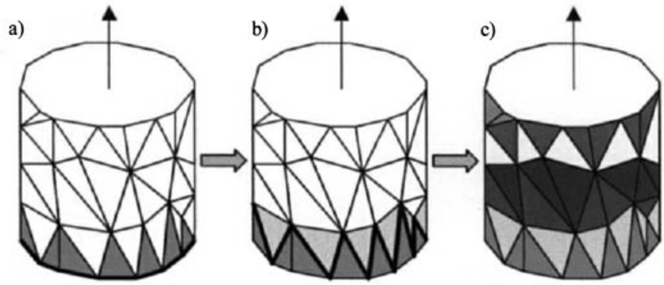
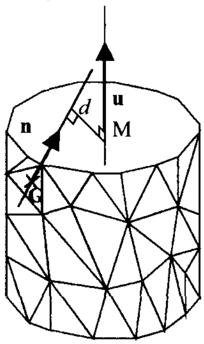


Figure 4: Detection of the cylindrical areas based on a front propagation process.

The algorithm stops when the second line is reached, or when the facets are not well oriented or are too far from the theoretical position of the cylinder (Fig. 5).



The triangle is rejected when :

- ***(n . u)*** *< threshold value 1*

or

- *d < threshold value 2*

Figure 5: Cases of triangle rejection in the detection process.

When two lines can't be associated, a specific algorithm makes the detection in a similar way, but it only use the knowledge of one line. The missing information is estimated by the algorithm : the axis as a first approximation is replaced by the normal of the line, and a pre-detection will take place. The axis of the cylinder is then computed and, if it differs too much from it first approximation, the detection will be carried out again with the improved parameters.

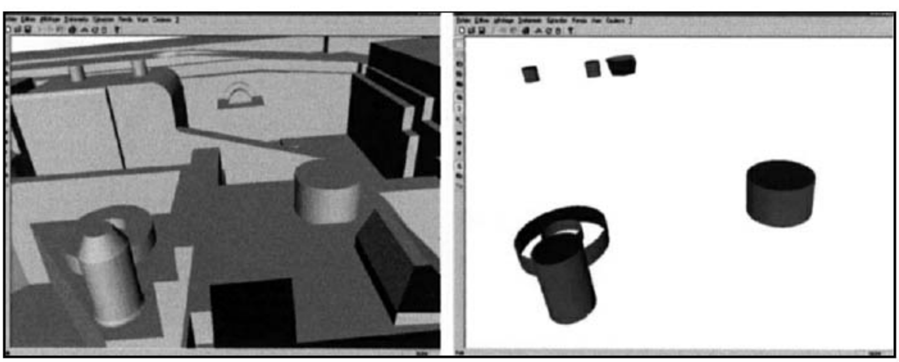


Figure 6: Example of cylinder extraction.

An example of cylinder extraction can be seen on Fig. 6. It should be underlined that the cylinder in foreground is well detected thought there is a chamfered edge and a transition radius.

3 Improvement of the machining strategy

The final objective of these tools is to optimise the tool path calculation for machining. To this end, specific information is needed and extracted from the identified features : the radius of the drill, its axis, the depth of drilling and the type of hole (through all or not).

Then, holes of the same diameter and/or which have the same accessibility can be grouped to improve the machining strategy and thus reduce manufacturing time.

3.1 Extraction of specific information from the identified features

The computation of specific information is very important for machining : the axis of the hole to define the direction of the drilling tool, the radius of the cylinder to define the radius of the drill, the length of the cylinder to define the length of the drill, etc.

1. *Extraction of the axis of the cylinder*

The cylinder is defined by two outlines. The axis can be obtained by joining the centres of the outlines (Fig. 7).

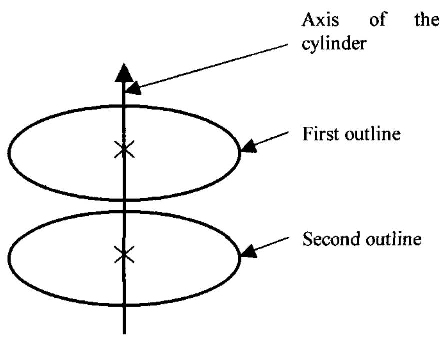


Figure 7: Computation of the axis of the cylinder.

To improve the quality, planes intersecting with the detected cylinder can be computed. The axis is then obtained from all the centres defined by calculation using the method of least square.

2. *Extraction of the radius of the cylinder*

The radius of the cylinder is obtained by computing the average of the distance from the barycentre of each triangle to the axis of the cylinder. The average is level-headed by the area Ai of the triangles.

The distance *di* from a point Mi to the axis (A , **u**) is done by : $di = || \boldsymbol{AMi} \wedge \boldsymbol{u} ||$

The area of the triangle (ABC) is found by : $Ai = ½ \| \mathbf{AB} \wedge \mathbf{AC} \|$

The average of the radius is then :

$$R = \frac{\sum_i Ai^*di}{\sum_i Ai}$$

The maximum distance between the average radius and the distances *di* is also computed. It can be used to decide if the detection of the cylinder was correct : *(max (R-Ri))/R* must be very low.

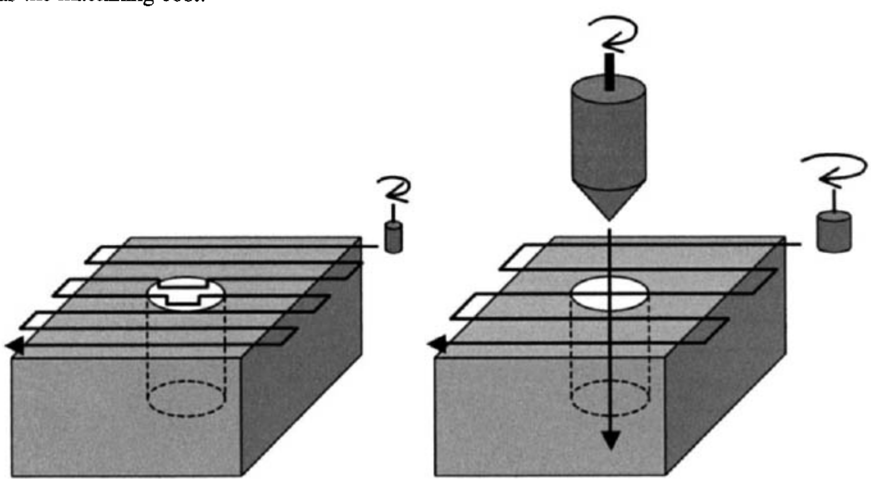
3. *Extraction of the length of the cylinder*

Each point belonging to the cylinder is projected on the axis. The length of the cylinder is then defined as the maximum distance between the projected points.

3.2 Improvement of the machining strategy by optimising tool paths

The knowledge of a particular feature enable the use of tools adapted to the object to be machined. For example, cylinders could be easily obtained by drilling.

With feature extraction, the tools have a specific usage. Their size is in many case much bigger than in the case of "blind" machining (Fig. 8). The machining time is reduced as well as the machining cost.



a) Without feature extraction, small tool is required to cut the form.

b) With feature extraction, specific tool adapted to the forms is used. Moreover, the quality is improved.

Figure 8: Machining with or without feature extraction.

3.3 Improvement of the machining strategy by gathering similar information

Only one change of tool is needed for each type of extracted feature : the holes with the same diameter can be drilled in a single step, thus decreasing drastically the machining time. The features can be easily sorted by criteria such as same diameter, same orientation, etc.

The information can also be used to detect high level features such as the type of motor fixings, or of standard parts often used by the company (Fig. 9). Specific machining strategies for such manufacturing operations can be stored in a database and be highly optimised. The required time will be lower, with a higher quality. The digitised model can also be modified easier, for example if the motor fixing is changed.

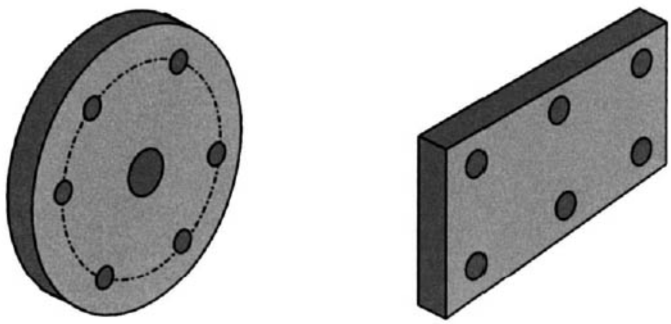


Figure 9: Example of high level features.

3.4 Improvement of high speed machining

Feature extraction enables the improvement of the tool paths in high speed machining processes. In fact, for mechanical parts, holes (as well as other geometrical features) are generally continuously, but not tangentially, connected to an other area of the object. The discontinuity in direction of the tool paths will imply a very low machining quality, particularly in the case of high speed machining : the change of direction can't take place immediately, because the acceleration can't be infinite. For example, if the speed is tenth the speed of that used in classical machining, the kinetic energy become one hundred times higher. If the deceleration must take the same time, the stresses must be multiplied by 100! For this reason, changes of direction have to be avoided : they stress components, increase the risk of tool or tool-holder breakage, and reduce the quality of the result. Feature extraction will help to do this : sharp edges are the result of trajectory intersection and are not the result of the trajectory itself.

4 Conclusion

The algorithm has proved its ability on industrial examples. Processing time is low, and only a few interactive interventions of the user are needed.

Information directly usable for machining applications are produced. This makes possible the best choice of cutting tools, the improvement of tool paths, particularly in high speed machining processes. They reduce also the machining time and therefore the cost of machining.

The current developments are restricted to cylindrical features. Future work will focus on the extraction of other form features like cones and prismatic surfaces. New criteria have to be developed to take advantage of invariance properties (in translation and/or rotation) of each considered surface.

References

[1] Motavali, S., "Review of Reverse Engineering Approaches", Computers industrial Engineering, 35, 1998, pp 25-28.

[2] Véron, Ph. and Léon, J-C., "Shape preserving polyhedral simplification with bounded error", Computer & Graphics, 22, n°5, 1998, pp. 565-585.

[3] Véron, Ph., Lesage, D. and Léon, J-C., "Outils de base pour l'extraction de caractéristiques de surfaces numérisées", Revue Internationale de CFAO et d'Informatique Graphique, 13, n°4-5-6, 1998.

[4] Yang, M. and Lee, E., "Segmentation of measured point data using a parametric quadric surface approximation", Computer-Aided-Design, 31, 1999, pp. 449-457.

[5] Meyer, A. and Marin, P., "Local properties determination and edges identification of digitised surface", IDMME'2000 conference, Montreal, Canada, May 2000.

[6] Prades, G., Djiguiba, A., Caenen, J.L., Mineur , Y., Castelain, J.M., "Segmentation of Measured Data Points from Orthogonal Cross Sections Extraction and Analysis", IDMME'2000 conference, Montreal, Canada, May 2000.

[7] Liu, S. and Ma, W., "Seed-growing segmentation of 3-D surfaces from CT-contour data", Computer-Aided Design, 31, 1999, pp. 517-536.

Dr. Philippe VERON, Eng. Michel LABROUSSE
Ecole Nationale Supérieure d'Arts et Métiers, Centre d'Etude et Recherche d'Aix-en-Provence, Equipe Ingénierie, Mécanique, Systèmes, 2 cours des Arts et Métiers, 13617 Aix-en-Provence, France, Tel: 33 442 938 124, Fax: 33 442 938 196, E-mail: Philippe.Veron@aix.ensam.fr, Michel.Labrousse@m6net.fr

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

EVALUATION AND GENERATION OF ATTACHMENT CONCEPTS BASED UPON A FEATURE-BASED SOLID MODEL USING A FEATURE-BASED RECOGNITION STRATEGY

D H Baxter and G A Gabriele

Keywords: Integral attachment, feature based recognition, embodiment design methodology

1 Introduction

Previous research [1] into introducing more rigor in the use of *integral attachment* in mechanical design (often called "snap-fits") was pursued on two levels, development of design methods to assist the designer in defining optimal assemblies using integral attachment features, and research aimed at understanding the performance of common integral features[2][3]. The first of these two areas, called *Integral Attachment Design*, led to a well-defined method for classifying each design situation into one of several finite cases with associated potential design solutions. This method also became a means of exploring and ranking design alternatives. This quickly leads to exposing potential optimal designs for integrally attached assemblies.

1.1 Research Objective

The thesis of this work is that the methodology defined for Integral Attachment Design can be automated by integrating existing methodologies with Computer Aided Engineering (CAE) tools; specifically with Computer Aided Design (CAD) tools. The greatest leverage appears to be in the area of feature based recognition strategies. For example, Computer Aided Manufacturing (CAM) has made several breakthroughs in feature recognition for tooling in a CAD system [4]. The goal of this research is to provide a similar integration of CAD and assembly feature design at the Embodiment Design stage. The intent is to provide the designer with an opportunity to explore attachment concepts early in the design work, within the environment of a feature based solid modeler. By integrating the integral attachment design method within the CAD environment, design solutions could be suggested to the designer for further exploration.

2 Overview of the Present Integral Attachment Design Method

The present design method consists of 6 steps: (1) identification of parts (or part classification); (2) identification of part combinations; (3) identification of applicable assembly procedures; (4) identification and selection of locking features; (5) kinematic constraining of parts and load bearing capacity of attachment features; and, finally, (6) evaluation of and selection among alternatives. An overview of the process is shown in Figure 1. Details can be found in [5] [6] [7] [8] [9] [10].

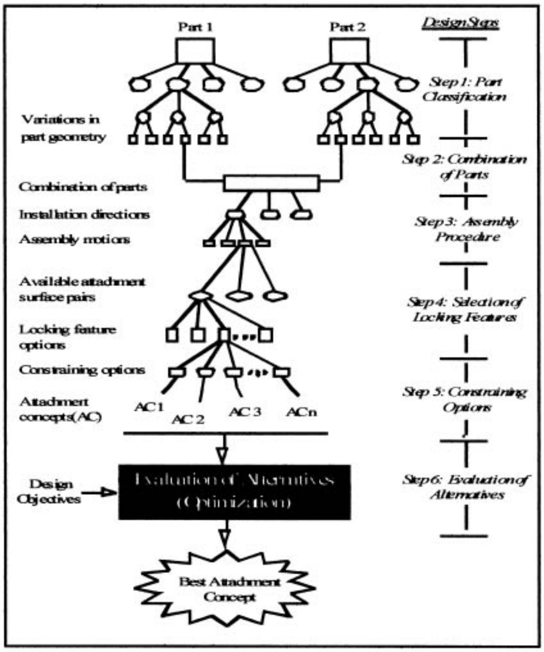


Figure 1. Integral Fastening Design Method

Alternative attachment concepts generation starts with identification of the basic or essential geometry of attachment surfaces and final part arrangement in the assembly, then continues with identification of possible assembly procedures, and ends with alternative attachment feature selection and placement. In this manner, attachment concepts evolve as roots of a tree, starting from a base and moving downward and outward with details.

Assembly procedure consists of installation directions and assembly motions. An installation direction is the primary axis that brings two parts together. In this paper, an installation direction is a vector defined in terms of the assembly's global coordinate system. Assembly motion refers to the final assembly procedure.

The focus of this paper is the determination of the available attachment surface pairs and the possible assembly procedures (Step 3) from data obtained from the solid models of the parts and the assembly. Within the solid model, the designer has placed information concerning the assembly. Specifically, the number of surfaces in contact between the two parts, the size of the surfaces, and their relationship to each other are all available. While visual inspection can be used to help determine the attachment surface pairs and assembly procedure, this research will show that it is often possible to obtain this data directly from the solid model in a precise mathematical manner. The results of the analysis will not always produce a single solution but, rather, a set of possible solutions given the geometry of the two assembled parts. [2] [3].

3 Integrated Design Tool

3.1 Process Flow

To determine the attachment surface pairs and possible assembly procedures, the integrated CAD fastener design tool works along two parallel paths as shown in Figure 2.

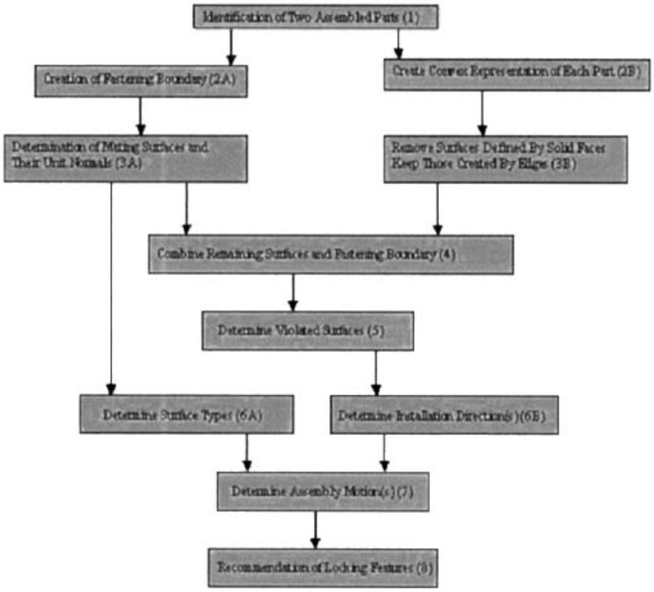


Figure 2 Design Tool Flowchart

Identification of Parts for Attachment (Step 1)

The new design tool begins with input from the designer. It is assumed that the designer has created reasonable representation of the parts and the assembly. The parts and assembly are expected to be embodiment designs; embodiment designs being created after the concept design and are used to determine the details of the design. While the parts and assembly should capture the intent of the design and some of the more prominent features, many of the fine details (such as fillets, draft angles, etc.) may not be known [11]. An example showing three plastic parts is shown in Figure 3. The television remote consists of an upper and lower case; these parts are assumed to be already assembled and thus can be considered a single entity for this exercise; the designer intends the upper and lower case to be assembled first. The third part is the battery cover; this example will explore how the battery cover can be fastened to the case. With the selection of the two assembled parts, the design tool will examine the two selected parts to create the fastening boundary.

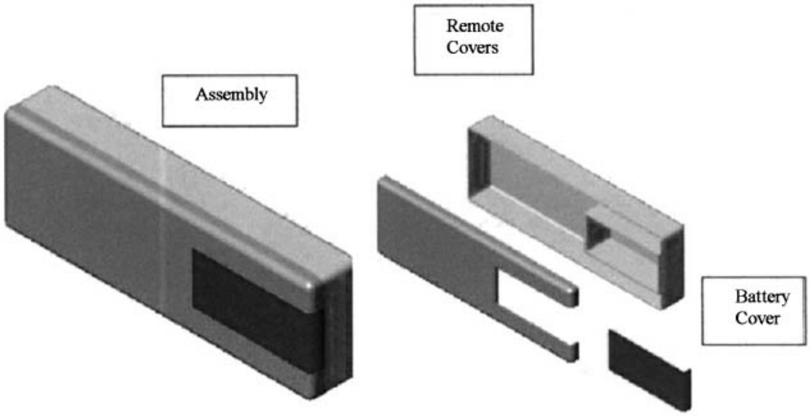


Figure 3. Television Remote Control Assembly

Determination of Attachment Surfaces and Their Unit Normals: Fastening Boundary (Steps 2A and 3A).

From the assembly, the fastening boundary can be created [12]. The fastening boundary is defined as the collection of mating surfaces and edges between the two assembled parts[1]. The design tool first checks for overlapping material between the two parts. Assuming the assembly is valid (that is, material from the two parts do not share the same space), all coincident edges and surfaces are found. These edges and surfaces are then used to create a new surface model, the fastening boundary, as shown in Figure 4. In the simplest case, where two parts butt together, the fastening boundary is comprised of a single surface.

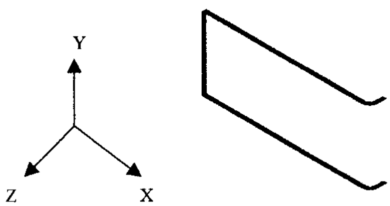


Figure 4. Fastening Boundary and Global Coordinate System

The fastening boundary retains the information from the surfaces of the two assembled parts. The fastening boundary shown in Figure 4 retains the unit normal information from the assembled parts. This information is critical as it mathematically defines which part is mated with its interior surfaces and which part is mated with exterior surfaces. Interior surfaces are defined as surfaces that have positive unit normals (that is, unit normals that point towards the part's centroid) and exterior surfaces are defined as surfaces that have negative unit normals (unit normals that point away from the part's centroid). The fastening boundary also contains all of the geometry information of the contacting parts; the size, location, and types of surfaces (straight or curved).

[1] The surfaces in the fastening boundary potentially can be either attached or fastened. The term *attachment boundary* was not used in keeping with terminology developed in feature recognition strategy literature.

Convex Representations (Steps 2B and 3B)

With the surface pairs identified and the possible attachment of the surfaces described, the assembly procedure must be determined. Some of the assembly procedure data has already been determined with the fastening boundary. While inspection clearly shows that the installation direction is negative along the quadrant formed by the ZX plane, the fastening boundary is not sufficient to prove the positive or negative direction. Information about the parts themselves must be known to ensure that a proposed installation direction does not violate any solid portion of the two parts. To accomplish this, a technique from CAM tool path determinations is used: the convex representation of the parts [13].

Briefly, a convex representation is created by enclosing the part in such a way that all surfaces are convex and no discontinuities (such as steps or cutouts) exist. The convex representation of the battery cover is shown in Figure 5.[2]

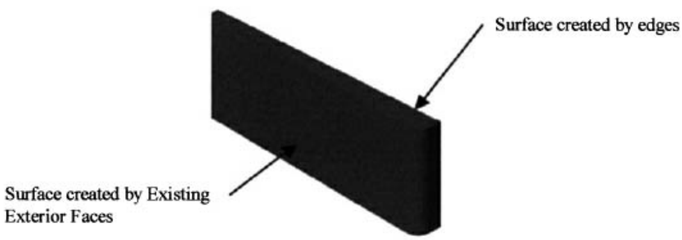


Figure 5. Convex Representation of the Battery Cover

Determination of Violated Surfaces and Surface Types for Attachment (Steps 4 and 5)

The interaction between the new surfaces created from the convex representation of the parts and the fastening boundary provides the remaining information necessary to calculate the assembly procedure. In Figure 6, the fastening boundary is superimposed with the assembled convex representations of the parts.

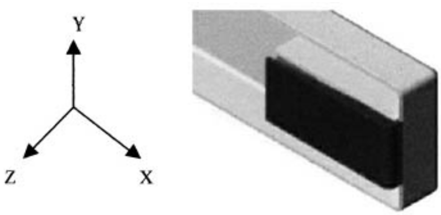


Figure 6. Assembly of Convex Parts and Fastening Boundary

In the design tool, the assembly shown in Figure 6 is not created. The information is gleaned from data received via the API.

[2] The remote cover convex representation is identical to the actual part and hence, not included in Figure 5.

Violated surfaces are surfaces in the complex representations that interfere with the fastening boundary. The first step in determining these surfaces is to remove any surfaces on the convex representations that were created directly from the exterior faces of the parts. The remaining surfaces are shown in Figure 7. The remaining surfaces were created from edges of the solid faces and thus there is some void between the solid faces of the parts and these surfaces.

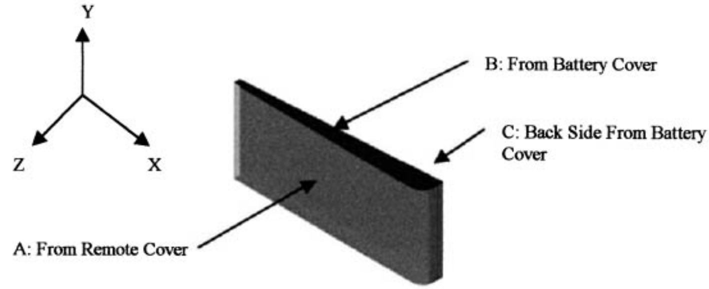


Figure 7. Possible Installation Direction Surfaces

The surfaces A and B intersect the fastening boundary in the fully assembled state; these are the violated surfaces. They are determined by querying the data for surface interference between the convex surfaces shown in Figure 7 and the fastening boundary.

Determination of Installation Direction (Step 6B)

The installation direction will be along the unit normal of one of surfaces shown in Figure 7 [14]. Surfaces B and C can be eliminated as possible installation directions. From the fastening boundary, it is clear that solid material would be violated if these surfaces were engaged. This leaves surfaces A: the installation direction is the quadrant defined by the positive ZX plane; the limits of the installation directions are the Z axis and the X axis

An examination of the violated surface B provides data about the installation direction along the X axis. The direction of assembly for the battery cover with respect to the remote cover is along the direction of decreasing slope for surface B [15]. The slope is calculated by examining the unit normal of the violated surface to the fastening boundary. If the surfaces are sloped, then the parts can be assembled along the line of decreasing slope; if the fastening boundary and the violated surfaces are parallel, then it may be possible to assemble the parts in either direction (as is the case along the Z axis). However, from data obtained from surfaces common to both the remote cover and the convex representation of the remote cover, the positive Z installation direction is eliminated. Ideally, a single degree of freedom should exist for integral attachment. When multiple degrees of freedom exist (as in this example), then a larger set of potential locking features may be presented at the end of the analysis.

Determination of Surface Types (Step 6A)

Surfaces can come together into an assembly in three ways, the surfaces can slide together as a lap joint, one surface can butt against another or one surface can butt against a portion of another surface (T-butt) [5][10]. The fastening boundary identifies these three possibilities.

The unit normals for each surface pair in the fastening volume will point in opposite directions. The sign of the unit normal will identify exterior or interior surfaces. If the interior surface unit normal points towards the interior of the fastening volume then the surface pair is a lap. If the interior surface unit normal does not point towards the interior of the volume, then the surface pair is a butt or T-butt. To determine the type of butting surface, the fastening volume surfaces are compared to the parent surfaces on the parts. If the fastening volume surfaces are identical, then the surfaces are butt surfaces, if one of the fastening volume surfaces is comprised of a portion of the part surface, then the surfaces are a T butt.

Determination of Assembly Motion (Step 7)

There are four basic types of assembly motions [12]: push, slide, tip, and spin. Push and slide are translation motions where the attachment surfaces for the two parts approach each other along a normal direction to the surfaces (push) or parallel direction to the surfaces (slide). Spin is a rotational motion between the two attachment surfaces. Tip is a combination of translation and rotation (such as a lid with a hinge). Spin can be directly determined from the fastening boundary; if any unit normals to the fastening boundary are parallel, then spin is eliminated. Push and slide can be evaluated from the violated surfaces and the fastening boundary; a sloped violated surface indicates slide, violated surfaces normal to the fastening boundary indicates slide. Tip is the most difficult of the four motions to analyze. It is believed that a fastening boundary consisting of one planar surface is the only condition that will allow tipping but this work is still in progress. With this example assembly, spin can not occur, slide can occur along the X or Z axis, push can occur along the X axis, and tip can occur along the Y axis but not the X axis (due to interference).

Display of Attachment Alternatives (Step 8)

It has been shown in previous research [6], that only certain combinations of latches and catches can be used with specific assembly motions and attachment surface pairs. With this information derived from the fastening volume and the convex representations of the parts, it is possible to display appropriate fastening combinations (latches and catches) to the user through the solid modeling tool.

4 Summary

The initial work has shown that an integrated design tool to select fastening alternatives by determining assembly motion and attachment surface pairs is possible. To complete the initial stage of the design tool, fastening selections need to be added to the solid modeling system. Further integration of the design algorithms and the solid modeling system must also be finished.

Rreferences

[1] Genc, S., Messler, R.W., Gabriele, G.A., and Bonenberger, P., 1997, "Enumerating possible design options for integral attachment using a hierarchical classification scheme," *ASME Trans., Journal of Mechanical Design*, Vol. 119, No. 2, pp.178-184.

[2] Genc, S., Messler, R.W., and Gabriele, G.A., 1997, "Generating alternative attachment concepts in integral snap-fit assemblies," *Proceedings of ICED '97, the 11th*

International Conference on Engineering Design, Tampere, Finland, August 19-May 21, 1997, pp. 749-752.

[3] Genc, S., Messler, R.W., and Gabriele, G.A., 1998, "A design methodology for attachment concept development in integral snap-fit assemblies," *Proceedings of ASME'98 DETC/DTM, the 10th International Design Theory and Method Conference*, Atlanta, Georgia, September 13-16, 1998.

[4] LaCourse, D., 1995, "Handbook of Solid Modelling", McGraw-Hill Inc., New York, New York.

[5] Messler, R.W., Genc, S., and Gabriele, G.A., 1997, "Integral Attachment Using Snap-Fit Features: A Key to Assembly Automation. Part 1-Introduction to Integral Attachment Using Snap-Fit Features", *International Journal of Assembly Motion,* Vol. 17, No. 2, pp. 143-155.

[6] Messler, R.W., Genc, S., and Gabriele, G.A., 1997, "Integral Attachment Using Snap-Fit Features: A Key to Assembly Automation. Part 4-Selection of Locking Features", *International Journal of Assembly Motion,* Vol. 17, No. 4, pp. 315-328.

[7] Genc, S., Messler, R.W., and Gabriele, G.A., 1998, "Integral Attachment Using Snap-Fit Features: A Key to Assembly Automation. Part 5-A Procedure to Constrain Parts Fully and Generate Alternative Attachment Concepts", *International Journal of Assembly Motion,* Vol. 18, No. 1, pp. 68-74.

[8] Genc, S., Messler, R.W., and Gabriele, G.A., 1998, "Integral Attachment Using Snap-Fit Features: A Key to Assembly Automation. Part 6-Evaluating Alternatives for Design Optimization", *International Journal of Assembly Motion,* Vol. 18, No. 2, pp. 153-165.

[9] Genc, S., Messler, R.W., and Gabriele, G.A., 1998, "Integral Attachment Using Snap-Fit Features: A Key to Assembly Automation. Part 7-Testing the Conceptual Design Methodology with a Case Study", *International Journal of Assembly Motion,* Vol. 18, No. 3, pp. 223-236.

[10] Genc, S., 1997, "Formalization of Conceptual Design In Integral Snap-Fit Assemblies", Doctorate Thesis, Rensselaer Polytechnic Institute, Troy, NY.

[11] Ullman, D., 1992, "The Mechanical Design Process", McGraw-Hill Inc., New York, New York.

[12] Baxter, D.H., Gabriele, and G.A., "Generation of Fastening Concepts Within CAE Tools Using a Feature Based Strategy", *Proceedings of CIRP 2000,* Haifa, Israel, May 2000.

[13] Kim, Y., and Wilde, D.J., 1992, "A Convergent Convex Decomposition of Polyhedral Objects", *Transactions of the ASME,* Vol. 114, September 1992. pp. 468-476.

[14] Hsu, W., and Woon, I., 1998, "Current research in the conceptual design of mechanical products, *Computer-Aided Design*, Vol. 30, No. 5, pp. 377-389.

[15] Wang, E., and Kim, Y., 1999, "Feature-Based Assembly Mating Reasoning", *Journal of Manufacturing Systems*, Vol. 18, No. 3, pp. 187-202.

Douglas H. Baxter
Director CAD/CAM/CAE, School of Engineering, Rensselaer Polytechnic Institute, Office JEC 3018, 110 East 8th Street, Troy, New York, United States, Tel: 518-276-2127, Fax: 518-276-2623, E-mail: baxted@rpi.edu, URL: http://www.rpi.edu/~baxted

FEATURE- AND CONSTRAINT-BASED DESIGN OF SOLUTION PRINCIPLES

T Brix, B Brüderlin, U Döring, and G Höhne

Keywords: feature-based design, constraint-based design, parametric design, conceptual phase

1 Introduction

The conceptual phase of the design process is of special importance to product development. Usually, several appropriate solution principles and corresponding parameters, which fulfill given requirements, will be determined. Such solution principles are often represented by means of symbols, which are intuitive to the human designer. Figure 1 shows the role of solution principle development within the design process and the relationship between functional, principle and part structure. Here, solution principles consist of principle elements and couplings (e.g. joints). This distinction is important when solution principles are used as basis for part design (couplings become connections)[1].

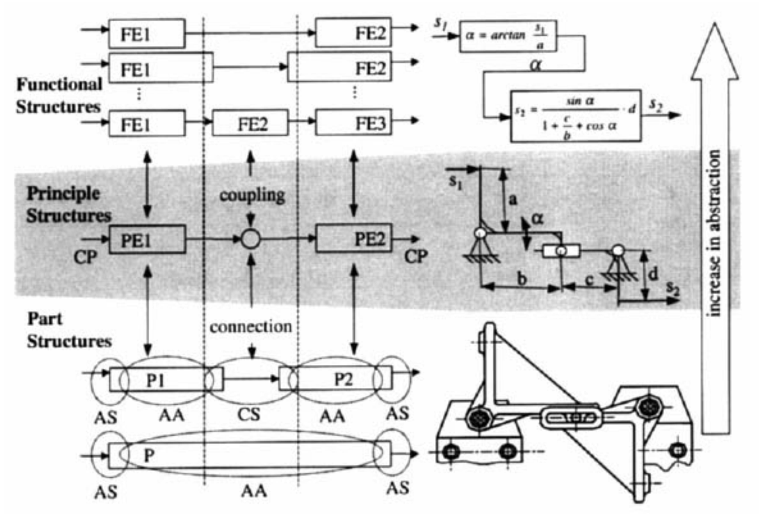


Figure 1. Relationship between functional, principle and part structure
(FE = function element, PE = principle element, P = part, CP = couple point, AS = active surface, CS = contact surfaces, AA = active area)

Constraint solving is a powerful technique for parametric design of 2D- and 3D-models [2]. The models are described by parameters, geometric elements and constraints between them. This modeling technique is suitable for modeling solution principles, also. Applying constraint solving to the development of solution principles means to simultaneously handle the following views - the intuitive, high-level description conveying the user's intent by symbols, on one side, and the constraint-based, low-level design on the other side (see Figure 2).

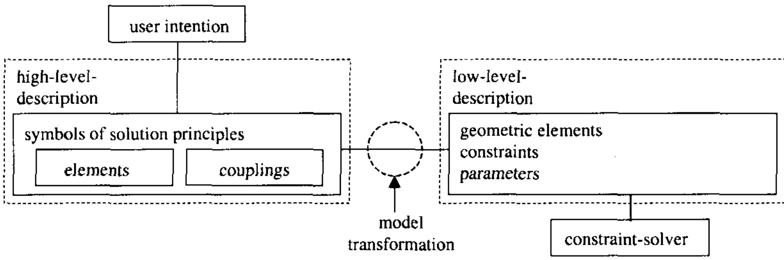


Figure 2. Connection between high-level and a possible low-level description of principle structures

2 Application of the Feature Concept

For a suitable combination of high-level and low-level description we employ the feature concept [3]. Features are a subsumption of both descriptions. Features combine data and methods of the two levels of description as one entity (see Figure 3).

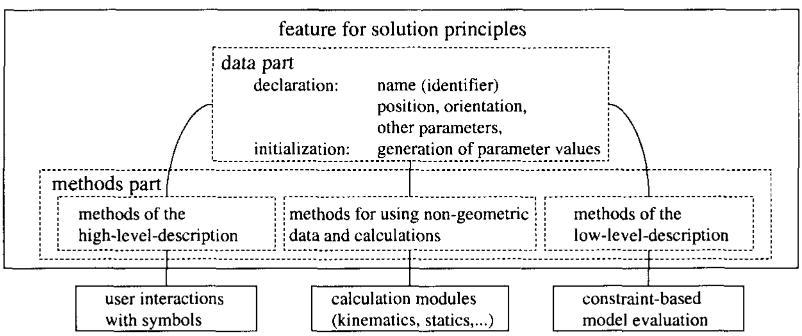


Figure 3. *Structure of features for principle solutions*

In the data part, common information (e.g. name, position and orientation) and symbol specific information (e.g. shape and IDs for the constraint solver) are stored. Data will be manipulated using the methods defined in the methods part. Three groups of methods can be distinguished. The methods of the high-level description implement user interaction and drawing of symbols. The low-level methods generate a suitable constraint-based description. Furthermore they implement the interface to the constraint solver for geometric evaluation and data transfer. The third group provides an interface to other necessary calculation modules, e.g. for kinematics or static evaluations. The shared data concept and according update mechanisms allow the synchronization between the different descriptions in the feature.

The high-level description will not be discussed in detail, because it uses self explanatory symbols, which are created, deleted or modified interactively by standard graphical user interactions (example see section 4).

3 Constraint-based Description

The following tables contain examples for the constraint-based description of selected principle elements, couplings and possible combinations of them. For each symbol (first column = high level representation) a sequence of modeling commands is given, which defines the structure of the constraint-based model. We use commands to generate geometric elements (Point, Line and Circle), parameters (Arg) and constraints (e.g. Fix = element fixing, PinL = point incident line, LinL = line incident line, PdistP = point distance to line). Command suffixes such as _XYA describe the semantics of the parameter list. Between command and parameters, an ID can be given, which will be used to reference the generated entities later on (refer to [4] for a more detailed description of the syntax of the modeling syntax).

Table 1. Constraint-based model of a simple bar (link, axle, ...)

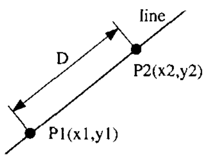
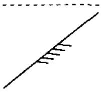
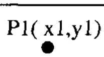
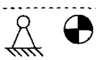
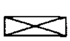
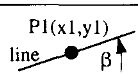
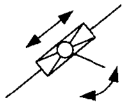
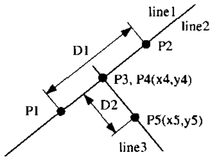
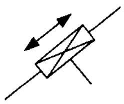
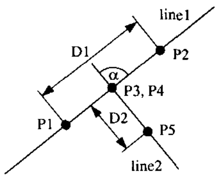
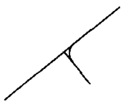
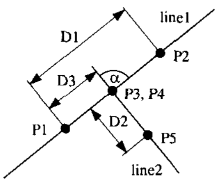
Symbol	Low-level description		Comment
		***Arg*** *bar.length D* ***Line_XYA*** *bar.line 0 0 0* (default line) ***Point_XY*** *bar.P1 x1 y1* ***Point_XY*** *bar.P2 x2 y2* ***Fix*** *bar.length* ***PinL_PL*** *bar.in1 bar.P1 bar.line* ***PinL_PL*** *bar.in2 bar.P2 bar.line* ***PdistP_PPmD*** *bar.dist bar.P1 bar.P2 bar.length*	*P1 and P2 are not fixed* *DOF = 3* (DOF - degree of freedom)
 fixed to the frame		***Fix*** *bar.P1* ***Fix*** *bar.P2*	*P1,P2 are fixed,* *DOF = 0*

Table 2. Constraint-based model of simple joints (turning joint, sliding joint)

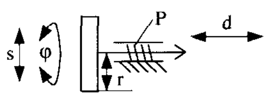
Symbol	Low-level description		Comment
		***Point_XY*** *tjoint.P1 x1 y1*	*x1,y1 - variable,* *DOF = 2*
 fixed joints		***Fix*** *tjoint.P1*	*x1,y1 - constant,* *DOF = 0*
		***Arg*** *sjoint.angle β* ***Point_XY*** *sjoint.P1 x1 y1* ***Line_XYA*** *sjoint.line 0 0 sjoint.angle* ***PinL_PL*** *sjoint.in1 sjoint.P1 sjoint.line*	*x1,y1,β - variable,* *DOF = 3*
 fixed joint		***Fix*** *sjoint.P1* ***Fix*** *sjoint.angle*	*x1,y1,β - constant,* *DOF = 0*

Tables 1 and 2 show the description of simple symbols, which are needed to model planar linkages. Note that the functionality of symbols will be obvious when the relations to other symbols are defined. Table 3 shows such relations of simple symbols.

Table 3. Combinations of simple joints and bars

Symbol	Low-level description	
		*< define bar bar1 (see Table 1) with line1, P1, P2, D1>* *< define sliding joint jt1 (see Table 2) with P3 and line2>* *< define bar bar2 (see Table 1) with line3, P4, P5, D2>* ***LinL_LL** jt1.in1 jt1.line2 bar1.line1* (following inequalities avoid that jt1 leaves bar1) ***PdistP_PPmD** jt1.d1 bar1.P1 jt1.P3 <= bar1.length* ***PdistP_PPmD** jt1.d2 bar1.P2 jt1.P3 <= bar1.length* ***PinP_PP** jt1.P3 bar2.P4*
		***Arg** jt1.angle α* ***Fix** jt1.angle* ***LangL_LLA** jt1.a1 bar1.line1 bar2.line2 jt1.angle*
		***Arg** jt1.length D3* ***Fix** jt1.length* ***PdistP_PPmD** jt1.dist jt.P3 bar.P1 jt1.length*

More complex symbols can be defined by means of equation constraints. This is used for example to model the screw mechanism shown in Figure 4 and Table 4.



$$d \cdot 2\pi \cdot r = s \cdot P$$

Figure 4. Screw mechanism and transfer function (*s* may be determined interactively by a mouse drag which defines angle *φ* with *φ=s/r*, *P* is the pitch of the screw, *d* is the resulting displacement)

Table 4. Constraint-based model of a screw mechanism

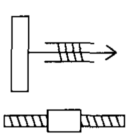
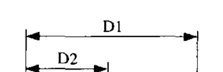
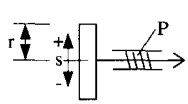
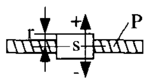
Symbol	Low-level description	
		***Arg** scrw.le1 D1; **Arg** scrw.le2 D2* ***Arg** scrw.le3* ***Arg** scrw.2pi 2*Pi; **Arg** scrw.radius r* ***Arg** scrw.sl P; **Arg** scrw.inp s* ***Arg** scrw.circumf* ***Arg** scrw.h1* (auxiliary variable)
		***Point_XY** scrw.P1 x1 y1* ***Point_XY** scrw.P2 x2 y2* ***Point_XY** scrw.P3 x3 y3* ***Line_XYA** scrw.line 0 0 0* (default line) ***Fix** scrw.le1* ***Fix** scrw.2pi* ***Fix** scrw.radius* ***Fix** scrw.sl*

Table 4. Continuation

Symbol	Low-level description
	 ***PinL_PL*** *scrw.in1 scrw.P1 scrw.line* ***PinL_PL*** *scrw.in2 scrw.P2 scrw.line* ***PinL_PL*** *scrw.in3 scrw.P3 scrw.line* ***PdistP_PPmD*** *scrw.d1 scrw.P1 scrw.P2 scrw.le1* ***PdistP_PPmD*** *scrw.d2 scrw.P1 scrw.P3 scrw.le2* ***PdistP_PPmD*** *scrw.d3 scrw.P2 scrw.P3 scrw.le3* ***Equ_YmM*X+N*** *scrw.h1 = scrw.s1 * scrw.inp* ***Equ_YmM*X+N*** *scrw.h1 = scrw.circumf * scrw.le2* ***Equ_YmM*X+N*** *scrw.circumf = scrw.radius * scrw.2pi* ***Equ_YmM*X+N*** *scrw.inequ1 scrw.le2 <= scrw.le1* ***Equ_YmM*X+N*** *scrw.inequ2 scrw.le3 <= scrw.le1*

It is possible to integrate non-geometric quantities into the model by means of equation constraints. An example is shown in Figure 5, where a piezo element causes a translation according to a given voltage.

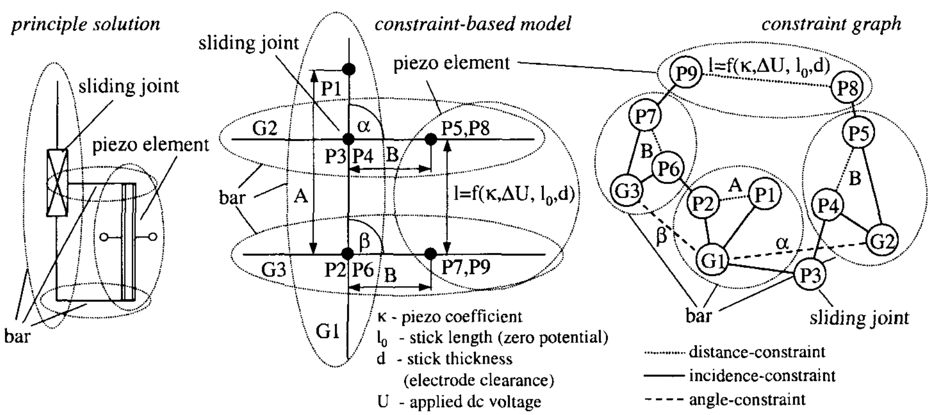
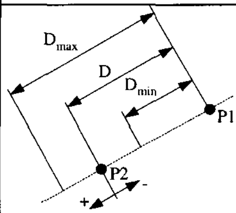


Figure 5. Example for using non-geometric data in a translation element

Previous examples illustrated the application of invariant constraints, which are used often in geometric modeling. All these constraints have the same priority; all must be fulfilled simultaneously to generate a valid model. Both restrictions (invariance and equal priority) must be relaxed to define symbols such as springs and geneva mechanisms in a reasonable way.

Soft Constraints are constraints with a lower priority. In over-constrained cases soft-constraints can be neglected according to their priority. The definition of default values for parameters is one possible application of soft constraints. In table 5 the current length D of a spring is defined by the distance constraint between the points P1 and P2. When both positions are determined (e.g. by a mechanism) the default value of D can not hold. Only after the distance is not determined by other constraints the soft constraint, which links D and its default length, can be applied during model evaluation. The restrictions D_{max} and D_{min} given by inequality constraints must always be fulfilled. They define the range of the spring (or a desired part of it). In section 4 interactive movement of a spring is discussed in more detail.

Table 5. Constraint-based model of a spring

Symbol	Low-level description
	 ***Arg*** *sp.deflength D* (default distance) ***Arg*** *sp.maxlength Dmax* ***Arg*** *sp.minlength Dmin* ***Arg*** *sp.length* ***Point_XY*** *sp.P1 x1 y1* ***Point_XY*** *sp.P2 x2 y2* ***Fix*** *sp.deflength;* ***Fix*** *sp.maxlength;* ***Fix*** *sp.minlength* ***PtoP_PPmD*** *sp.length sp.P1 sp.P2* ***Equ_YmM*X+N*** *sp.length <= sp.maxlength* ***Equ_YmM*X+N*** *sp.length >= sp.minlength* ***DefaultValue*** *sp.length sp.deflength*

Conditional Constraints can be used to define a set of active constraints depending on certain conditions. In mechanism design it is important to model contact couplings, which can be active or inactive. In Figure 6b a possible abstraction of the geneva mechanism is given. The interaction between pin and slots are modeled by "point in line"-constraints, where the pin is a point (P2) and the symmetry line of a slot is a line (L1 or L2). The distance between P2 and P3 (the hub of the geneva wheel) is used to check if the pin is in contact with the wheel. In case of contact, the constraints "P2 in L1" or "P2 in L2" respectively become active. A second condition controls which the two. In [4] a more detailed description of conditional constraints is given.

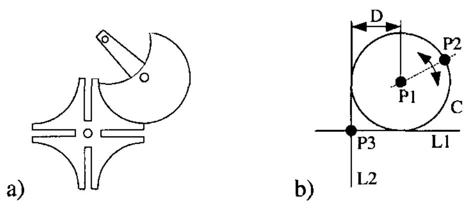


Figure 6. Geneva mechanism a) as symbol and b) definition by means of conditional constraints

4 MASP

The ideas described in the previous sections have been implemented in an application called MASP (program for Modeling and Analyses of Solution Principles). The interactive modeling of solution principles is done by selecting symbols in the context of chosen activities. An example of a modeling sequence for a crank-rocker mechanism is given in Figure 7 (1)-(4).

During interactive high-level modeling, a low-level description by constraints is generated automatically (Figure 8 shows the resulting feature model). MASP enables the user to immediately test the functionality of the current design concept, for instance, by interactive mouse drags or by applying further calculations (e.g. kinematics or static calculations) based on the evaluated constraint model [5].

The interplay of the different description levels is illustrated in figure 7(5). The user modifies the model interactively by dragging a joint. This information will be saved in the data part of the joint. Based on the current parameters and positions in the low-level description the constraint-solver computes the new positions of all connected geometric entities as well as non-geometric data. Furthermore other calculation modules will be used to recalculate dependent data, for instance to determine physical forces. After this the updated high-level description is used to modify the representation on the screen, for instance the symbol of the spring, which may include a visualization of the force.

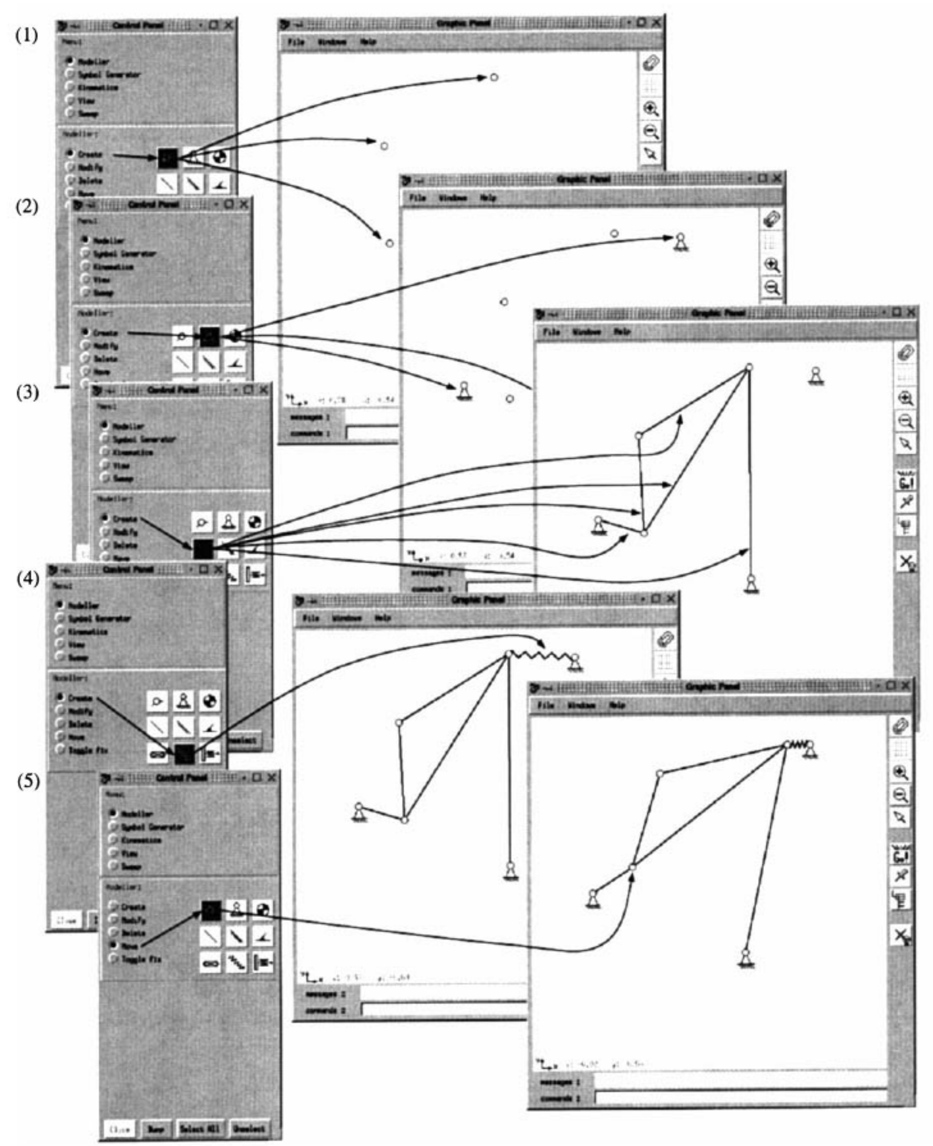


Figure 7. Interactive modeling of a crank-rocker mechanism

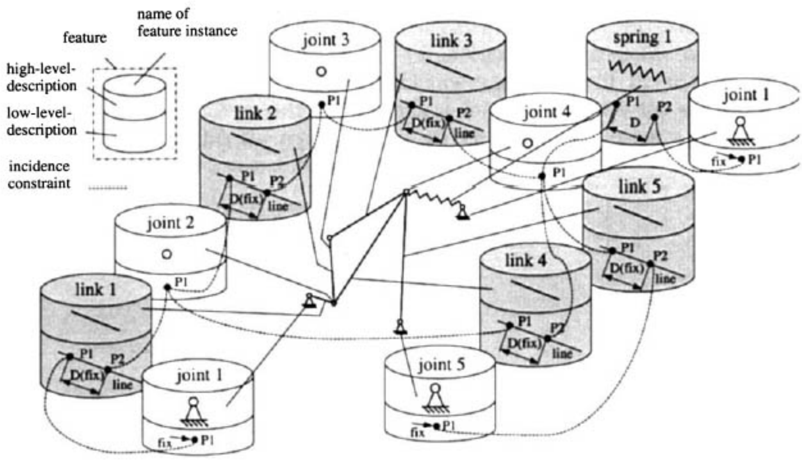


Figure 8. Feature model of the crank-rocker mechanism from Figure 7

5 Conclusion and Further Work

We developed MASP, an application which supports the conceptual design phase. The feature concept was used to combine different description levels. The constraint-based model allows to perform various analyses to find a solution that fulfils the requirements. In future work, additional symbols and calculation methods as well as visualization of calculation results will be integrated into MASP. Furthermore, in a research project - sponsored by German Research Foundation (DFG) - the transition between 2D-principle solutions and 3D solid models is investigated. Here bidirectional model transformation shall be achieved.

References

[1] Pahl, G.; Beitz, W.; Wallace, K., "Engineering Design", Springer-Verlag, Berlin, 1999.

[2] Roller, D.; Brunet, P., "CAD-System Development", Springer-Verlag, Berlin, 1997.

[3] Shah, J.J.; Mäntylä, M., "Parametric and Feature-based CAD/CAM", John Wiley & Sons, New York, NY, 1995.

[4] Brix, T.; Brüderlin, B.; Döring, U., "Conditional Constraints in Conceptual Design", Proceedings of 32st ISATA, Vienna (Austria), 1999, pp. 441-448.

[5] Brix, T.; Brüderlin, B.; Döring, U.; Höhne, G.: "Using a Geometric Constraint Solver for Analysis in the Early Phases of Product Modeling", Proceedings of the 12th ICED, Munich (Germany), 1999, 1869-1872.

Prof. G. Höhne and T. Brix, Ilmenau TU, Dept. of Mech. Engineering, Engineering Design Group, PO Box 100565, 98684 Ilmenau, Germany, Tel: +49 3677-69-2473, Fax: +49 3677 69 1259, E-mail: guenter.hoehne@mb.tu-ilmenau.de and torsten.brix@mb.tu-ilmenau.de.
Prof. B. Brüderlin and U. Döring, Ilmenau TU, Dept. of Computer Science, Computer Graphics Program, PO Box 100565, 98684 Ilmenau, Germany, Tel: +49 3677-69-2785, Fax: +49 3677-69-1285, E-mail: bdb@prakinf.tu-ilmenau.de and udoering@prakinf.tu-ilmenau.de.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

REDESIGNING WITH FEATURE GRAMMARS AND FBS MODELS

S C Chase and P Liew

Keywords: redesign; design for assembly; design for manufacture; feature-based design; function-behaviour-structure model

1 Introduction

Design generally involves different disciplines, each forming a particular design domain with its specific viewpoint of the design. For example, a product designer may consider the general form of a product and leave the structural details to the mechanical designer. The mechanical designer may consider the engineering requirements and leave the manufacturing details to the manufacturer.

An initial design created in one domain is usually subject to modification by another, as each domain has different requirements. In this context, *redesign* is considered as the process of modifying an existing design based on additional criteria from another domain. The current workable design is the starting point for the redesign process; additional requirements from different domains can lead to modification of the initial design in order to conform to these new requirements. Redesigning is common if the designer does not have enough knowledge regarding the manufacturing process to be used. This is normally the case when the manufacturing process is yet to be selected.

This paper describes a framework for redesigning. A model of design based on the derivation of a design grammar with Function-Structure-Behaviour (FBS) descriptions is proposed. Stylistic changes as defined by rule modifications [1] are used to replace rules used in the derivations of the original design with new rules that produce designs conforming to additional requirements. The mechanism that enables this replacement is based on the FBS characteristics of the design.

2 FBS model of designing

The FBS model of designing explicates the relationships between function, behaviour and structure of an artefact and facilitates explicit reasoning among them [2]. These dependency relationships among function, behaviour and structure are described using dependency networks. Each node in the network represents a function, behaviour, structure or their characterising variables; the link across nodes describes the dependency between them.

Graphs can be used to represent the dependency network [3]. Figure 1 illustrates a sample FBS representation for a fastening subassembly consisting of a bolt and nut binding two plates.

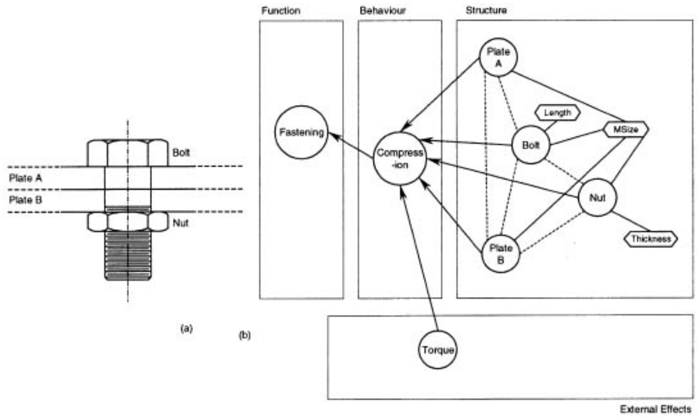


Figure 1. A bolt and nut subassembly and its corresponding FBS representation

3 Design descriptions

A design grammar can be used to generate a design with an associated FBS description. As the design is generated using the transformation rules, its FBS description is created by a description function that maps designs in the language of the grammar to descriptions in a predefined description set. The description function is defined by a description for the initial element in the design grammar and a mapping function, g_i for each transformation rule, i [4].

In this work, design descriptions are used to model the FBS characteristics of a design. The description function creates an FBS representation of the design as it is generated by a design grammar. This description is then used for the redesigning of the original artefact based on different requirements operating on the FBS characteristics of the design.

4 Stylistic change

Terry Knight has demonstrated how systematic modification of shape grammars can encapsulate stylistic change in art and architecture [1]. This occurs through addition, deletion or modification of grammar rules, often by shape replacement or modification of spatial relations. While capturing these changes in a very clear manner, there is little mention of the motivation for any transformation (decidedly outside the scope of her work).

In our work we expand the scope of these transformations to include ones based on modification of the functional, behavioural or structural characteristics of designs, as defined in their FBS descriptions. These transformations manifest themselves as rule replacements, and are motivated by specific requirements for redesign, e.g. new functional requirements.

5 Redesign framework

The redesign process is based on specific operations acting on the functional, behavioural and structural properties of the original design within the context of additional requirements. The FBS description of the original design is first constructed and later modified. Figure 2 illustrates the various processes for this redesigning framework.

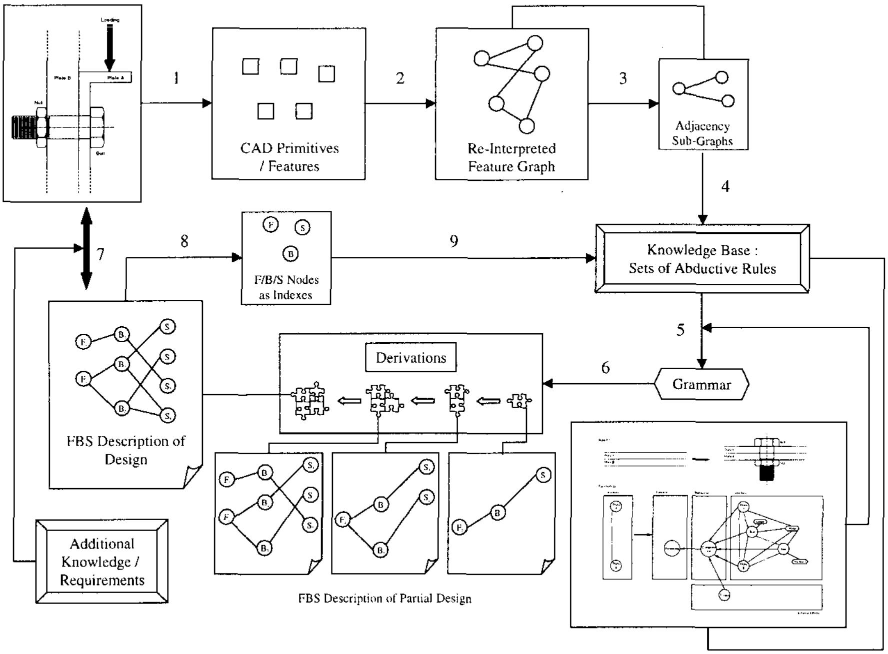


Figure 2. Framework for the construction of FBS descriptions

5.1 Construction of FBS descriptions

A design is created with a design tool such as a solid modeller by using its native geometries or predefined feature sets (Step 1). The resulting model is reinterpreted as graphs of the required features specific to the relevant domain for redesigning (Step 2). Typically, geometric features are represented as graphs of geometric entities in a solid model data structure and graph grammars can be used for domain-specific feature interpretation [5, 6].

Various adjacency sub-graphs of this feature graph are extracted (Step 3) and used as indexes (Step 4) to search for abductive rules that model part of the designer knowledge base [7]. Each of these rules is modelled with a mapping function, g_i that generates its FBS description. The structure part of this description contains adjacency information between structural elements that is matched against the adjacency sub-graphs from the original design for the retrieval of relevant rules. Figure 3 illustrates an example of one such rule.

A grammar is constructed from a set of such rules and used to construct the derivations of the original design (Steps 5 & 6). This derivation recreates the original subassembly with an associated FBS description. This description is represented as graphs that define the various functional, behavioural and structural properties of the original design as nodes and their interrelationships as arcs. Additional input from the designer is required to form a complete FBS description, as the derivation provides only a partial description (Step 7). This additional information can be obtained from a functional analysis of the original design.

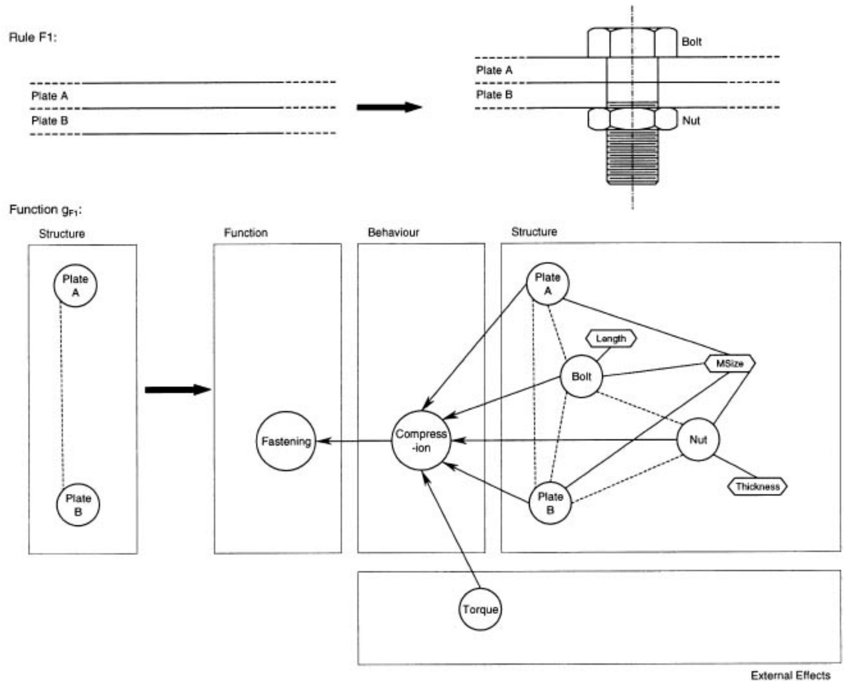


Figure 3. A sample abductive rule with its associated function g_{F1} to generate the FBS description

5.2 Redesigning

To carry out redesigning, the various nodes in the FBS description of the original design are used as indexes for the search of alternatives in the library of abductive rules (Steps 8 & 9). New rules are selected, based on additional requirements for the functional, behavioural and structural properties of the original design. An example of a requirement in the domain of Design for Assembly (DFA) considers the use of alternative devices for fastening in place of bolts and nuts. The function "fastening" and structures "plate A" and "plate B" are used as indexes to search for the relevant rule that has the function "fastening" and structures "plate A" and "plate B" in its associated FBS description.

The original rules in the grammar are replaced by the newly selected rules, resulting in a *transformed grammar*. A new design is regenerated using this modified grammar and its corresponding FBS description is created. A comparison of the original and modified FBS descriptions highlights potential requirements for any elements that have been added or modified.

6 DFMA redesign example

The following example illustrates the application of the above framework in the context of redesigning for DFMA (Design for Manufacture and Assembly) conformance.

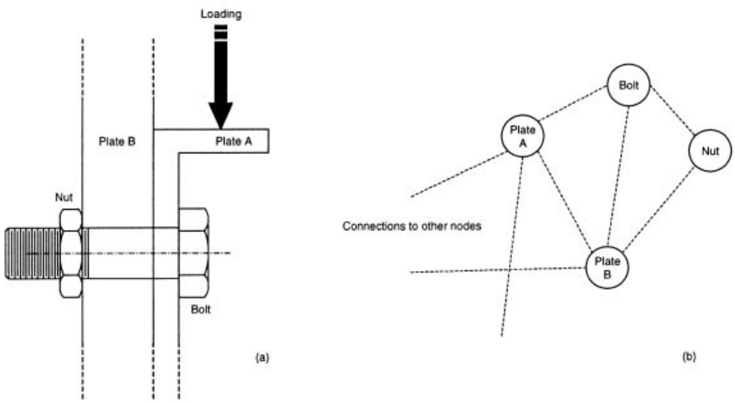


Figure 4. Subassembly to support vertical load and its adjacency graph

We assume that the user uses features to design the subassembly (Figure 4a) and that the requirements for DFMA utilise the same set of features. The feature recognition process can thus be ignored; the bolt, nut, plate A and plate B form the features involved in the redesign of the subassembly.

The adjacency between these features can be obtained from the CAD database of the original design. A portion of this adjacency graph is also shown in Figure 4b. This adjacency subgraph is used is an index to search for abductive rules that have the same adjacency information in their FBS descriptions.

A grammar is constructed from a set of such rules and used to construct the derivations of the original subassembly. This derivation recreates the original subassembly with an associated FBS description. The description for the subassembly is illustrated in Figure 5. Note that rule Fl provides only part of the FBS description. The parts on *Supporting Vertical Load* and *Rigidity* (dotted arrows with question marks) are obtained from the designer's knowledge base via function analysis. This will highlight the additional requirements for the structures used in the rule to provide the additional behaviour of *Rigidity* to support the load.

Based on the FBS description created for the original subassembly, the different nodes are used as indexes to search for alternative designs within the knowledge base of the designer. For this example, the rule F2 (Figure 6) is selected based on its provision of the fastening function and the use of snap fits. By replacing the original rule (F1) in the derivations of the subassembly with a new rule (F2), the original grammar is transformed into a new grammar that produces designs that conform to the new requirements from DFMA.

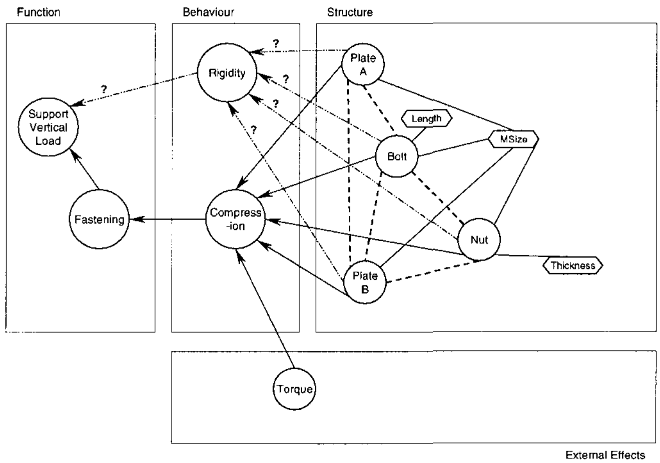


Figure 5. FBS description of the subassembly

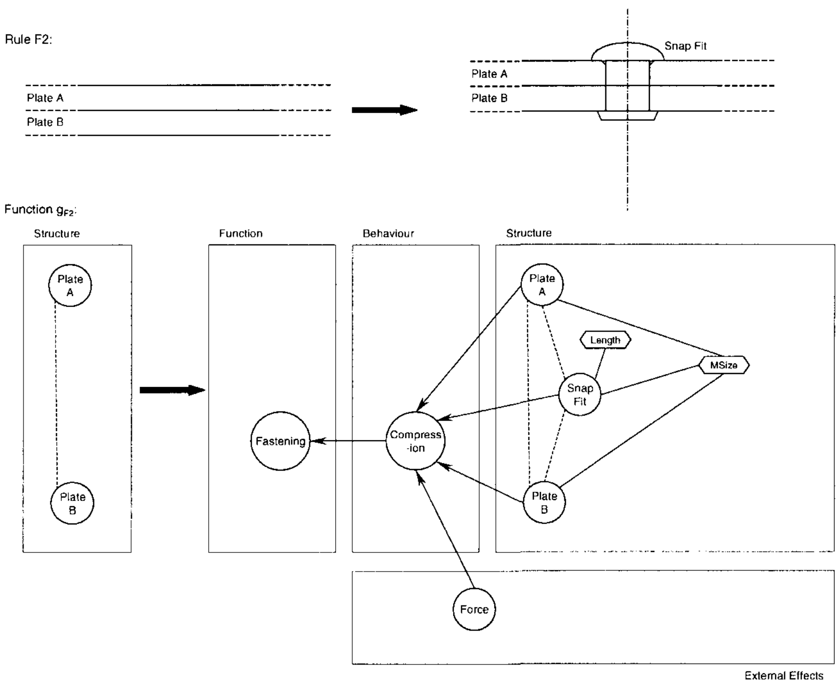


Figure 6. Rule F2 for an alternative fastening mechanism

The new design for the subassembly is constructed from the derivations of the evolved grammar and its corresponding FBS description is generated. To model the interactions of the new element (snap-fit) in the original design, the FBS descriptions of the old and new designs can be compared through graph comparison operations.

7 Discussion

It should be noted that the design state space of the original grammar can be transformed according to Figure 7. If the rule replacement strategy for the modification of the original grammar is based on the context of DFA alone, this can be modelled as a single objective in the transformation process as shown in Figure 7(a). Since the rules are replaced, the original state space is moved to another location. If the old rules are not replaced but augmented by the new rule, Figure 7(b) results. Figure 7(c) illustrates the case where rules based on a number of different objectives (e.g., DFA, DFM and other forms of DFX) are added to the existing grammar. In this situation, conflict resolution and some form of multi-criteria optimisation techniques are required to generate an optimal design based on multiple objectives.

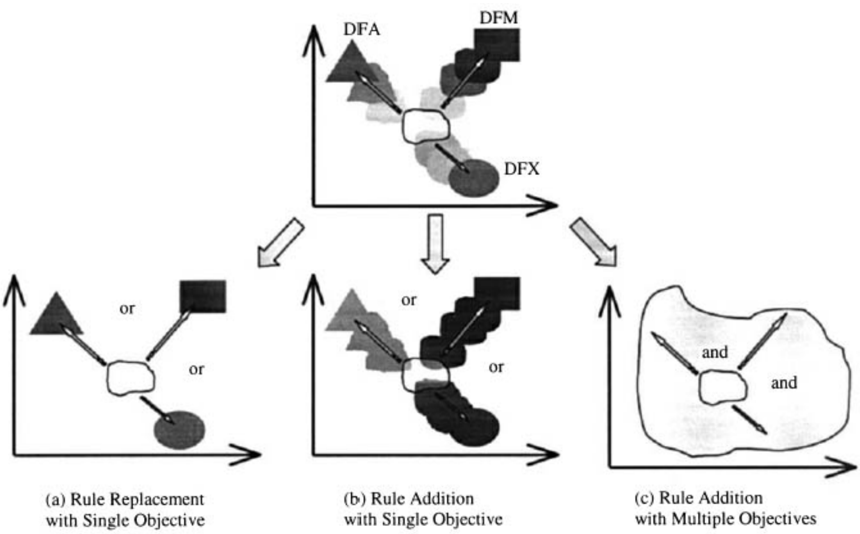


Figure 7. Different approaches to modification of the design state space of the original grammar

In this paper we have proposed a framework for redesign by grammar modification through rule replacement. Further work, including testing of this framework, is required. Additionally, other areas worthy of further investigation include:

- Control of rule selection and invocation, especially in ambiguous, non-deterministic cases.
- Exploration of how prototype libraries can be used in redesign. For example, other possible paradigms used for redesign not examined here include combination, analogy and case-based reasoning.

- Formalisation of design principles by graph operations on FBS descriptions. These operations, based on redesign requirements from DFMA, would act as feedback to the evolution process of the design grammar.

8 Conclusion

In this paper we described a methodology for formalising redesign based on functional, behavioural and structure requirements, utilising the FBS model and design grammars. We proposed a process for the generation of FBS descriptions of designs based on adjacency graphs of their components and knowledge bases of component behaviour. By replacing the rules used to generate a design (evolving the grammar), new designs are created that meet new requirements. This framework has the potential to support the formalisation of engineering design methodologies such as DFMA.

Acknowledgements

This research was partially funded by the Australian Research Council Small Grant Scheme.

References

[1] Knight, T. W., Transformations in Design : A Formal Approach to Stylistic Change and Innovation in the Visual Arts, Cambridge University Press, Cambridge, England, 1994.

[2] Gero, J. S., Tham, K. W., and Lee, H. S., "Behaviour: A link between function and structure in design," Intelligent Computer Aided Design, D. C. Brown, M. B. Waldron, and H. Yoshikawa, Eds., North-Holland, Amsterdam, 1992, pp. 193-225

[3] Qian, L. and Gero, J. S., "Function-behaviour-structure paths and their role in analogy-based design," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 10, 1996, pp. 289-312

[4] Stiny, G., "A note on the description of designs," Environment and Planning B, 8, 1981, pp. 257-267

[5] Chuang, S. H. and Henderson, M. R., "Compound Feature Recognition by Web Grammar Parsing," Research in Engineering Design, 2, 1991, pp. 147-158

[6] Pinilla, J. M., Finger, S., and Prinz, F. B., "Shape feature description and recognition using an augmented topology graph grammar," Preprints: NSF Engineering Design Research Conference, College of Engineering, University of Massachusetts, 1989, pp. 285-300

[7] Gero, J. S. and Maher, M. L., "Theoretical requirements for creative design by analogy," First International Workshop on Formal Methods in Engineering Design, Manufacturing and Assembly, P. Fitzhorn, Ed., Colorado State University, Fort Collins, Colo., 1990, pp. 19-27

Dr. Scott C. Chase
Department of Architecture and Building Science, University of Strathclyde, 131 Rottenrow, Glasgow G4 0NG, United Kingdom, Tel: +44 141 548 3023, Fax: +44 141 552 3997, E-mail: s.c.chase@strath.ac.uk, URL: http://homepages.strath.ac.uk/~cas01101
Former affiliation: Department of Architectural and Design Science, University of Sydney

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

THE USE OF DISTRIBUTED VIEWPOINT-DEPENDENT FEATURE-BASED MODELLING AND THE RESPONSE SURFACE METHOD IN DESIGN ASSESSMENT

K Lee, I Kaymaz, and C A McMahon

Keywords: Computer-aided design, viewpoint dependent modelling, feature-based design, response surface methods

1 Introduction.

The engineering design process is typically highly complex and involves the collaboration of many specialists over a significant period of time. During the process, various engineering representations are created and used by the different specialists depending on their needs in terms of manufacturing, structural analysis, process planning, maintenance, and so on. These various representations may be regarded as the different **viewpoints** of the specialists on the representation of the emerging design. In general, each viewpoint on an engineering design may correspond to a different underlying representation or a variation on a representation. These representations are referenced to each other [1], and the specialists exchange information through a common vocabulary [2]. During the design process, a great deal of effort is spent in moving or converting data from one representation to another.

The effort required in model conversion is particularly acute in those engineering applications involving repetitive operations. Examples include the repeated analysis required in design iteration or optimisation, or in probabilistic analysis. In these applications the multiple execution of analysis processes is in itself very time consuming and a significant constraint on the application of these methods. In this paper an approach will be described that seeks to address performance issues, in both model conversion and in repetitive analysis, through a combination of feature-based design and response surface analysis.

In feature-based design the design representation is created from features – prototypical parametric shapes with some engineering significance or meaning. By using features the effort required in the construction of a model may be very much reduced. In viewpoint-dependent feature-based design there may be automatic conversion between the feature-sets used for different viewpoints, or an underlying design feature representation may be presented differently in different viewpoints.

In this paper a viewpoint dependent modelling approach is proposed in which a design feature model is translated into a finite element model for stress analysis. Since the feature-based models are described by variable parameters, variations on the models may be defined for the purposes of probabilistic assessment simply by varying the values of the describing parameters. However, a computationally expensive process, such as the execution of multiple finite element analyses for probabilistic analysis by the Monte Carlo method or by the First Order or Second Order Reliability Methods, is very time consuming [3]. In these circumstances, the Response Surface Method (RSM) can be used to alleviate the difficulties of the probabilistic assessment methods by forming a mathematical function, called a response surface function (RSF), that can be used in place of the computational analysis. We will describe an adaptive response surface methodology, called ADAPRES, which attempts to find the best RSF for the problem under investigation, by applying statistical tests to the RSF, and by increasing the polynomial order of the RSF as required [4]. ADAPRES is incorporated into a design support system that combines the response surface methodology with a viewpoint dependent feature modeller used to generate data points for the response surfaces.

This paper will give a brief overview of each of the elements of the system, and will then describe their integration into the design support system. The techniques will be illustrated by an example in which the approach is used to remotely and repeatedly generate a geometry-based viewpoint model for response surface analysis in a distributed object system.

2 Research aim and objectives.

The objective of the present research is to see whether a combination of feature-based models and distributed computing can begin to address issues of both viewpoint dependency and computational expense. The work is particularly relevant in variational design areas such as automotive engineering, in which it is known in advance what form engineering parts will take. The objective is to identify whether, by describing parts as collections of features, they may be modelled rapidly from different viewpoints, and then these models used to build response surfaces functions automatically to characterise the relationship between inputs and outputs for applications such as stress or geometric analysis. The method that has been employed is to use the feature-based modelling capability of a commercial CAD system, called through the system's Application Programming Interface (API), using a macro language to allow repeated construction of features. Distributed computation is archived partly through the Common Object Request Broker Architecture (CORBA), and partly through a socket-based network programming approach.

3 Research background.

3.1 Feature-based design.

An early use of features in computer-aided design was the identification of model elements of significance to process planning through feature recognition [5]. Early usage of feature-based design may be attributed to Arbab, who introduced the idea of the construction of a part representation by successive set-theoretic subtractions of features corresponding to manufacturing operations in destructive solid geometry [6]. In 1984, Pratt [7] formed the idea of designing directly with features. Nowadays, many designers understand that features are now part of mainstream CAD, in which feature-based modelling represents part models in terms of functionally significant high-level entities in geometry such as holes, slots, pockets, or bosses [8]. Feature-based design is a means for designers to abstract the level of design by working with high-level entities instead of dealing with low-level geometric details. In feature-based modelling methodologies, the transformation of a geometric representation into a feature representation, such as design by features (in which the part is created *ab initio* in terms of features) and feature recognition (in which features are recognised on an existing part), is generally considered a prerequisite for implementing automation and integration in Computer Aided Design.

Approach to viewpoint-dependent modelling.

Each engineering discipline has its own viewpoint on a design. Each of the engineering activities that are carried out at a design stage needs a particular representation of the product to address the information requirements of the activity [9]. However, a representation that addresses a specific viewpoint may be incompatible to fulfil the requirements of other viewpoints, such that features may be of little value outside of the design or engineering discipline for which they were created. For example, a set of rib features created for airframe structural analysis modelling may be of little use to a manufacturing engineer who will view the same component geometry as a set of machined pockets. In recent years there have been a number of attempts to devise representations that allow multiple design viewpoints to be supported using feature technology [9]. In the MG-IT project, for example, different viewpoints on the product model give the user access to different subsets of the design environment for analysis and simulation applications [10], and an integrated feature-modelling system for a multi-viewpoint modelling system is built by Martino [11]. Kugathasan [12] focuses on the development of a representation that allows derivation from a single central design representation of the differing viewpoints from the various engineering specialists. Limitations in current feature modelling systems with multiple viewpoint models are mentioned by Bidarra [8], while Hoffmann describes maintaining views consistently in his object-oriented and distributed product information database [13].

The design by features approach typically provides a designer with a library of features. The feature library may not only store geometric meaning but may also have associated attributes that are meaningful in the context of engineering applications. In viewpoint dependent features, each different viewpoint on the design process may have its own library of features, and these libraries may intersect with each other through shared features.

3.2 Design automation.

Through increasing automation of design and other engineering processes, the quality and efficiency of the design process may be improved. The key to success in achieving automation is the integration of the information processing required by the various disciplines involved at the various stages of the design process. Several research works describe approaches to the automated design process. Bodkin [14] demonstrated the automation of the design and FE simulation process with feature-based technology. His paper describes a structural analysis and design system which is based on geometric primitives that represent certain form-features of the part and can be assembled into complete solid models that are defined in terms of a small set of design parameters. Automatic generation of manufacturing information from design has also been researched by Liu [15]. Modern design systems often depend on a numeric or algebraic approach to design automation, but systems that work using low-level geometric entities may have a problem of data access to other applications, and in this respect a feature-based design may be a better solution. Feature construction may be achieved in design automation through use of dedicated knowledge-based engineering systems, through programmatic interfaces to CAD systems (such as the APIs provided by the major vendors), or by the use of macro or program files that allow the repetitive execution of command sequences. It is this latter approach that has been employed in the present work, with variation in feature parameters achieved by rewriting the command and external data files.

Distributed object systems.

Design automation often necessitates repeated execution of modelling and analysis processes, which imposes a heavy computational load. Achieving results in satisfactory timescales can be assisted by distributing computing load across a computing network. In recent years a number of systems, which we have termed "Tool Management Systems" have been developed with the purpose of handling the flow of data and the control of multi-process execution for such distributed activities. Generally, these employ an object-oriented approach and use such mechanisms as CORBA for the inter-process communication and object sharing. Example systems are Lockheed's NetBuilder [16] and Samtech's Boss-Quattro [17]. In the design support system described here, CORBA is used for remote access to the SDRC I-DEAS modeller, and socket-based communication for process sequencing.

3.3 Response Surface method.

The response surface method (RSM) can be described as a collection of statistical tools and methods for forming and exploring an approximate functional relationship between a response variable and a set of design variables. The most widely used form of the functional relationship is a low order polynomial that is referred to as a response surface function (RSF). The response surface method was originally developed to analyse the results of physical experiments and to form empirically-based smooth models of the observed response values, thus any noise that is associated with the experiments is filtered out. However, its general usefulness in approximation processes was realised, and the application field of the method has spread into many engineering areas. Structural optimisation is a particular domain in which the technique has been applied. Although the response surface methods differ according to the fields in which they are used, a general response surface method comprises the following steps: (i) choosing the experimental points (Design of Experiments), (ii) choosing the type of model function (model choice) and (iii) evaluating the modelling error (model error evaluation).

Design of Experiments (DoE) is used to select a small set of data points to construct the RSF [18]. Generally, statistical DoE is performed using a minimum-variance criterion in which all errors are assumed to be random errors or variances. A classification of DoE can be described according to whether experimental points are selected in a regular pattern such as a rectangular shape or in a irregular pattern by spreading the sample points out in the design space, where the former is called a regularly shaped design space, and the later is an irregularly shaped design space. For a regularly shaped design space, the most widely used DoE approaches are Central Composite Design (CCD) [19] and the Box-Behnken method [20]. For an irregularly shaped design space, a computer-generated design is used, of which the most popular design is D-optimal design [21].

4 Research methodology.

4.1 Design automation with a feature-based model.

The basis of the viewpoint-dependent modelling approach is a feature-based design description method in which a design feature model is translated into the representations required for the different viewpoints, omitting features as necessary (e.g. for unwanted detail), and presenting features according to the application domain [1][11]. Each design primitive is defined by a pre-determined number of geometric model entities, which are associated with a set of dimensions that specify the size. For structural analysis, loads and boundary condition fields are associated with geometric model entities. Boundary conditions are normally applied on the face or edge of a feature. The generic characteristics of geometry are captured in the feature class definition, and the feature combination and characteristics (for a particular viewpoint on a particular part) are achieved by assembling a program file to construct the representation, together with data files for the feature parameters required. Any external program may then request the construction of the viewpoint design model by requesting the execution of the program file or modification of the model parameters through the system API that is CORBA compliant. The program file also allows the execution of analysis tasks such as Finite Element analyses, extraction of results and writing of these to a file or an external process. A simple example of such a process is the assembly of a beam (which could be a single feature or parametric part) by: (i) subtracting one block feature from a second, (ii) adding loads and boundary constraints (which are attached to the end faces of the larger block) and (iii) then meshing and executing the analysis, as shown in Figure 1. In this model, the location of the block that is subtracted to form the interior of the section may be varied in two directions to represent the possibility of a core shift in the manufacture of a casting. The values of the offset in the x and y directions are stochastic variables, as are the loads on the end of the section.

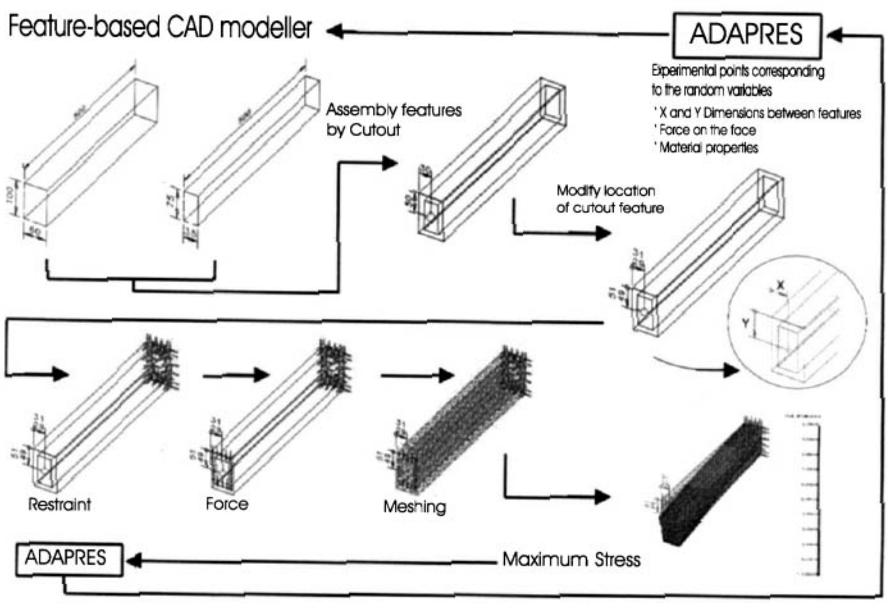


Figure 1. Assembly and finite element analysis of a beam

4.2 ADAPRES, the Response Surface Method tool.

The feature-based modelling approach may be called repeatedly by an external program in order to generate data for optimisation, probabilistic analysis and so on. In the present application, it is used to generate response surfaces that may then be used as substitutes for the objective functions in such applications. Conventional response surface methods used in both probabilistic design and optimisation use a fixed type of a response function such as a linear or a quadratic polynomial. However, for implicit problems in which the relationships of the input variables in the performance function are not explicitly defined, it is almost impossible to assume a fixed type of response function. Therefore, a second-order polynomial has been used widely as the type of the response surface function. However, this approach may not be suitable for problems having either linear or highly non-linear responses. Thus, an adaptive response surface method (ADAPRES) is proposed by Kaymaz [4], in which the type of the RSF is determined according to the response of the system. In this method, both the RSF and the design matrix that is formed to specify the input and output parameters of the system are improved until a set of conditions (set to obtain a well-approximated RSF) are satisfied. In order to determine if both the RSF and the design matrix are adequate to use for a particular application, ADAPRES introduces a method to determine the effect of the input parameters on the system, through a modelling error assessment. By using this error assessment, ADAPRES can form the RSF dynamically according to the response of the system, starting from a linear RSF and adding non-linear terms as required.

The operating sequence of the design support system is as follows. Initially ADAPRES forms a design matrix to identify the initial problem parameters to be used in forming a trial RSF. This matrix is used to request the construction of feature models for trials for the initial matrix entries. Error assessment is then carried out, based on further trial(s), and if necessary additional test points processed to achieve a satisfactory response surface approximation for the response of the system. In the case of the beam model shown in Figure 1, the uncertain parameters for the beam were as noted the x and y offsets of the central void (x and y) and the applied load F. The mean and standard deviations used in the assessment are as shown in Table 1, and for these values ADAPRES gave a response surface function (RSF) for stress analysis as

$$Y = -92.6516 - (0.007296 \times x1) + (0.1877 \times x2) + (1.6676 \times x3)$$

Table 1. Beam parameters used in RSF Formation

	Applied load, $F(N)(x1)$	x-direction offset, $x(x2)$	x-direction offset, $y(x3)$
Mean	1.5E4	30	50
Standard deviation	1E3	1	1

which the unit of Y is N/mm^2. The RSF can be used for the stress analysis of the beam instead of calling I-DEAS for each iteration required by the optimisation algorithm.

ADAPRES_NET

A single RSF is valuable in itself for optimisation or probabilistic analysis, but the greatest value is perhaps when multiple RSFs are used in an analysis network. The evaluation of a real engineering system often involves many different analyses, the interaction and communication between which are crucial in ensuring a satisfactory design. Fatigue analysis, for example, requires results from stress analysis, for which the response surface function of the stress analysis becomes an input for fatigue analysis. More complicated problems such as designing a vehicle involve different analyses such as modal analysis, fatigue analysis, dynamic analysis (in which the output from one analysis may form the input for another), stress analysis and fatigue analysis. For each analysis, a response surface function can be used in place of the performance function established based on the design constraints. In order to carry out different analyses efficiently, each formation of the response surface function can be carried out concurrently in addition to distributing processing to other workstations over the network. The management and control of such problems is achieved by Tool Management Systems such as those described in section 3.2.

In the present work, the adaptive response surface method has been incorporated into these tools, and other enhancements have been incorporated in a computing application called ADAPRES_NET. In this program, the evaluation of multiple performance functions for a structural reliability problem or an optimisation problem (using response surface functions or direct calls to the performance function) can be distributed over multiple workstations on a network. In addition, the evaluation of the approximate response surface performance functions can also be carried out concurrently on the network, which can reduce the time spent on the evaluation of the multiple performance functions of the structural reliability problem. The communication and data transformation between necessary applications are handled by a Petri net-based design process modelling technique [4].

5 Conclusions

This paper has described a design system in which feature-based computer-aided design is combined with a system for forming response surface functions. An adaptive response surface methodology is used to request repeated construction or modification of feature-based part models in order to generate data for the construction of RSFs. By using viewpoint dependent features, different RSFs may be constructed for different aspects of a problem: for example one feature set might allow a part to be represented as kinematic model for the purposes of establishing an RSF for kinematic performance; another model, based on the same underlying design description, may be used to develop an RSF for maximum stress. A very simple application has been demonstrated, but more complex models are being developed to represent for example automotive power train performance. It is proposed that by adopting this combination of feature-based design and the response surface methodology, models of design alternatives may be easily and rapidly constructed at an early design stage for the purposes of sensitivity analysis or the evaluation of design alternatives.

References.

[1] Rosenman, M.A., and Gero, J.S., "Modelling Multiple Views of Design Objects in a Collaborative CAD Environment", Computer-Aided Design, 28,1996 , pp.193-205

[2] Bond, A.H. and Ricci, R.J., 1992, "Co-operation in Aircraft Design", Research in Engineering Design, 4, 1992, pp.115-130

[3] Sundararajan, C.(ed.), "Probabilistic Structural Mechanics Handbook", Chapman & Hall, London, 1994.

[4] Kaymaz, I., "An Adaptive Response Surface Method for Engineering Analysis", PhD Dissertation, University of Bristol, UK, 2000

[5] Shah, J.J., and Mäntylä, M., "Parametric and Feature-based CAD/CAM", Wiley-Interscience, New York, 1995

[6] Arbab, F., "Requirements and architecture of CAM oriented CAD systems for design and manufacture of mechanical parts" PhD Dissertation, University of California, Los Angeles, USA, 1982

[7] Pratt, M.J., "Solid Modeling and the Interface Between Design and Manufacture", IEEE Computer Graphics and Applications, 4, 1984, pp.5259

[8] Bidarra, R., de Kraker, K.J. and Bronsvoorf, W.F., "Representation and Management of Feature Information in a Cellular Model", Computer-Aided Design, 30, 1998, pp.301-313

[9] Rosenman, M.A., and Gero, J.S., "Modelling Multiple Views of Design Objects in a Collaborative CAD Environment", Computer-Aided Design, 28, 1996, No 3, pp.193-205

[10] Rèmondini, L., Lèon, J.C. and Trompette, P., "High-level Operations Dedicated to the Integration of Mechanical Analysis within a Design process", Engineering with Computers, 1998, 14:81-92

[11] Martino, T., Falcidieno B. and Haβinger S., "Design and Engineering Process Integration Through a Multiple View Intermediate Modeller in a Distributed Object-Oriented System Environment", Computer-Aided Design, 30, 1998, No 6, pp.437-452

[12] Kugathasan, P., "A Feature Based Approach to Design Information Management - Multiple Viewpoint Dependent Models for the Product Introduction Process", PhD Dissertation, University of Bristol, UK, 1998

[13] Hoffmann, C.M., and Joan-Arinyo, R., "Distributed Maintenance of Multiple Product Views", Computer-Aided Design, 32, 2000, pp.421-431

[14] Bodkin, M.E., Bajorek D., and Prasad, M., "Feature-based Structural Design of Solid Automotive Components", Finite Elements in Analysis and Design, 16, 1994, pp.15-26

[15] Liu, X., "CFACA: Component Framework for Feature-Based Design and Process Planning", Computer-Aided Design, 32, 1999, pp.397-408

[16] Dabke, P., Cox, A., and Johnson, D., "NetBuilder: an Environment for Integrating Tools and People", Computer-Aided Design, 30, 1998, pp.465-472

[17] Samtech, "BOSSquattro", http://www.samcef.com/pdf/boss-va.pdf

[18] Box, G.E.P., Draper, N.R., "Empirical Model-Building and Response Surfaces", John Wiley & Sons, New York. 1987

[19] Myers, R. H. and Montgomery, D. C., "Response surface methodology: process and product optimisation using designed experiments", John Wiley & Sons, New York. 1995

[20] Rajashekhar, M.R., and Ellingwood, B.R.. "A New Look at the Response Surface Approach for Reliability Analysis", Structural Safety 1993;12(3):205-220

[21] Kim, S., and Na, S., "Response Surface Method Using Vector Projected Sampling Points", Structural Safety, 19, 1997, pp.3-19

Mr Chris McMahon
University of Bristol, Department of Mechanical Engineering, Queen's Building, University Walk, Bristol BS8 1TR, United Kingdom, Tel: +44 117 9288100, Fax: +44 117 9294423, email: chris.mcmahon@bris.ac.uk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

WEB-BASED DESIGN AND MANUFACTURING OF CUSTOM MANNEQUIN MODEL

T K K Chang, C C L Wang, and M M F Yuen

Keywords: Rapid Manufacturing, Computer-Aided Design, Parametric Design, And Feature-based Design

1 Introduction

New techniques in information technology are now changing our daily life. With the rapid development of Internet, web sales become a common practice of many peoples life. However, the lack of fitting is the major defeat of garment sale on Internet, which hesitates the user in mail ordering. The current work aims at providing an efficient ways for construction of mannequin model representing the body shape of individuals for virtual garment fitting on Internet. Using the semantic feature network [1] and multi-resolution representation [2], a feature-based human model is constructed. Sixteen features are defined to represent human features such as head, neck and torso of a human body. To satisfy the requirement of virtual garment fitting, a set of metric constraints, representing the body measurement used by tailors, is defined on each features of the model. A parametric design tool is designed and implemented to adjust the size of human model to satisfy the metric constraints. Using the Component Object Model (COM), users can select their body type and input their body dimension on Internet. The parametric design tool will be used to adjust the human model to satisfy the body dimensions, giving the customized human model in VRML format for immediate verification. The model can then be used in garment fitting on web.

Besides virtual garment fitting, the proposed system can also be applied to mannequin manufacturing with the use of layered-manufacturing technology. A large-scale rapid-prototyping machine is developed to manufacture the mannequin model. Using the digital model created, the model can be built in short period of time, normally two days. The model can then be used in garment manufacturing industry for garment fitting. This reduced the cost for custom mannequin manufacturing for garment industry

The above system has been implemented and example will be given in this paper.

2 Mannequin Design

Recently, the modelling of human model has been a hot topic in computer graphics and computer aided design. Different modelling tools and representation schemes have been applied to construct digital human model. Among those tools, feature-based technique has aroused attention due to its flexibility and reusability of the features. Besides, the effectiveness of the feature-based modelling and constraint integration, e.g. parametric design, is also a major concern. However, convention feature definition such as graph [3], rule-based [4] and syntactic method [5, 6] are limited to the domain of regular object and therefore not suitable for modelling of human features which are sculpture in nature.

Unlike regular shape objects, only a few feature representation schemes [7,8,9,10,11] are suggested for sculpture object domain. Among those schemes, Semantic feature network proposed by Au [11] is applied owing to its ability to describe the relations with its neighbouring features and, therefore, to maintain continuity between features.

2.1 Feature definition and Representation

A semantic feature representation consists of semantics and geometry. Semantic describes the relations with its neighbouring semantic features within the model and the associated application specific information. The geometry includes geometrical and topological information of a semantic feature. The semantics and geometry of an object are represented by a semantic network (fig. 1), a graphical representation of a semantic feature model.

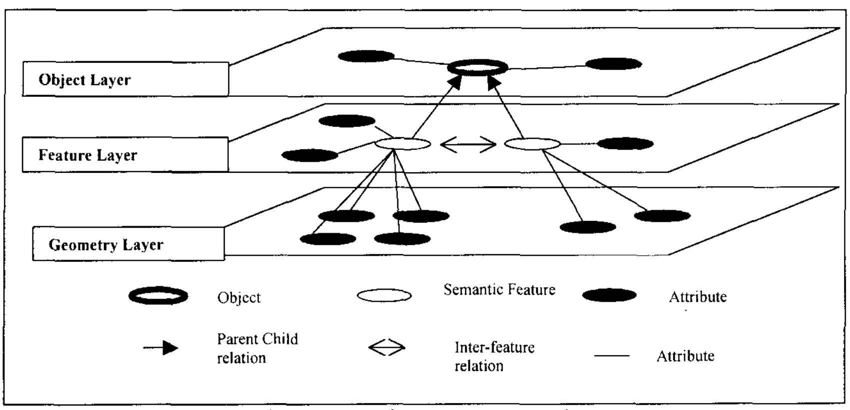


Fig 1. Semantic Feature Network

To create a feature-based model, the definition of the feature must be performed. Based on the anatomical data and study from the fashion designer, a natural physiological partition of human model was done which sub-divided the human model into several regions called "basic shape region". Features are defined to represent the basic shape region including head, neck, upper and lower torso, left and right upper and lower arms, left and right upper and lower legs, left and right palms and foots. The following is the feature representation of the human model

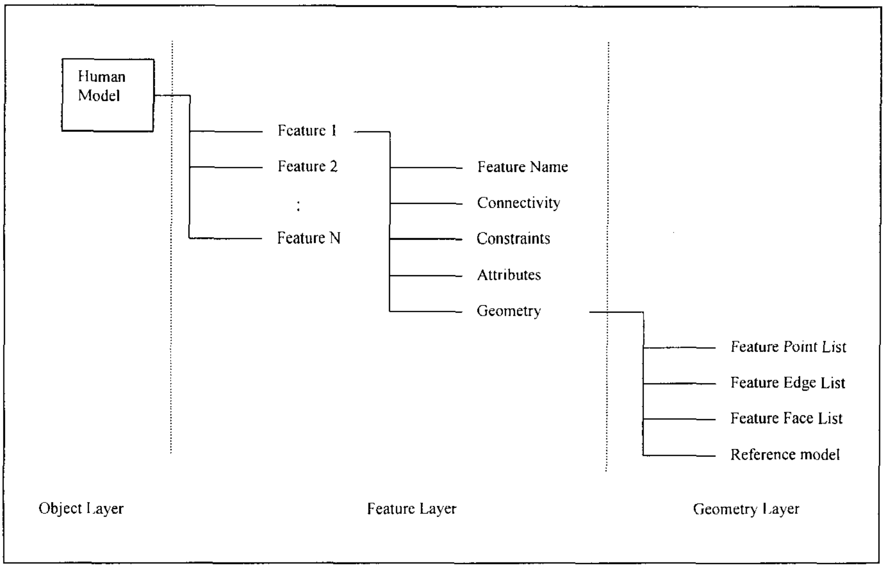


Fig 2. Semantic Feature Representation of Human feature

Using the semantic feature representation scheme, three-layer-structure is defined for the sculpture human model. The object layer consists of the information about the domain of interest and its specification, which is human model in our case. In the feature layer, the semantic of each feature are defined. These include the feature name, connectivity, constraints, attributes as well as geometry information.

Having defined the semantic of each feature, we can then define the topological and geometry structure in the geometry layer. With multi-resolution mesh representation, we defined the base face, base edge and base point (feature point) of each feature. The feature points are located on the region where the metric constraints are defined, which also act as control points for shape and metric adjustment for each feature. This forms the basic model as shown in Fig3.

Sub-division of surface is used to refine the feature to satisfy the required continuity and the tolerance requirement for each feature. The following figure shows the topological structure and appearance of a feature before and after mesh sub-division.

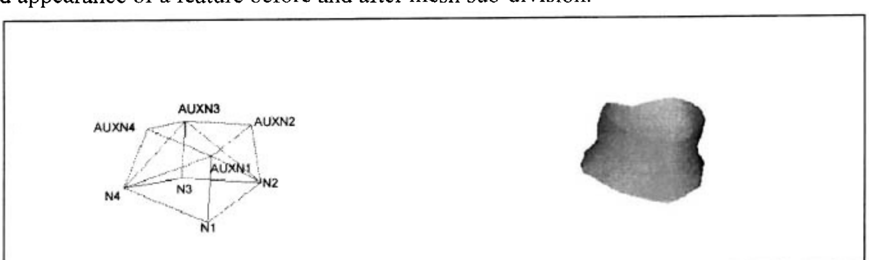


Fig 3. Topological and multi-resolution representation of neck feature

After construction of the feature library, custom human model can be built using the feature in the feature library, which best represent their body shape. Continuity between features is well maintained to form a smooth model as shown below.

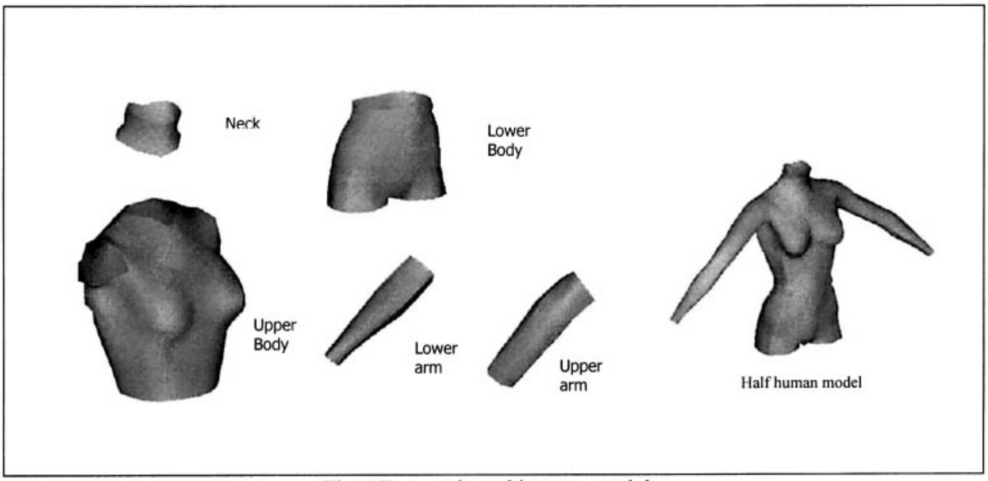


Fig 4 Feature based human model

2.2 Parametric Design

In the feature-based model, more than twenty metric constraints are defined. The constraints can be classified into two main categories including height and girth constraints. In each of the feature, the feature points can be adjusted by a set of pre-defined rules to satisfy the metric constraint and refinement is performed to ensure the intra and inter-feature continuity. The following shows the results of parametric design of a male mannequin model

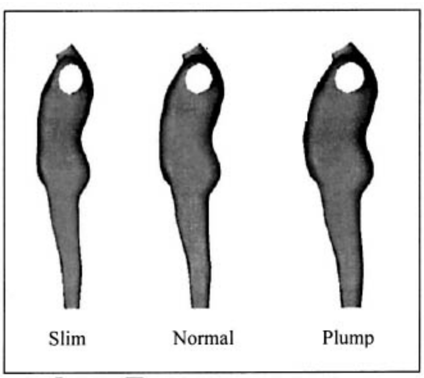


Fig 5. Parametric Design of mannequin model

3 Mannequin Manufacturing

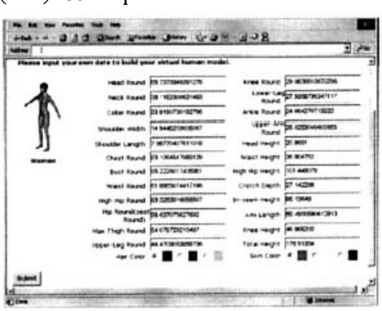
In the previous section, the model representation and parametric have been introduced which allow user to build their custom mannequin model for virtual fitting on Internet. However, only virtual model may not be sufficient in some cases such as window display in shops and garment fitting in factory. In such cases, physical models are required. To produce the custom mannequin model, Layered-manufacturing (LM) has been adopted which build the model layer-by-layer repeatedly. There are numbers of commercial LM process are available on the market. These include Stereo Lithography (SL) from 3D Systems, Laminated Object Manufacturing (LOM) from Helisys, Selective Laser Sintering (SLS) from DTM, Solid Ground Curing (SGC) from Cubital, Fused Deposition Modelling (FDM) from Stratasys, and Sander Model Maker (MM) using ink jet technology. However most of the processes suffer from high cost, long building time and limited to small object size, which is not capable of building mannequin model.

To allow to construction of mannequin model, a large-scale LM machine, using hot wires and foam, is built. Using the custom model constructed, the tool-path of the hot wires is calculated by repeatedly intersecting the mesh model by a series of parallel cutting planes normal to the building orientation. The result of physical model will be shown in the following section.

4 Implementation

The above feature-based human model has been implemented using Microsoft Visual C++. A component for human modelling, parametric design and tool path generation has been built using component service. The component is running on Microsoft Transaction Server. With the web-based software, user is able to choose their body type and input their body dimension into the server. 3-Dimensional model in VRML format can be viewed by user immediately through a standard web browser. The model can then be used for virtual garment fitting on Internet. Besides, if physical model is required, manufacturing can be done by using the prescribed layered-manufacturing machine developed.

The following figures show the parametric design result and the 3-dimensional model in VRML format after adjustment and the manufactured model, using Layer Manufacturing (LM) technique



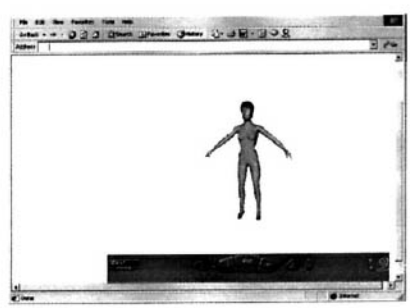


Figure 6 & 7 Virtual mannequin model construction and 3-Dimensional custom model on Internet

Figure 8. Full-scale custom size mannequin model using Layered-manufacturing

5 Discussions

In this paper, a feature-based human model is constructed using semantic feature representation. The merit of the feature-based modelling lies in the reusability of the feature library to represent different type of model. Once the topology and semantic of the feature are well defined, the different types of shape of feature can be obtained by altering the geometric information. In our case, once a human feature, such as upper torso, has been constructed, different shape of the same features can be obtained by altering the geometric information – feature points. The topological structure and relationships with other features can be reuse. As a result different body shapes can be represented by using the same feature model effectively.

In comparison with other modelling method such as laser scanning or image capturing, our system doesn't require a standard set up for data acquisition and hence more suitable for web-based application. However, due to the limitation in the feature library and lack of information in parametric design (fig 9), the model constructed may not truly represent individual body figure although the parametric design guarantees the body dimensions for reasonable accurate virtual fitting.

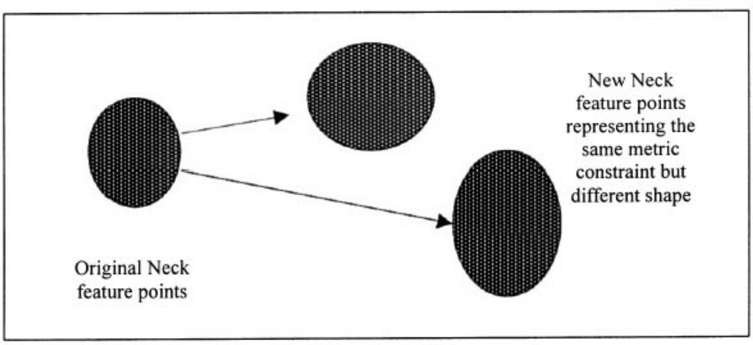


Fig 9. Cross-section shape changes in parametric design

In mannequin manufacturing, LM techniques offer an effective tool to consult large physical model. However, the technique results in staircase effect for each layer manual smoothing of the foam model is required, which lengthen the processing time and reduce the accuracy of the model. The effect can be minimized by reducing the layer thickness of the material with the trade-off of manufacturing time.

6 Conclusion

In this paper, an efficient method for construction of virtual human model on Internet is presented. The user is able to select their body type and input body parameters to obtain a digital model representing their body shape. The constructed model is in triangular mesh format, which can be used for virtual garment fitting on Internet. This provides a convenient tool for E-commerce of garment related items on Web.

Moreover, the integration of the system with large scale LM machine is capable of manufacturing the customized human or mannequin model for use in manufacturing environment in which real fitting is required. This greatly reduces the cost of custom mannequin manufacturing.

Reference

[1] C. K. Au, "Semantic Feature Modelling for Sculpture Objects", Ph.D. Thesis, Hong Kong University of Science and Technology, Hong Kong, 1998.

[2] Kobbelt L, Campagna S, Vorsatz J, Seidel H-P., "Interactive multi-resolution modeling on arbitrary meshes.", Computer Graphics Proceedings, SIGGRAPH 98, pp.105-14.

[3] Joshi, S. and Cheng, T.C., "Graph-base Heuristics for Recognition of Machined Feature From a 3D Solid Model", Computer-Aided Design, vol. 20, no.2 pp58-66

[4] Haralick, R. M., Elliot, G. L. "Increasing tree search efficiency for constraint satisfaction problems", Artificial Intelligence 14:p 263-314, 1980

[5] Henderson, M. R. and Anderson, D. C., "Computer Recognition and Extraction of From Feature: A CAD/CAM Link", Computers in Industry, vol. 5, pp 329-339, 1984.

[6] T. De Martino, B. Faleidieno and F. Petta, "Classification of generic shape features for the recognition of specific feature instances". Appeared in CAD Systems Development: Tools and Methods. D. Roller P. Brunet(eds.) pp 143-155, 1998

[7] Cavendish, J.C., "Integrating Feature-based surfaces Design with Free-Form Deformation", Computer-Aided Design, vol. 27, No. 9,1995

[8] H.K. Toshoff, J.C. Aurich, M. Ehrmann, N.D'Agostino, "A Unified Approach to Free-Form and Regular Feature Modeling", in Annals of the CIRP Vol. 45/1/1996, pp125-128

[9] Mitchell, S. R. Jones, R. and Newman, S.T., "Feature-based systems for Design and Manufacturing of Sculptured Product", International Journal Production Research, vol31, no.6 pp 1441-1452, 1993.

[10] Mitchell, S. R., Jones, R and Hinde, C., "An Initial Data Model, Using the Object-Oriented Paradigm, for Sculptured-Feature Based Design", Research in Engineering Design, pp 19-37, 1995.

[11] C. K. Au, M. M. F. Yuen, "A Semantic Feature Language for Sculpture Object Modelling", Computer-Aided Design, 32, pp. 63-74, 2000

Mr Terry K. K. Chang
Mechanical Engineering Department, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Tel: (852) 2358-8650, Fax (852) 2358-1543, email meterry@ust.hk

Mr Charlie C. L. Wang
Mechanical Engineering Department, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Tel: (852) 2358-8650, Fax (852) 2358-1543, email wangcl@ust.hk

Dr. M. M. F. Yuen
Mechanical Engineering Department, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Tel: (852) 2358-8650, Fax (852) 2358-1543, email meymf@ust.hk

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

6DOF MANIPULATORS FOR CURVES AND SURFACES BI-MANUAL MODELLING IN VIRTUAL ENVIRONMENT

M Fiorentino, G Monno, and A E Uva

Keywords: Virtual Reality and Prototyping, VR user interfaces, Curve and Surface Design

1 Abstract

We propose an original software module for the direct manipulation of Non-Uniform Rational B-Splines (NURBS) curves and surfaces using a Virtual Reality (VR) based interface. NURBS describe curves and surfaces in parametric fashion and are difficult to be modified and controlled interactively in a traditional desktop environment due to their three-dimensional nature. VR offers the possibility to visualize and modify the several degrees of freedom of a complex NURBS in a more efficient and user-friendly way. We suggest a new manipulation paradigm called Dynamic Weighted Manipulation (DWM). In our interface, the user selects special virtual tools able to pick and modify several control points at the same time. Each tool has a different function and geometrical representation as a proxy always following the user's hand in position and orientation. The tools allow the user to select simultaneously several control points according to relations dynamically defined.

2 Introduction

Virtual Prototyping is a technique that uses VR to simulate appearance, functionality and behaviour of products not yet physically realized. The possibility of real time interaction on current design to evaluate the impact of the changes reduces significantly the need for mock-ups. For this reason the most recent attitude in CAD software is to try to integrate this powerful tool inside the design and manufacturing cycles, where the creative phase is followed by several trial-and-error iterations. Unfortunately, VR techniques are still not in a mature phase for this aims and most of the research nowadays is focused on the man-machine interface to fully employ the potentialities. Visualization in an immersive virtual environment, six-degrees-of-freedom (6DOF) manipulation and gesture recognition require a specific development for methods and tools focused on the specific application. The user must concentrate on the actual designing task, not on the techniques used to achieve the results as he currently does in a traditional CAD due to the bi-dimensional visualization and mouse and keyboard based input.

The goal of this work is to develop, at least for a particular application, techniques and manipulation tools in virtual environment to define a more "transparent" man-machine interface.

2.1 Related work

VR based CAD programs can be divided into two categories based on the type of modelling they perform: free-form surface manipulation or solid modelling. Commercial CAD packages contain both these modelling types. Solid modelling is the construction of a closed boundary object using a set of Boolean construction operations and a base set of primitives. Solid modelling is very efficient for creating models of object that are composed of flat surfaces, sharp edges, and curves of constant radius. Examples include engine connecting rods, linkages, frames, and casings. VR solid modelling programs focus on providing a faster interface for setting up Boolean operation parameters and primitive scaling information.

Free-form surfaces are commonly used to represent shapes that cannot be created with Boolean operations performed on primitives [1], as car bodies, ship hulls, appliance housings, and turbine blades. Free-form shapes are generally curves or surfaces with non-constant curvature and stringent requirements for surface properties and continuity. These surfaces are most often stored as polygonal or parametric surface meshes, and require the user to manually adjust large number of control points and associated weights in order to design shapes. VR free-form surface manipulation programs generally have the goal of granting the user control similar to that found when manipulating deformable objects in the physical world. Interaction paradigms tend to mimic the effect of sculpting stone or clay. Galyean and Hughes [2] created an early 3D-input sculpting program that used a voxel representation. 3-Draw [3] uses a 6DoF input device to create 3D curves. One traces out splines in 3D space and the other controls the orientation of the modelling world. The goal of the application is to generate splines that could eventually be meshed into surfaces and used in the early stages of conceptual design. The use of a 6DoF interface means that the user can create 3D shapes without performing the 3D to 2D mapping required when using a desktop mouse. Furlong [4] created a Bezier free-form deformation based sculpting program for use in a CAVE type virtual reality environment. The workpiece is a polygon surface imbedded in a bezier volume. Users push or pull individual points on the surface and the bezier volume provides spline-based continuity to the polygon surface. The program is designed for sketching free-form shapes during the conceptual design stage. The single point interaction method does not provide enough control for the modification of production CAD models. Perles and Vance [5] presented some virtual tools, which are used to operate directly on the CAD data and change the shape of the NURBS surfaces in a virtual environment. The virtual tools can be dragged along or pushed into the surface in real-time to change its shape. The surface normal at an arbitrary point can be manipulated in three-dimensional (3D) space. Constraint-based surface manipulation is used to obtain multiple point direct manipulation of NURBS surfaces. Further research and implementation is done by Dani et al. [6]. Their shape-manipulation technique is based on Indirect interaction on a 3D projection wall, by moving a set of control points (which can be a row, a column, or a rectangular array), previously selected using a virtual ray beam. Guillet and Léon [7] describe a very efficient method to modify free-form surfaces using an analogy between the control polyhedron of each surface and the mechanical equilibrium of a rigid bar network. The provided mechanical model allows parametric deformations maintaining continuity conditions and respecting geometrical constrains provided by the user (also with non linear tangency).

2.2 NURBS

Non-uniform rational B-splines (NURBS) have become the industry standard for representation of free-form curves and surfaces. A point on a NURBS surfaces with parametric coordinates (u,v), is evaluated using a net of control points (CPs) P, of weights w, and of basis functions N

$$S(u,v) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (1)$$

Where, p and q represent the degrees and $n+1$ e $m+1$ define the number of control points in the two parametric directions u e v. The values of u and v range between 0 and 1.

3 Manipulation Paradigm

Manipulation methods for tree-dimensional NURBS curves and surfaces are categorized by the way a designer interacts with the surface. Since CPs generally do not lie on the NURBS itself, the adjustment of the CPs is called ***indirect manipulation***. In ***direct manipulation*** the user moves points selected on the curve or surface and mathematic algorithms update the CPs mesh. Manipulation can further be divided into ***single point manipulation*** and ***multiple point manipulation*** according to the number of CPs moved at the same time. Single point manipulation lets the designer only edit a single point on the surface at a time and uses the properties of NURBS to determine how the rest of the surface responds. Therefore, the perturbation on a surface is narrow and localized. Multiple point manipulation allows the user define a new shape for an area of the surface. The user is still limited by the properties of the surface, but a higher level of control is available. Multiple point manipulation does require a longer set-up time because multiple points have to be specified.

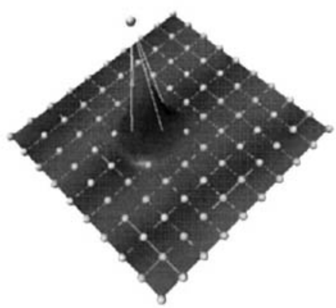


Figure 1. Single Point Manipulation.

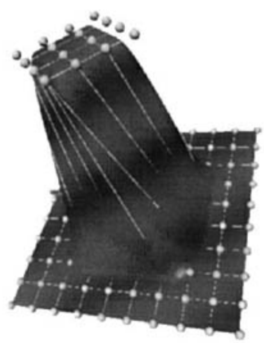


Figure 2. Multiple Point Manipulation.

3.1 Rigid Transformations

The simplest way to perform multiple point manipulation is to consider a selected number of CPs as parts of a solid. Moving interactively the selected CP, the Kinematics of Rigid Body's laws evaluate the transformation of all the other CPs. Ribs, extrusions, holes and others modelling operations are easily performed simply varying the pattern of the moved CP. Rigid transformations involve an heavy use of operations and compositions among multiple local transformation. Rotation about arbitrary axes is tricky and the problem gets worse when we try to interpolate smoothly between orientations. Rotation matrices are extremely hard to use for interpolating rotations between two orientations as involves numerical integration, which can be computationally expensive. Euler angles also introduce the problem of "Gimbal lock" or a loss of one degree of rotational freedom. Quaternions do not suffer from Gimbal lock, because they do not express rotations relative to three separate axes. The second argument that is commonly given for the use of quaternions is the possibility of smoothly interpolating between two orientations. We use to store orientations quaternions, which can be used to represent rotations about an axis in 3D just like rotation matrices. The advantage to using them is that they describe the same amount of information in four parameters, whereas the rotation matrix uses nine to describe three degrees of freedom. This over-description yields larger numerical inaccuracy and faster error build-up. The four parameters are described as a scalar and a vector given by:

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \mapsto \mathbf{q} = [w, (x, y, z)] \qquad \text{or} \qquad \mathbf{q} = \left[\cos\frac{\theta}{2}, \mathbf{v}\sin\frac{\theta}{2}\right] \tag{2}$$

where **q** is the quaternion, **v** = (x, y, z) is a vector and w is a scalar. Quaternions allow easily combination operations including linear interpolation. The classic rotation matrix can be computed from the quaternion for graphics purposes using the following formula:

$$\mathbf{q} = (w,x,y,z) \mapsto \mathbf{M} = \begin{bmatrix} w^2+x^2-y^2-z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & w^2-x^2+y^2-z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & w^2-x^2-y^2+z^2 \end{bmatrix} \tag{3}$$

4 Virtual Environment

We propose a new modelling paradigm for NURBS curves and surfaces in virtual environment, using six-degrees-of-freedom input and bi-manual interaction. Complex operations like modifying three-dimensional curves and surfaces can be achieved in virtual environment with simple and intuitive actions, borrowed from daily behaviour, like modelling a thin aluminium plate. We designed and tested a module for 6DOF NURBS manipulation inside a framework for interaction in a virtual environment called ***3DIVS*** *(3-Dimensional Immersive Virtual Sketching)* already developed by the authors in the past years [8]. For optimal graphics performances 3DIVS is implemented on SGI Octane/SI (MIPS®R10000 a 225MHz). The rendering pipeline ensures a refresh rate adequate for stereo rendering (minimum 30 Hz per eye). A shutter-glasses system Crystaleyes® by StereoGraphics is used to synchronize the two images while the positions of the user's head and hands are tracked using 3SPACE FASTRAK™ (6 DOF) by Polhemus. The PinchGloves system by Fakespace handles gesture recognition for fingers contact. 3DIVS uses an Open Inventor GUI front-end. The ***6DOF_NURBS_Manipulation*** module is implemented in C++ and is able to manipulate even very complex NURBS curves and surfaces.

Figure 3. Virtual environment hardware setup.

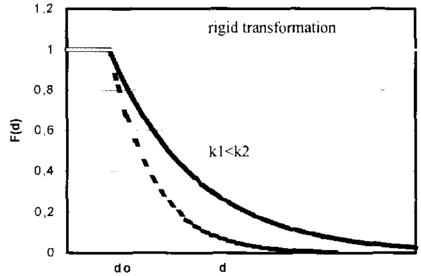


Figure 4. Influence function *F(d)*.

5 Our contribution: Dynamic Weighted Manipulation (DWM)

We defined a multiple approach interaction technique based on an asymmetrical performance for the two hands according to hands position and fingers contact. Ideally, a designer should not need to worry about the details of the control points but rather should be allowed to directly manipulate the surface. However, in this work we decided to avoid direct manipulation with its time-consuming update routines to always guarantee real time interaction. We focussed the attention on multiple points indirect manipulation and the relation constraints between the transformations of the CPs. After experimenting several design cases using single and multiple point manipulation the results show that the user needs both modes. Moreover, in the multiple point fashion a rigid deformation is not always appropriate. Therefore, we decided to blend the transition between a single and a multiple point rigid manipulation by means of a dynamically changing transition function.

We defined a new type of manipulation called Dynamic Weighted Manipulation (DWM) where all the CPs of the NURBS are influenced by the transformation T* of the single selected CP*. Using quaternions we can compute the transformation of any CPi as a linear combination of T* and the transformation null according to an influence factor F (ranging between 0 and 1) called influence which is a function of the distance d between CP* and CPi. We defined and evaluated several mathematical descriptions for the $F(d)$, as well several interpretations of the distance d.

The distance d can be estimated between CPs as a Cartesian distance $d_1(x,y,z)$ or a distance on the parameter space $d_2(u,v)$. It is evident that the d_2 will not change even after previous movement while d_1 will assume different value after any change

$$d_1(p,i) = \sqrt{(x_p - x_i)^2 + (y_p - y_i)^2 + (z_p - z_i)^2} \quad (4)$$

$$d_2(p,i) = \sqrt{(u_p - u_i)^2 + (v_p - v_i)^2} \quad (5)$$

---

where p is referred to CP* and i to the generic CPi. The influence function is defined as following:

$$F(d) = \left(1 - e^{-k(d-d_0)}\right) \qquad (6)$$

where d is the distance evaluated according to (4) o (5). d_0 is a minimum threshold defining that a rigid transformation is applied to all the points closer than d_0. k is a parameter influencing how the distance reduces the F (Figure 4). The values of d_0 and k are dynamically changed in real time using the second hand therefore allowing the user to evaluate the best influence function for each case.

6 Manipulators

Picker

This is the basic manipulator unit, as it provides the fundamental function of changing the CPs position according to the x,y,z coordinates of the proxy. Both hands can activate the picker simply pinching an idle CP with the thumb and the index. During the "picking mode", we can modify the values of d_0 and of k, both parameters of the $F(d)$ simply pinching the index and the thumb of the other hand and varying the distance of both proxies. This is a very intuitive and efficient way to control the transition between single and multiple point manipulation and to experience dynamically the effects of modelling.

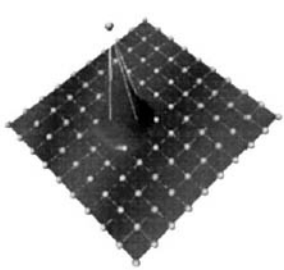


Figure 5. Picker on Single Point.

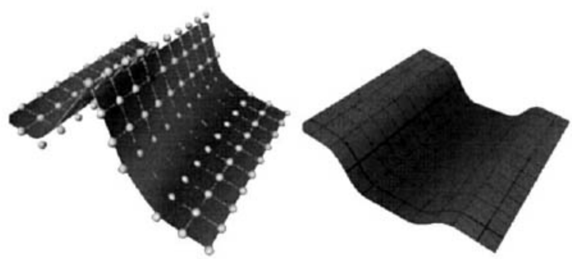


Figure 6. Picker on Multiple Points.

Twister

Twister is an enhanced picker as it provides an association to the 6DOF of the proxy and it implements the aforesaid DWM. It is easily activated with one hand, pinching a selected CP with the index and the middle. The other hand is used to modify in real-time the values of d_0 and of k, both parameters of the $F(d)$, pinching respectively the index with the thumb or with the middle. The variation in the distance between the two hands is reflected into changes for the parameter. The result is similar to holding and twisting a piece of cloth. If a CP on the surface is twisted, the points nearby will "feel" the touch, moving accordingly the function F.

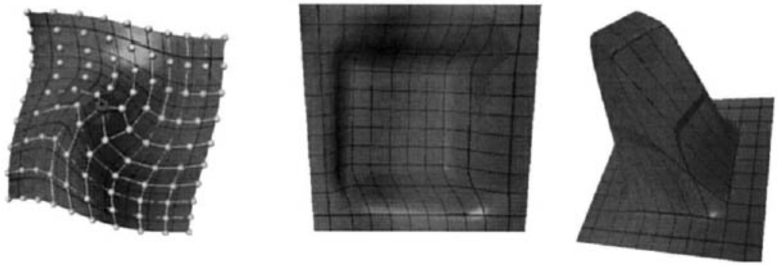


Figure 7. Twister manipulator.

Weighter

This manipulator is similar to the twister as concerns the parameter control, but instead of modifying the values of d_0 and of k the two hands distance is used to change interactively the value of the weight w of the surface's CPs. All the weights for the other CPs change according to the function F. The Weighter is activated by touching the index with the ring in the selecting hand.

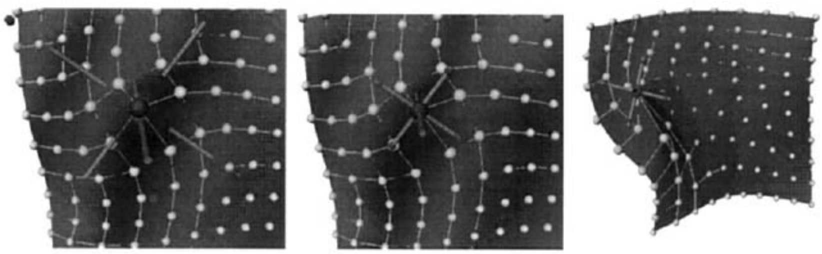


Figure 8. Weighter manipulator.

7 Conclusion and future work

We evaluated the behaviour of these manipulators on both NURBS curves and surfaces of different complexity. We also tested the response of different users both experienced and novice with VR. Everybody is immediately able to "feel" the presence of the curve or surface and feels very comfortable switching among the different manipulators. In addition, the proposed interface to change dynamically the values of the parameters in the influence function has revealed to be robust and user-friendly.

This work shows clearly the advantages of virtual environments in free-form modelling over traditional desktop environments. According to us the main benefits are in order of importance: the bi-manual interaction, the tree-dimensional input and the stereoscopic visualization.

A direct manipulation on the points of a surface, still not available in real time for very complex models, is definitively a key point, which will be guaranteed in a very next future due to the rapid hardware development.

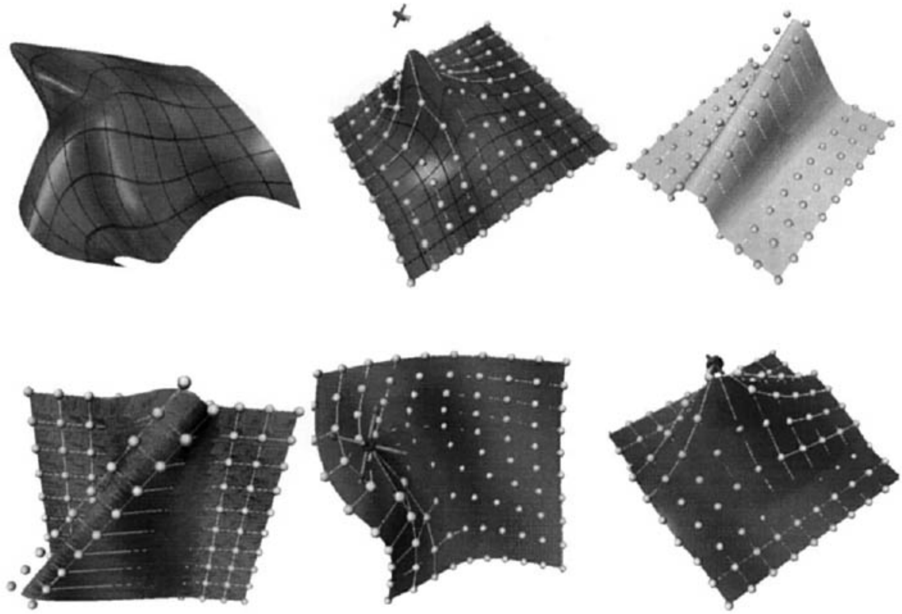


Figure 9. Manipulation examples.

References

[1] Sederberg, T. W., 1986, "Free-Form Deformation of Solid Geometric Models," Computer Graphics, Vol. 20, No. 4,August, pp. 151-160.

[2] Gaylean, T. A., and Hughes, J. F., 1991, "Sculpting: An Interactive Volumetric Modeling Technique," Computer Graphics, Vol. 25, No. 4, pp. 267-274.

[3] Sachs, E., Roberts, A., and Stoops, D., 1991, "3-Draw: A Tool for Designing 3D Shapes," IEEE Computer Graphics & Applications, Vol. 11, No. 6, pp. 18-26.

[4] Furlong, T. J., 1997, "Virtual Reality Sculpture Using Free-Form Surface Deformation," 1997 ASME Design Engineering Technical Conferences, Sacramento, CA, Sept. 14-17.

[5] Perles, B.P., Vance, J.M., 1999, "Interactive Virtual Tools for Manipulating NURBS Surfaces in a Virtual Environment", ASME Symposium on Virtual Environment for Manufacturing, Chicago, Illinois, Nov. 1-2.

[6] T. H. Dani, L. Wang, R. Gadh., 1999, "Free-Form Surface Design In A Virtual Environment", Proc. of ASME '99 Design Engineering Technical Conferences, ASME.

[7] S. Guillet and J.C. Léon, 1998, "Parametrically deformed free-form surfaces as part of a variational model", Computer Aided Design, Vol. 30, pp. 621-630.

[8] F. Kuester, A. E. Uva, B. Hamann, G. Monno, 1999, "3DIVS: 3-Dimensional Immersive Virtual Sketching", ICED 99 Munich.

Michele Fiorentino, DIMeG, Politecnico di Bari, Italy, Tel. +39 080 5962774, Fax. +39 080 5962777, e-mail: fiorentino@dimeg.poliba.it, URL: dimeg.poliba.it/~vr3lab.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

AN ASSEMBLY TOOL TO SUPPORT THE DESIGN OF COMPLEX PRODUCTS

L Bongulielmi, S Dierssen, U Leonardt, and M Meier

Keywords: Design for assembly; Optimisation techniques; Digital Prototypes.

1 Abstract

The traditional process during embodiment design is mainly based on 2D- assembly-drawings and text documents. The contents for these documents depend on the individual knowledge of the employees in the design and assembly department. Many iterative steps are needed between the designer and the assembler in order to generate the necessary detailed assembly documents. This process depends on the individual experience of the employees and cannot guarantee consistent data management.

Even with the use of 3D-CAD Systems in the mechanical industry, this situation has not changed yet. However, by the efficient reuse of 3D models and the use of current IT (Information Technology) products, the assembly process of complex machines can be significantly improved. This paper presents an approach how to use standard VR (Virtual Reality) technologies [1], [3], [5], [7] with 3D-CAD models to support the assembly process.

In an early stage of design, assembly tests [10] can be made to show both, possible design errors as well as complex and very expensive assembly sequences. The ideas of the designers can be easily visualized by the assembler using VR-technology. The communication between the design and assembly department are improved and lead to a continuous exchange of knowledge and experience [9]. Moreover the assembly sequences can later be used in other departments as sales, maintenance or training.

All data are stored in a Product Data Management (PDM) system which ensures correct and updated information management. The use of standard VR technology, for example VRML (Virtual Reality Modelling Language), enables a cost-efficient software solution in comparison to other IT products.

2 Introduction

The traditional assembly process in the mechanical industry is based on 2D – assembly drawings and text documents. Figure 1 shows a general form of the usual process of creating assembly instructions. The process can be divided into several iterations. The demonstrated task allocation for design and production planning can vary in the different companies.

The development of the necessary assembly documents starts with the design that defines the geometrical form, structure and functionality of the product. In a series of iterations the specific knowledge of the assembler is added to the assembly documents during prototype and pre- production series assembly. Only during the last iteration - the series production – are documents for assembly completed.

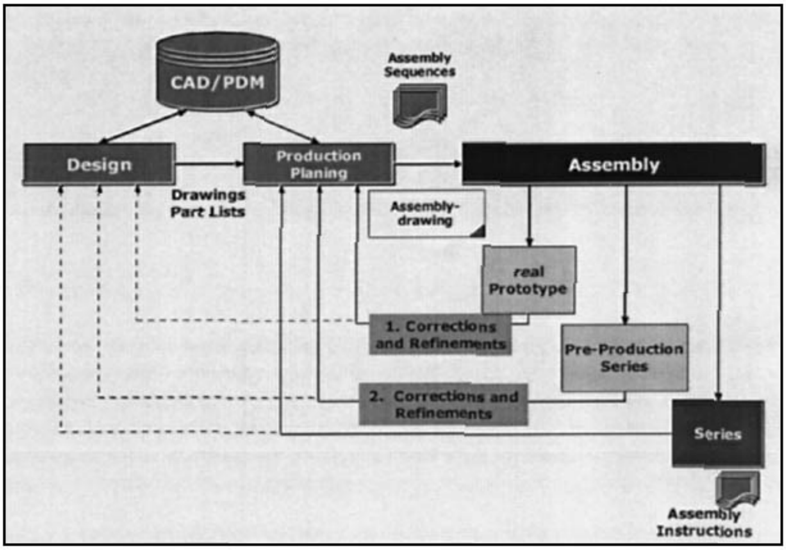


Figure 1. General form of the usual process of creating assembly instructions

The analysis of the assembly process demonstrates that the generation of the assembly instructions is an iterative process that takes place at a late stage of product design. In addition, information concerning the assembly process is defined twice. First of all information is defined by the designer who also has to keep in mind assembly aspects, and by the assembler himself. In order to reduce this ineffective double work, the knowledge and experience between assembler and designer have to be transferred constantly in both directions.

3 The virtual assembly

The use of 3D models and user friendly software can basically help to solve the communication problems between the design and the assembly department. Our proposed concept with the integration of VR-techniques is presented in Figure 2 and described below.

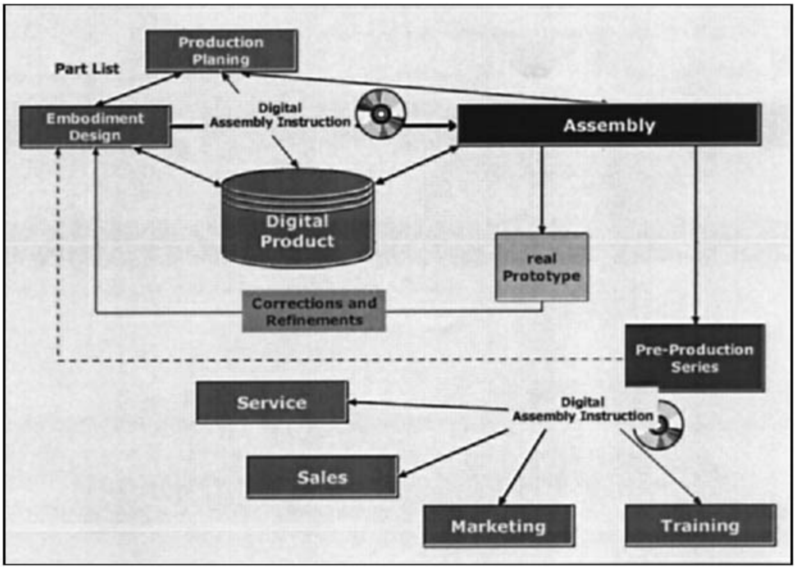


Figure 2: VR- supported assembly process

3.1 Communication platform for assembly & design

The objective of the new concept is to create a communication platform between the design and assembly departments. The core element is built by a PDM system, that can support and manage the assembly drawings and part lists. All drawings and part lists can be generated by the design department and directly be transferred to the assembly department. Furthermore, the process of assembly is replaced by 3D assembly animation. The designer has to transfer the assembly models of the CAD into animation software. The software enables him to animate the CAD-models with several features including assembly or disassembly of the model. Moreover, the animation sequences can be transferred to the assembly department and vice versa using the PDM functionality.

3.2 The virtual assembly process

The generation of the assembly instructions is based on the data of Digital Product. The Digital Product [12] is defined by all information that represents the product and occur during the product life cycle. Within early stage of design this information is typically 3D CAD data, sheets and text documents that describe the characteristics of the product as well as the assembly instructions itself.

The assembly instructions that are forwarded from the designer to the assembler are based on digital data only. This means that no parts have been ordered nor produced when the assembler checks and revises them. Errors in construction that appear at this stage can easily be reduced at low cost.

In addition to the graphic assembly sequences the assembler typically adds written instructions which now can be recorded digitally and related to the appropriate assembly step.

The complete assembly instructions including comments and changes is re- integrated into the design department that corrects it accordingly. This process is repeated for each part group until all assembly instructions are completed.

Virtual assembly-based instructions can help the designer and the assembly planner team define the different working steps and recognise problems like insufficient assembly space, but there will still remain new problems arising the real assembly process. For this reason, feedback from assembly to design remains necessary during series production in order to make further corrections and improvements.

3.3 Reuse of assembly animations

Our last aspect within the presented concept aims at reuse of the assembly animations.

Single assembly steps are stored separately and are related to the geometry objects by the PDM system. This allows for quick modifications of the single assembly steps to create other applications such as service instructions. The assembly sequences can, for example, be shown in the opposite direction and thus become disassembly instructions. For company sections such as sales, marketing, service or internal training these animations can easily be used to create product visualisation.

In addition, new developments of the future can also profit from this new concept. As the animation sequences can be divided into sub-sequences they can be formed themselves according to a new part group configuration [1] and be reused again. Only the order of the different animation objects has been changed.

4 Technology and software architecture

In this chapter a software tool that was developed at the centre of product design of the ETH of Zurich is presented.

The requirements for such a tool are as follows:

- easy integration into the value chain
- simple and fast embedding into the software environment of a company
- task- orientated and user- friendly implementation of the interaction.

In order to fulfill these requirements the standard visualisation format VRML (ISO/IEC 14772-1) [8] was used. The geometrical data of CAD systems can easily be generated by an interface and imported into the software tool. Furthermore, it is possible to achieve acceptable frame- rates at middle class object complexity with a standard PC.

The software tool presented hereis based on the VRML program interface EAI (External Authoring Interface) which consists of a library of Java classes that enable the modification of VRML objects during run-time. The communication between EAI and VRML is realised as a interprocess communication between the java applet and the VRML-plug-in of an Internet browser. Therefore, interaction is possible by mouse just as in a CAD program. The latter enables navigation around the object as well as being able to select the chosen objects within the same application window. The detailed information for animation of objects can either be stored in a separate file or saved together with the geometry in a new VRML file.

Figure 3 shows the complete architecture and data management concept of the software. As explained in the description of the concept two different types of users have to be taken into consideration.

On the one hand there is the designer who develops the assembly instructions and on the other hand there is the user, e.g. the assembler, a trainee, a maintainer. The user observes the assembly sequences and acts accordingly. The assembling tool also consists of two different applications: one builder application for the designer and one viewer application for the end-user.

All necessary data of the builder and viewer applications are consistently stored and managed in the PDM system. The builder application adds animation data, textual notes and explanations to the geometric data. The geometric data can again be called and visualised by the viewer application.

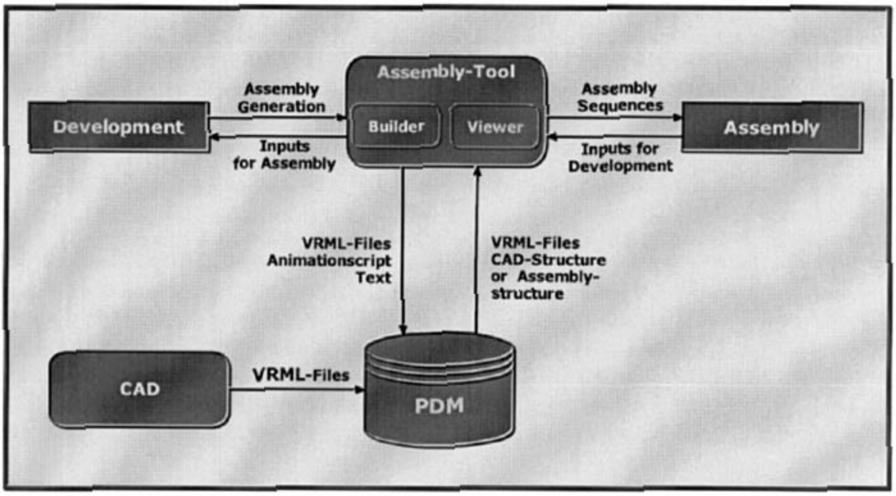


Figure 3: Complete architecture and data management concept

4.1 CAD Interface

Nowadays, most modern CAD Systems are available with a VRML 2.0 interface. Focussing on details, however, these interfaces show different quality standards in their output format. The formats mainly differ in the conversion of CAD part structure into a VRML object hierarchy. Further differences can be found in the identification of objects. Some converters retain the CAD numbering system while others change them for example to consecutive numbers [12].

Generally speaking the CAD part hierarchy normally does not fit the assembly structure. This fact leads to the necessity of restructuring the VRML hierarchy. Therefore it is important to easily identify the different part groups by names and stage within the structure.

4.2 Builder application

The builder application serves to create assembly sequences through animation and text information. Since most design departments use a different product view than the assembly department, it is necessary for the tool to offer facilities in order to group, re- name and restructure the object hierarchy. The objects can either be transferred from the CAD as single parts or assemblies.

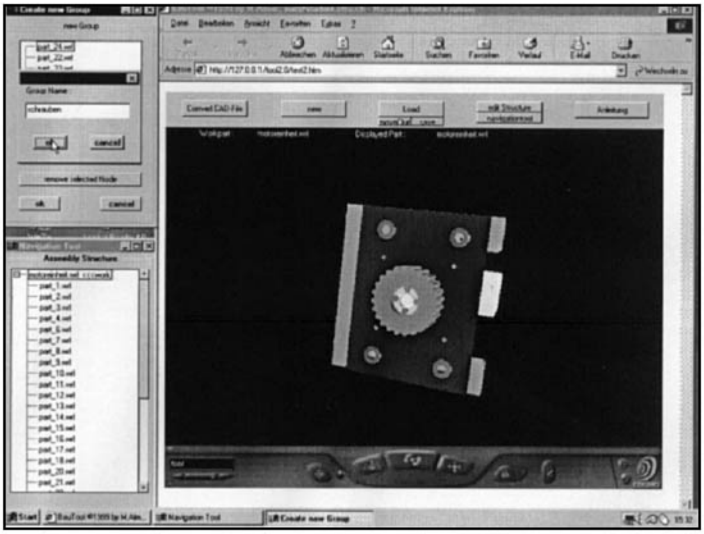


Figure 4: Generation of assembly relevant part groups in the Builder application

The first step consists of the generation of assembly relevant part groups (figure 4) that can be handled together. Since in the CAD system the part groups are mostly represented in assembled condition, they first have to be disassembled with the tool. One assumes that assembly and disassembly sequences show the same contents although in different directions. The movement of disassembly originates from the animation of the assembly instructions in opposite direction.

If additional, non-CAD related objects (e.g. supplier parts) are needed, they can be imported and with the existing part group structure using a mating function.

The creation of animations is based on the definition of a start and end position of the part group or single parts as well as the type of motion (translation, rotation or a combination of both). Every single step can be defined and saved.

Each assembly process can be accompanied by text information and description. This can be displayed by an individual HTML frame other than the plug-ins. The text contains information that can rarely be visualised in geometric forms as for example, used specific hand tools or various details like torque for screw fixing.

4.3 Viewer application

In the viewer application the informational text and the different animation sequences are shown in a two- part- browser at the same time. The user can hereby choose between a step-wise or a continuos run. Critical comments or notes can be added at any time within the appropriate text window. This information is retransfered to the design department in order to improve the assembly. This leads to further improved communications between design and assembly department.

5 Conclusions

The customer imposed reduction for product development time and the increasing requirements on product quality are enforcing companies to restructure their whole product development process and also the assembly process.

The concept presented above (including the software tool) shows that current CAD technology and standard visualisation formats can establish and improve communications between design and assembly. This results in shorter development and realisation working time. Furthermore, the resulting assembling sequences can be used for sales purposes, training and service.

The software tool has been verified by several companies in different departments. Satisfying results have shown the possibility for further development of the software tool for commercial use.

References

[1] Prof. Dr. M. Meier, U. Leonhardt: Customer Driven Design" Ein neuwertiger Technologie-Ansatz zur Konfiguration modulbasierter Investitionsgüter. Projektverbund WZMO Oktober 1998.

[2] G. Burdea, P. Coiffet: Virtual Reality Technology. John Wiley & Sons, ISBN 2-866601-386-7 (1994)

[3] Fan Dai, Virtual Reality for industrial applications; Springer Verlag, Berlin Heidelberg New York, 1998

[4] Simulation, diagnosis and virtual reality applications in the automotive industry including supercomputing applications; Croydon : Automotive Automation, (1997), ISBN 0-947719-88-1

[5] Peli E. (1995); Real vision & virtual reality; Optics & Photonics News 6

[6] Bauer C. Nutzenorientierter Einsatz von Virtual Reality im Unternehmen – Anwendungen, Wirtschaftlichkeit, Anbieter; Computerwoche Verlag München, (1996)

[7] R.A. Earnshaw, J.A. Vince, H.Jones:Virtual Reality Applications. Academic Press, ISBN 0-12-227755-4 (1995)

[8] Hase, H.-L.;Dynamische virtuelle Welten mit VRML 2.0, dpunkt Verlag, Heidelberg 1997.

[9] Heger R.; Entwicklung eines Systems zur interaktiven Gestaltung und Auswertung von manuellen Montagetätigkeiten in der virtuellen Realität, Springer Verlag, Berlin Heidelberg New York, 1998.

[10] Deneb Robotics; USA: factory simulation, telerobotics, software; http://www.deneb.com

[11] Engineering Animation Inc. (EAI); 3D Animation, Software, CAD/CAE, VR; http://www.eai.com

[12] Meier, M.; Bichsel, M.; Elspass, W.; Leonhardt, U.; Wohlgensinger, M.; Zwicker, E.: Neuartige Tools zur effizienten Nutzung der Produktdaten im gesamten Produkt-Lebenszyklus. Konstruktion (1999) Heft 9/99, S. 11-18.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

COMMUNICATION TO ACCELERATE PRODUCT DEVELOPMENT IN A COLLABORATIVE ENGINEERING ENVIRONMENT

K-H Grote and I Kimura

Keywords: Collaborative engineering, product development, rapid and virtual prototyping, Internet-enabled communication

1 Introduction

The market has become extremely competitive because of its globalization and interconnection. In order to take the lead in such a market, it has been widely realized that competitive products which meet customers' lofty demands must be not only introduced in shorter time-to-market but also cost-effective. To satisfy these design goals, many research efforts have been undergone [1, 2, 3, 4]. These efforts have resulted in systematic design methodologies and highly sophisticated computer-aided design tools such as CAD/CAM/CAE, rapid prototyping, virtual prototyping, and so on.

These computer tools are usually developed specifically for design or product development (PD). However, other computer technology disciplines which were originally developed for disparate purposes have also rapidly emerged in engineering design. One of the well known computer technology new to PD is the Internet. It provides enormous amount of information such as products and technologies and enables one to team up with diverse disciplinary individuals located at various places. Teaming up with the people in different locations is increasingly important to develop new products because holding all resources such as expertise, computer tools and human at one place is cost inefficient. Instead, a team with appropriate resources should be assembled through the Internet at the start of a new project.

This interdisciplinary team brings great benefits to companies. For a large company, the team can be down-sized so that the investment and the PD cost can be minimized. In fact, the large companies have teamed up with other companies such as part suppliers, subcontractors and original equipment manufacturers (OEMs). These member companies are involved in earlier PD cycles, and the responsibilities are widely spreading in the entire teams. On the other hand, a small company can also form a team with other small, specialized companies and become a potential consortium against the giants in its market. Therefore, the Internet is becoming a popular design tool and part of a design methodology, so-called *collaborative engineering* (CE).

Even if the advancement of computer technology has a significant impact on PD, design is still a human-centered activity—designers still need to interact with collaborators, data and tools so that communication which is a primitive design activity is also of importance and significance in the new design methodology. Therefore, this paper discusses communication in the early PD stages. Here, the term communication is defined as means to interact between designers, between designers and product data, and between designers and computer applications.

2 Internet-Based Technology

The Internet enables one to communicate synchronously (e.g., videoconferencing) and asynchronously (e.g., scheduling applications, authoring tools, email and file depository servers), and a design environment with it is called Computer-Supported Cooperative Work (CSCW). CSCW differentiates from other computer-aided design tools in cognitive and interpersonal design activities and supports information management activities; i.e., manually creating, organizing, and maintaining information in various media and modes, manipulating information by various tools, and sharing and communicating information [5].

CE project participants can benefit from working with others with the Internet. Knowledge is generally highly personalized and can be shared by communication. Since the information exchanged through the Internet becomes more informal and frequent, the participants can accumulate the necessary knowledge from the others. The informal communication means that whenever she or he encounters a question, a person can immediately contact and inquire an appropriate correspondent or source for that question, while the formal communication is that questions can be brought up only at a right time and place such as a face-to-face meeting so that the inquirer might have to wait for the answers for an extended time.

3 Rapid and Virtual Prototyping as Communication Tools

Although more frequent, informal communication has become common with it, CSCW is largely relying on verbal discourse. Describing designs in words sometimes does not make all people understand because geographically distributed team members have different cultures, languages and thinking. Different disciplinary individuals located at the same place even use different terminology to express the same engineering matters. Therefore, visualization of product ideas plays an important role in communication.

In PD, visualization can be done by sketching, drawing, and prototyping. Sketching and drawing are good and simple means to convey design intentions, but once a product becomes complex, it limits to express a designer's thoughts. Sketching her or his ideas, a designer can clearly explain the ideas; however, if the designer prepares sketches by himself in brainstorming and hands them out to other members, the others may not exactly understand his sketches unless he has artist-like skills to draw figures. Technical drawings have also a problem to communicate design concepts with others. Technical personnel may understand the drawings, but non-technical people are most likely not. Conversely, physical or digital prototypes are much clearer to everyone than the two preceding methods so that the prototypes might be able to be universal communication agents [6].

To respond the ever-increasing pace of PD, the traditional prototyping can not be the best solution to current design practice, but new prototyping methods rapid prototyping (RP) and virtual prototyping (VP) can. The major benefit of RP is that physical prototypes can be generated at the earlier PD stages and in much shorter time than the traditional prototyping methods. As with the advancement of RP technology, commercially available RP machines have become diverse and can be chosen for appropriate design stages. The RP machines specialized for conceptual design are called a concept modeler or 3D printer. With a concept modeler, an RP process is much faster, easier and cheaper and can be done even in an office environment. It sacrifices a model's accuracy and strength for the prototyping speed and cost, but the prototypes are sufficient to evaluate design intentions and to be used as communication means in the early PD stages.

VP, on the other hand, aims at dramatically slashing the number of physical prototypes by modeling and evaluating designs digitally. For example, a product idea is first created by a solid CAD modeler, and then various analyses such as kinematics, dynamics and functionality on this solid model are performed without leaving its computer environment. Also, using the advanced user interfaces such as a head-mounted display, a projection-based display and datagloves, designers can review the designs as if they evaluated the designs in reality.

Visualization by RP and VP can be most appreciated when a design becomes complex as well as when ergonomics and appearances of a design are of concern. Discussing these design issues without either physical or digital prototypes might be simply impossible especially when collaborators are geographically dispersed.

It may sound contradicting that the production of physical prototypes is encouraged with RP but the number of them is reduced by VP. The bottom line of this contradiction is that formal physical prototypes are not necessary in the early stages of PD where decision making is critical and has a significant influence in downstream PD activities. It should be noted that RP and VP have advantages and disadvantages in their own right [3]. Therefore, it is desirable that they are complementarily used in PD.

4 Lessons Learned from a Collaborative Engineering Project

The collaborative engineering project to develop a TV backlight system, a lighting system to improve quality of watching TV, was provided by Phillips Electronics NV. The necessity of such a lighting system came from situations where many people watch TV in dark rooms—the glares from a TV screen strain a viewer's eyes. The project participants were students of three universities: Otto-von-Guericke University Magdeburg, Germany, Fontys University of Professional Education, the Netherlands, and Lehigh University, USA. In this section, how communication was performed in this collaborative engineering project is described.

At the onset of the project, the team had to rely only on a conventional communication method, telephone, to discuss the project and obtain information on available computer systems and software at each location. Once the information became available, the software and computer tools which were used through the project were standardized—email as a primary means to communicate with the collaborators, iVisit for videoconferencing, and the Basic Support for Cooperate Work (BSCW) server as a document depository.

Exchanging emails was the most convenient way to communicate with the others in this project. Whenever questions or problems arose, inquiries by email were simply sent to not only a corresponding person but also all of the team members because all the information taking place among all the collaborators had to be exposed so that all the members were able to comprehend the progress of the project. However, the number of emails that each member received became larger and larger, and most of the mails were irrelevant or confusing to the individuals. The voluminous messages without any priority led the receivers not to answer the emails in a prompt manner. There should have been a protocol to react upon [8]:

- Is the message received, read and understood?
- Are the required actions taken? If so, when? If not, what are alternative actions?

In addition, it would have been very useful if all the messages were automatically archived.

For desk-top videoconferencing, iVisit (see Figure 1) was chosen because unlike other videoconferencing software packages, iVisit allows users to run a multi-party conference without any central server. It also supports both the Windows and Macintosh systems, has a small package size, and requires low system requirements. However, it offers only basic functions such as audio-video conferencing and text chat.

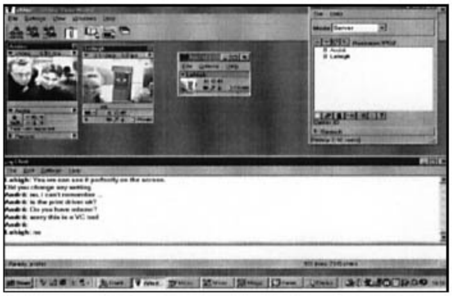


Figure 1 Videoconference session with iVisit which is freely available from www.ivisit.com

During videoconferencing, the quality of sounds was sometimes disturbing. At the beginning of each meeting, it took some time to adjust the system at each site, and when the systems did not come along well, text chat was instead used. The video pictures were also slow. Thus, iVisit limitedly allowed the members to verbally exchange information. Soon after the project started, the difficulty of this verbal exchange was encountered—the collaborators in the US had a hard time understanding a concept of the TV backlight and could not picture the living rooms in Europe where the TV backlight system was marketed. To make sure if all the collaborators correctly understood the discussions and information given in the sessions, conference minutes were prepared at each site and distributed to the other two locations after every videoconference session.

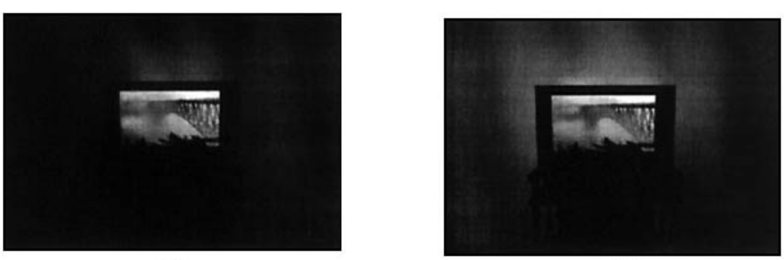
Unlike face-to-face meetings, the conference participants were not able to have eye contacts or read a body language so that the conversations in videoconference became rather awkward. When one participant was speaking, the other participants were not sure to whom the speaker was talking unless she or he directly named a person to speak so that it was difficult to know whose turn it was to speak or when was the right timing to speak. Usually, videoconferencing software does not allow multi-users to speak at the same time so that the users must wait and talk in turn.

BSCW, as shown in Figure 2, was developed by the German National Research Center for Information Technology as a tool for CSCW, and is a Web-based shared workspace where data files can be uploaded and stored and collaborators are allowed to view all the files and to conditionally modify them. This document depository server was used as posting and archiving new and old documents and data files, respectively. Instead of sending all files as email attachments, all the new electric information was conveniently uploaded to the BSCW server. As the project progressed, each data file expectedly grew bigger so that it became cumbersome to attach the file to an email. Meanwhile, all the old information was kept in the server. Archiving old information is important for traceability and knowledge management. The archived files can be also used to produce another new product, and PD time can be eventually reduced by making use of the archives.

Figure 2 BSCW shared workspace (http://bscw.gmd.de)

The stumbling blocks of using the BSCW server were that the member who posted a data file had to attract the other members' attentions to look to the workspace and that the organization of the shared workspace was jumbled. Once a document was delivered on the server, all the collaborators relevant to the new information must have been noticed of the new file available; otherwise, they could not know its availability. Moreover, the workspace became disorganized as the project went on because the amount of information grew larger and larger and anyone in the team was able to access and upload files on the server. Some members had expressed that it was sometimes difficult to find the exact files for which they were looking.

VP was used to simulate light conditions (see Figure 3) and model the design. Without VP, it was difficult, or even impossible, to explain the optimal light condition to the other team members with accountability because there was a numerous number of colors and types of light sources such as bulbs, fluorescence and halogens. The animations were very effective to convince the others of not only the type, position, and number of the light source but also the concept of the TV backlight—it was not difficult to imagine what the TV backlight was when the project was first assigned, but none of the participants had seen such a light. Moreover, the product concepts themselves were visualized with the solid CAD modeler (see Figure 4).



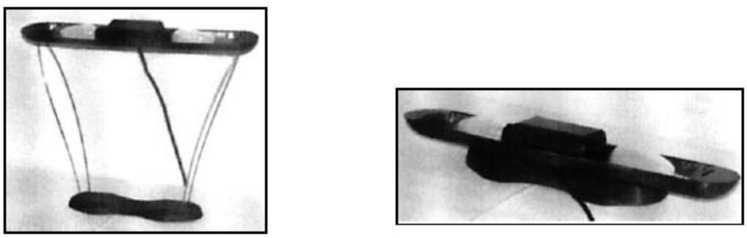
(a) (b)

Figure 3 Computer simulation of a light source behind a TV screen: (a) one bulb and (b) two bulbs

However, it has been said that CAD images are never better than real physical prototypes and do not appeal the viewers' senses of sight and feeling even though they look real. This might be because the 3D images are, after all, depicted only on 2D space so that it is difficult to comprehend design concepts. Therefore, the representation of design concepts by physical prototypes can not be completely eliminated. Figure 5 shows the physical prototype for this project.

(a) (b)

Figure 4 CAD models of the TV backlight: (a) higher position and (b) lower position configurations



(a) (b)

Figure 5 Physical prototypes of the TV backlight: (a) higher position and (b) lower position

5 Product Development

PD with the Internet is becoming a must in current design practice. From the experience, it is found that desktop videoconferencing is not as convenient as it has been said because, in part, the network environments around personal computers are not adequate yet so that the expensive V-span ISDN videoconference might be necessary at the important points such as start-ups, milestones, and final presentations as the project progresses [8]. Videoconferencing in a worldwide collaborative environment can be also hindered by different time, cultures and languages. In such a synchronous communication environment, aggressive, dominant individuals might press others hard to gain acceptance of their opinions; however, the asynchronous tools soften the social aspects of interaction, e.g., emails and file depositories allow any author to express the ideas in words freely. Therefore, email and a file server are realized to be the most useful and beneficial means to communicate with other people.

The shortcoming of the asynchronous tools, on the contrary, is that the responding time is slow. This belated response is critical for reducing PD time. Even if the design process has become significantly efficient by the recent technological advancement, it comes to nothing if the procrastinating habit to answer the received emails is not changed.

In reality, the number of meetings increases as PD becomes faster because most of the decisions are made in the meetings. In some cases, decision makers spend most of their working time in attending the meetings. Thus, reducing the number of meetings and increasing the speed of decision makings are urged. Videoconferencing is just another means of a face-to-face meeting, and possibly it takes more time to discuss issues at each conference than the face-to-face meeting. Even if it reduces the number and cost of traveling, videoconferencing might barely help reduce the number of meetings dramatically nor PD time, but the asynchronous means have great potential to shorten the lead time.

Email in PD is mainly used in the following cases: (1) to propose new ideas and seek decisions and feedback, (2) to inquire information and (3) to notify collaborators of new information on the file server. Figure 6 shows the flow of emails. A sender or a group of senders is the one who sends emails to a corresponding receiver or receivers for one of the three cases. The corresponding receiver is the one who is directly asked and has a responsibility to take actions for the received emails. A relevant receiver or a group of relevant receivers does not have to reply the emails but can observe what is taking place. If necessary, the relevant receiver can send messages to the sender(s) and the corresponding receiver(s). Top management also receives and can observe all communications. Besides this, top management has an exclusive right to decline or reject the proposed or suggested ideas.

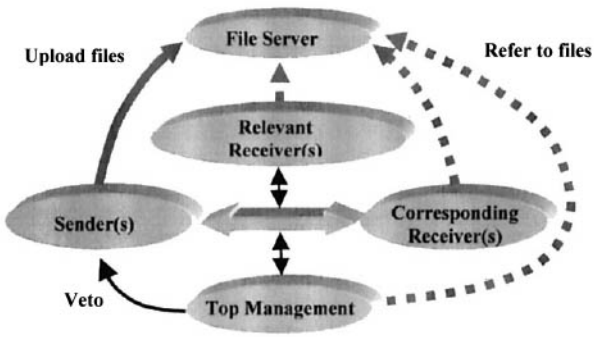


Figure 6 Email flow

If the corresponding receiver receives an email, she or he should first follow the protocol mentioned in section 4. When the email is seeking a decision, it might be urgent to reply the mail because design activities by the sender might be stopped until proper decisions are made so that there should be a rule to handle belated responses. For example, if she or he does not receive any answer from the receiver for more than a certain amount of time, the sender should proceed her or his actions based on her or his own decision. The sender becomes more aware of responsibility for the propositions or suggestions in the email, and that makes the sender carefully scrutinize and research the matters before sending the email.

Additionally, it is necessary to standardized the formats of email for each of the three cases mentioned earlier. The formatted email is easy for the receivers to read and to have the sender's points so that the time to read and reply emails can be reduced.

6 CONCLUSION

It has been realized that communication in PD is still of importance and making use of the computer communication means such as RP, VP and the Internet is essential to reduce PD time and increase product performance.

In this study, collaborative engineering with RP, VP and the Internet were briefly discussed. Then, focusing on the communication issues, the lessons learned from a collaborative engineering project was described. With this experience, the importance of email and a file sever were realized, and how effectively these tools should be used in PD was also argued.

Finally, it should be noted that fully exploiting basic functionality of the Internet should be more weighed before investing in new technologies. In fact, the companies are frustrated by the short lifecycle of software. Even though they invest enormous amount of money in their infrastructure, the software easily becomes an old release, and the old and new releases might not be compatible. Additionally, the collaborative team might last only for a project or two. Buying commercial software every time might not be so reasonable. Therefore, the usage of email and a file server should be carefully reconsidered.

REFERENCES

[1] Grote, K.-H. and Miller, J., "Integrated Product Development with State of the Art Engineering Tools", Proc. International Conference on Computer Integrated Manufacturing, Zakopane, May 14-17 1996.

[2] Pahl, G. and Beitz, W., Engineering Design: a Systematic Approach, Translated by Wallace, K., Blessing, L. and Bauert, F., Springer-Verlag, London, 1996.

[3] Ulrich, K.T. and Eppinger, S.D., Product Design and Development, McGraw-Hill, New York, 1995.

[4] Voland, G., Engineering by Design, Addison-Wesley, Massachusetts, 1999.

[5] Reich, Y., Konda, S. and Subrahmanian, E.,. "Information Infrastructure for Collaborative Product Development", Available at http://or.eng.tau.ac.il:7777/course/cim97/iicpd.html, 1997

[6] Grote, K.-H., Birke, C. and Kimura, I., "Rapid Prototyping in the Early Design Cycle", Proc. 3rd International Workshop for IPD, Magdeburg, pp.80-90, 2000.

[7] Grote, K.-H., Torby, B. and Kimura, I., "Rapid Product Development", Proc. ASME DETC, Baltimore, Maryland, Sep. 10-13, 2000, DAC-14491.

[8] Van Kollenburg, P.A.M., van Schenk Brill, D., Schouten, G., Mulders, P., Ochs, J.B., Zirkel, M., Kimura, I. and Klette, G., "Collaborative Engineering Experiences", Proc. E&PDE, Brighton, UK, Sep. 6-7, 2000.

Corresponding author

Mr. Ikuo Kimura
Otto-von-Guericke University, Magdeburg, Mechanical Engineering/Engineering Design (IMK/KT), Universitaetsplatz 2, Magdeburg, D-39106, Germany, Tel: +49 (0)391 67 11 666, Fax: +49 (0)391 67 12 595, E-mail: kimura@mb.uni.magdeburg.de, URL: http://imk.uni-magdeburg.de/

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

VISUALIZING THE IMPACT OF TOLERANCES ON COSMETIC PRODUCT QUALITY

N P Juster, M R Fitchie, S Taylor, P Dew, J Maxfield, and J Zhao

Keywords: Tolerance modelling, variational modelling, virtual reality

1 Introduction

This paper describes a previously unreported application of virtual environments – the prediction of product cosmetic quality. Successful prediction of cosmetic quality without the production of a physical prototype requires product variation to be modelled and presented to the user. Current virtual models only allow presentation of nominal, or perfect, geometry. The paper presents the underlying *tolerance model* and its application to an automotive case study.

2 Cosmetic Quality

Many companies, particularly in the automotive and aerospace sectors have begun to use virtual prototypes (also known as digital mockups or electronic builds) to remove the need for some or all of the physical prototypes traditionally constructed during the product development process. There are many reports of significant business benefits obtained through the use of virtual prototypes within the design and manufacturing process. Many aspects of this process have been considered from initial clay modelling and concept design through to detailed assembly and manufacturing planning [1-5]. An area yet to be reported is that of predicting the *cosmetic quality* of a product. Cosmetic quality has no precise definition. It is a customer perceived product attribute. It may be loosely defined as the 'look' of the product. Features such as the size and shape of gaps and the flushness between mating components are areas that need to be controlled in order to maintain the cosmetic quality of a product. Cosmetic quality can have a dramatic impact on the aesthetics, style and market differentiation of the final product and the ability to predict it early in the design process should enable companies to reduce the number of late design changes and costly rework.

In order to predict cosmetic quality, virtual prototypes must be able to show how the cosmetic quality changes (i.e. gaps and flushes between components) as each component varies within its allowable tolerance (or manufacturing variation). This is because products that may be acceptable to the customer when produced from perfect components may be totally unacceptable when assembled from components towards the limit of their allowable manufacturing variation. One of the most significant trade-offs in automotive design is that of tolerance assignment (with a direct link to production cost) against cosmetic quality. Providing both accurate dimensional and aesthetic analysis in the same format would enable the *visualisation* of the impact of tolerance assignments. This would assist engineers to achieve a suitable compromise between production cost and cosmetic quality. Unfortunately, current commercially available tolerance analysis systems produce data in the form of histograms that are not easy for a non-expert to interpret and CAD systems usually only display nominal geometry.

This paper describes the interim results of a collaborative research project, known as VITAL (Visualization of the Impact of Tolerance Allocation in Automotive Design), that is researching and developing prototype software to enable designers to easily assess cosmetic quality as components and assemblies deviate from nominal.

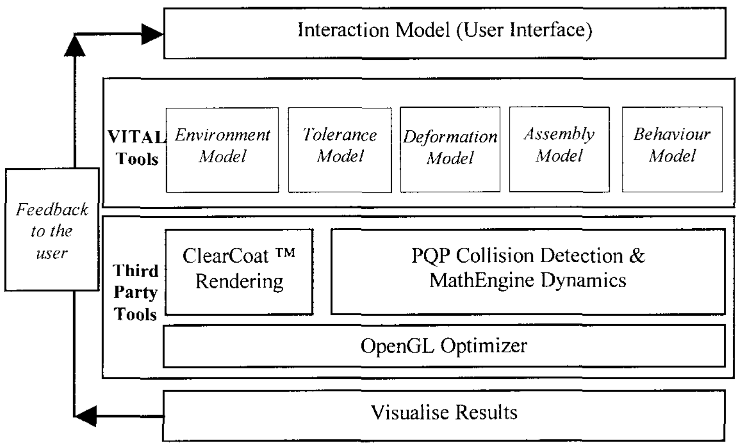


Figure 1: VITAL System Architecture

The VITAL software architecture is shown in Figure 1. The system integrates several simulation models to improve the accuracy and realism of the simulated environment. The aim is to produce a photo realistic, interactive 3D view of the product that 'responds' to external forces and can display geometry that varies from the perfect (or nominal) due to manufacturing and assembly variation The *interaction model* provides the interactive user interface to the system and enables the user to steer the simulation within the environment. The *environment model* simulates a photo-realistic environment, in which the product resides and generates a realistic lighting model for the product. The *tolerance model* gathers and filters the results on an off-line tolerance analysis to simulate the effects of different tolerance scenarios, such as most likely, worst-case, and best-case conditions, and enable "what-if" studies. The *deformation model* simulates the effects of component deformation as a result of different tolerance conditions, by deforming the geometric representation of the components. The *assembly model* represents and maintains assembly constraints between components and supports the interactive assembly and disassembly of the product within the environment. Finally the *behaviour model* simulates the physical dynamics of the product within the environment, maintaining consistency against physical laws such as gravity, friction and contact dynamics. The strategy of decomposing the system into assembly, tolerance, behaviour, environment and interaction models to support cosmetic quality inspection, can be seen as a particular instance of an Immersive CAD environment as defined by the National Institute of Standards and Technology (NIST) and Washington State University [3]. The

VITAL system is described more fully in [6]. This paper will concentrate upon the tolerance model.

3 Tolerance Model

In recent years a number of software tools, such as CE/Tol 3D+ from Raytheon, VSA-3D from Variational Systems Analysis and 3-DCS from Dimensional Control Systems have become available that enable potential tolerancing problems to be identified and their impact on the total design assessed. These tools cover several aspects of tolerancing within the design process and most offer Geometric Dimensioning and Tolerancing ‘Advisors’ to assist in the interpretation and understanding of dimensional data. Advisors can check and associate symbol correctness as well as identify critical measurements associated with the key product characteristics. It identifies those parts that may require additional tolerancing, have multiple tolerances that conflict with each other or are redundant.

Tolerance analysis tools allow the dimensional variation in an assembly to be predicted. The tools provide true statistical simulation modelling, with a graphic display of the variation and view of the assembly sequence defined as a tree structure and diagram or animation. Most tools can be integrated with modern CAD packages such as I-DEAS and CATIA. The tools are feature based, using multiple windows and models and no knowledge of programming languages is required for operation although an open architecture allows custom routines to be coded.

The tools automatically generate mathematical relationships to vary the form, orientation, location and size of geometric features. Monte Carlo simulation of a production run is executed in which feature dimensions are randomly varied based on the assigned tolerance, the allocated process capability (C_p and C_{pk}), and knowledge of the assembly sequence. Results can be obtained that include the standard deviation of the tolerance under investigation, percent of products that will be produced out of specification and the minimum and maximum values of the tolerance. The tools also allow "what if" scenarios to be created and analysed. However skill is needed to interpret the output of the tolerance analysis tools and it is difficult to visualise what effect the values obtained from histograms and distribution curves have on the product’s cosmetic quality. In VITAL the output of a tolerance analysis package is used to automatically vary the position and size of the bounding surfaces of the virtual prototype being analysed. This is achieved as shown in Figure 2.

Data from the native CAD system (in this case CATIA), including the position of measurement points for analysis are translated into the B-REP format used by the VITAL Software for visualisation. The VITAL system uses the fast surface tessellators within OpenGL Optimizer to produce and manage polygonal representations at several levels of detail (to be switch depending of view distance). This is a common rendering optimisation technique used in many visualisation and CAD systems. The CAD data, and associated dimensions, tolerances and measurement points are also imported into VSA. Using Monte-Carlo simulation 5000 simulations are run on the model to generate a dataset containing 5000 values for each measurement point. The data is then filtered and used to generate a tolerance control box(es) in the VITAL software. This then enables a two-way interaction between the control box and the VITAL assembly model.

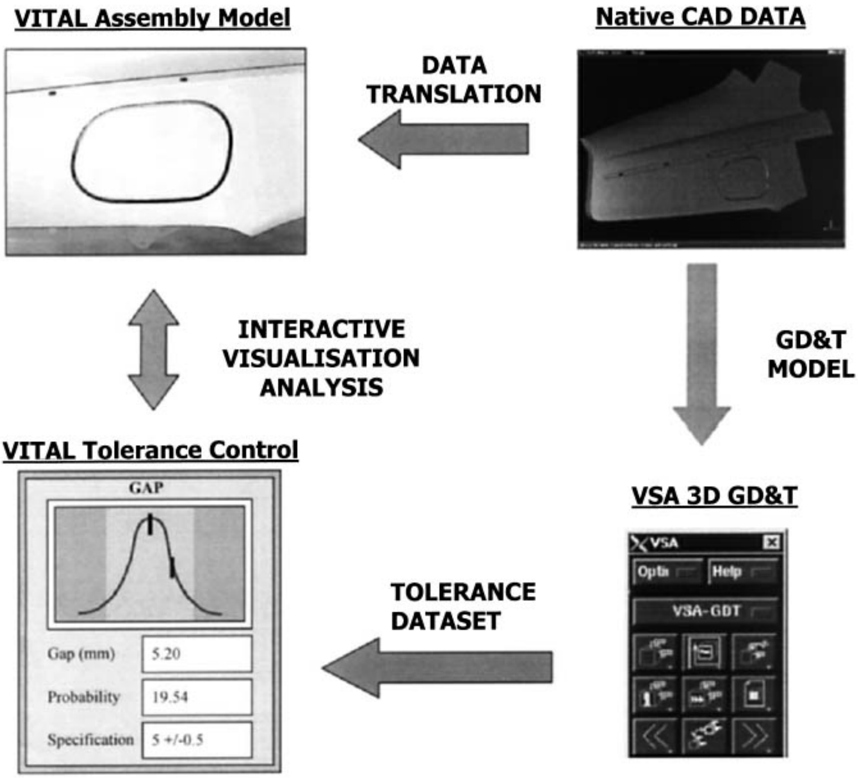


Figure 2: Tolerance Data Flows in VITAL Software

The tolerance control box is shown in more detail in Figure 3. The control box contains two main areas: a graphical representation of the probability distribution for the gap under investigation and a set of numerical values. The graphical section of the box shows: an x-axis representing gap sizes, a probability distribution curve derived from the VSA dataset and four different coloured bands. The outer bands, marked 'U' and shown in dark red on the screen, correspond to sizes of gap that are physically impossible since parts would need to interfere for the gap to be this size. The bands marked 'O', shown in light red on the screen, correspond to sizes of gap that are out of specification but physically achievable. The area marked 'I', shown in green on the screen, are gaps that are in tolerance and the final band, which is unmarked, is a 'slider' that will move as the designer investigates the variation in the assembly. The slider shows the current position of the gap as shown by the assembly model.

The textual part of the box shows the tolerance specification for the gap (here it is 2.50 to 5.00 mm) and the probability of the gap being within the 'gap bin' (here 3.95 –4.05 mm) indicated by the slider. The user can alter the resolution, or size, of the bin.

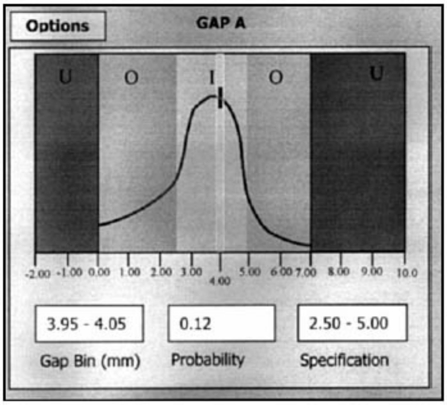


Figure 3: Tolerance Control Box

The user can mover the slider in the tolerance control box to alter the position and size of components in the assembly. Additionally interactively moving parts in the assembly will alter the details in the tolerance control box. This combination gives designers access to a powerful design tool.

4 CASE STUDY

The VITAL system has been applied to the evaluation of the cosmetic quality of the fuel filler flap assembly from the Rover 75. The body in white of the Rover 75 was modelled by BMW Group using CATIA. The model used in the case study consists of the rear wing, the fuel filler pocket, a hinge bracket and pin, and the flap itself. The reason that this model was chosen is that it represents a highly visible "A" class surface and thus any cosmetic problems, such as the gap and flush between the flap and the panel are immediately obvious. This can have a detrimental effect on a potential customer's first impression of the overall build quality of the entire vehicle. Cosmetic problems might occur as a result of component variation and assembly misalignments during manufacture.

Figure 4 shows a view of the filler flap with two gaps of interest. Both gaps are at near nominal position. Figure 5 shows two other scenarios. In Figure 5a the designer has dragged the slider to move Gap A out of tolerance. The size of Gap A was then locked and Gap B moved to near nominal. In Figure 5b the designer has moved the Gaps to sizes that are a physical impossibility.

The tool can be used in two ways. Firstly before tolerances are set the designer can move the assembly to determine gaps and flushes that are cosmetically acceptable. Secondly, when the tolerances have been assigned the tool can be used to investigate the probability of an assembly being in a certain position.

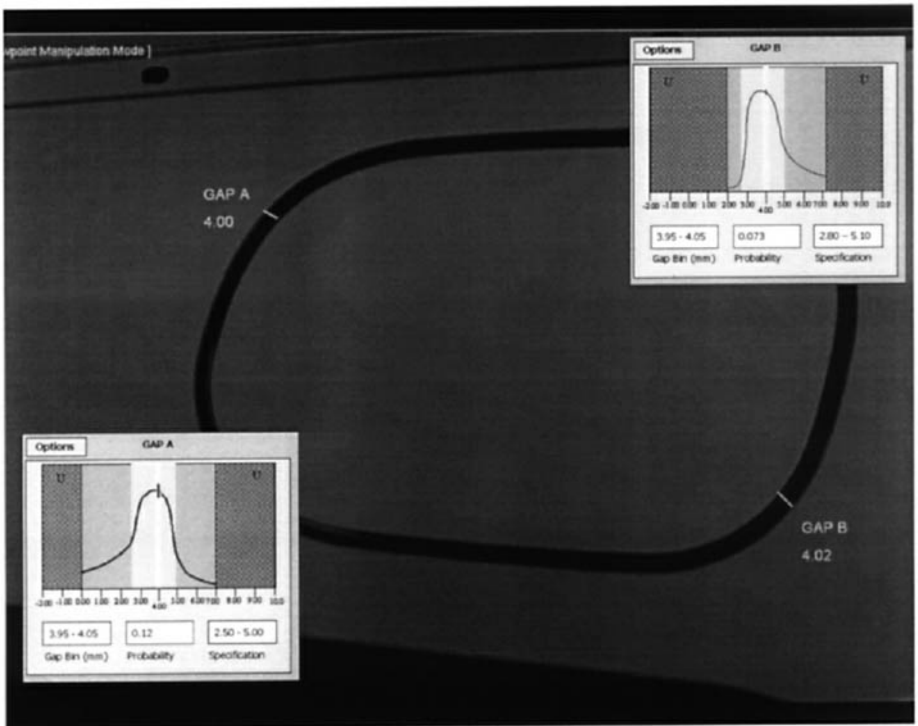


Figure 4: Fuel filler flap in near nominal position

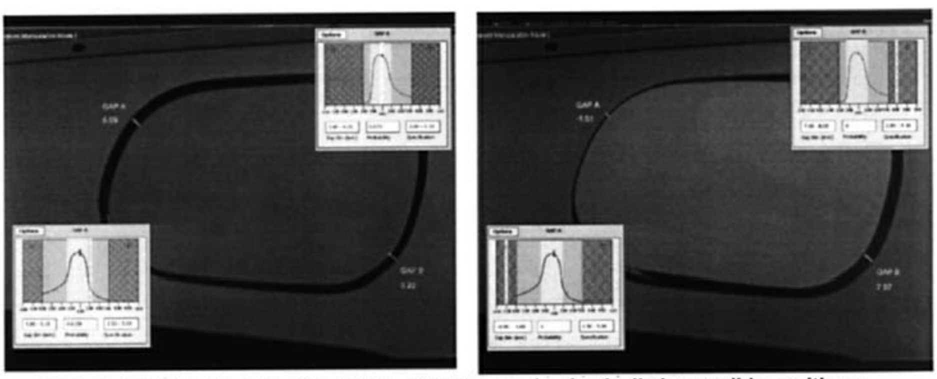


Figure 5: a) One gap out of tolerance. b) Both gaps in physically impossible positions

5 Conclusion

This paper reports on the progress made by the end of the second year of a three year funded project. The output of the project has already given the design engineers at the collaborating companies confidence that virtual environments can have a significant positive impact on design and manufacturing business processes. The main positive outcome of the evaluation is that engineers are convinced that a commercially robust system similar to the VITAL prototype would greatly assist with the setting of quality targets. In particular, by providing visual representations of flush and gap limits, previously only listed as numbers on paper, within realistic looking environments, such a tool would help to remove some of the subjectivity associated with setting quality targets. Discussions on acceptable quality targets could take place with the design and manufacturing team and potential customers.

A common concern expressed by the engineers was the number of business processes, and software tools, that will have to be integrated within the concept design stage in order to make proper use of the VITAL software. The current problems associated with data exchange issues and the time taken to set up scenes (light types and positions and environment maps) combine to reduce efficiency. Unless engineers can quickly move design data into the VITAL environment, conduct 'what if' analyses, and return the results to the design environment within a reasonable time scale, the software will not be used. The definition of the term 'reasonable' is still a subject of debate. Engineers would prefer that a single design iteration could be done in a matter of minutes – current data exchange and scene definition issues mean that currently this cannot be achieved in less than a day.

The VITAL software has, to date, only considered assemblies that consist of rigid geometry. As the project progresses it will tackle more challenging case studies and require the development of more sophisticated simulation techniques to enable flexible assemblies to be considered. These components contain a significant proportion of the surfaces visible to the customer in an automobile.

Acknowledgments

The research has been conducted with the support of Grants GR/M17099 and GR/M12094 from the UK Engineering and Physical Sciences Research Council. The research is collaboration between the Universities of Leeds and Strathclyde, Magna Interior Systems and SGI. Data was supplied by the BMW Group. Steve Watson (stephenw@reading.sgi.com) at SGI contributed to the solution of data exchange issues and provided advice on developments in visualisation technology.

References

[1] Baylis G. M., Bowyer A, Taylor R. I., and Willis P. J., Virtual Manufacturing, *Proceedings of CSG '94*, Winchester, UK, April 1994, Session: Set-theoretic Solid Modelling: Techniques and Applications, pages 353-365.

[2] Beier, K. P., Virtual Reality in Automotive Design and Manufacturing, in *Proceedings of Convergence '94*, International Congress on Transportation Electronics, SAE (Society of Automotive Engineers), Dearborn, Michigan, October 1994.

[3] Jayaram S., Wang Y., Jayaram U., Lyons K., and Hart P., A Virtual Assembly Design Environment. In *Proceedings of IEEE Virtual Reality 99*, Houston, Texas, USA, March, 1999.

[4] Thompson M., Maxfield J. and Dew P.M. Interactive Virtual Prototyping. In *Proceedings of Eurographics UK '98*, pages 107-120, March 1998.

[5] Wilson J. R., Brown D. J., Cobb S. V., D'Cruz M. M., and Eastgate R. M., Manufacturing Operations in Virtual Environments (MOVE), *Presence*, Vol 4, No. 3, Summer 1995, pages 306-317.

[6] Maxfield J, Juster N. P, Dew P. M, Taylor S, Fitchie M, Ion W. J, Zhao J, Thompson M, Predicting Product Cosmetic Quality Using Virtual Environments. In *Proceedings of 2000 ASME Computers and Information in Engineering Conference*, September 2000, Baltimore, Maryland, USA Paper Number CIE-14591

Corresponding Author: Professor Neal Juster
Department of Design, Manufacture and Engineering Management, University of Strathclyde, Glasgow G1 1XJ, UK, Tel: (+44) 141 548 2091, Fax: (+44) 141 552 7896, E-mail: n.p.juster@dmem.strath.ac.uk, URL: www.dmem.strath.ac.uk, www.vsp.co.uk/tenants/vital/

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

INDUSTRIAL DESIGN ENGINEERING – FACILITATING COMMUNICATION OF PRODUCT PROPERTIES USING WEB-BASED VIRTUAL REALITY

P Schachinger

Keywords: Industrial design engineering, design representations, virtual reality, product modelling

1 Introduction

Communicating life-cycle-criteria, "soft values" such as semantic properties and so on between engineering and industrial designers in their eye-to-eye meetings is not an easy task. This is due partly to most engineers' inability to sketch, partly because physical models tend to direct the thinking into solutions (internal properties) and not the desired behaviour (external properties). This paper introduces a way to show intended properties together with geometry using existing computer technology, and proposes how it could be handled together with additional information in the work context specified above, during concept and structure synthesis activities.

2 More details/Background

2.1 The use of representations in a real ID-ED context

To correctly understand the daily design work, the researcher attended some of the weekly design meetings held at larger car company during a half-year period. During these meetings, engineering designers and industrial designers meet and discuss critical issues. Meetings visited have been on two levels, one for the whole interior (more informative) and one for the instrument panel (more operative). Manufacturing possibilities and cost were discussed mostly, leaving out product semantics and ergonomics. Communication during meetings were overhead- and paper-based, using printed drawings of sections and cad-renderings. This led to the typical problems of representations not being up-to-date, information for a satisfactory decision missing and information missing due to people's absence. Also, printed-out sections were often misinterpreted and hard to understand, making some meeting participants incapable to discuss the problem. This worked better using the physical mock-ups as discussion objects, though these were several weeks old making the geometry out-of-date. The large section drawings in true product scale seemed to work as a product representation to gather the meeting round, and problems and changes were highlighted using colored markers. Only a few computerized product representations, such as the meeting agenda, were presented and edited directly on a back-projected screen, even though such equipment was available. Similar experience is shown by Perry and Sanderson [1], reporting for example that "Drawings littered the workplace...". An interesting fact is that within the premises, the company has modern vr equipment such as a power wall and digital mock-up rooms, and all kinds of software for vr. Still, none of this equipment was used.

These observations indicate that it might be fruitful to show upon easy ways of using computers in communication, keeping the complexity low.

More commonly, regarding mechanical product development, Rezayat has proposed two issues for managing the ever increasing information amount [2]: First, "Capturing the information in the form of knowledge by putting the data in a specific context with attached rules...". A product model is contributed in this article for this purpose, emphasizing system-level design. Second, "Creating the means to update, manage, and reuse this knowledge throughout the extended enterprise.". The product model is represented in a management tool, aiming at showing the right knowledge to the design team.

2.2 Other vr and web contributions

This section displays some contributions having more or less similarities with this work, i.e. web and vr tools and methods supporting work of an industrial or engineering design team, regarding synthesis and evaluation of concepts and forms.

Table 1 Some earlier work regarding web and geometry.

Dahan et al. [3]	Concept evaluation by a group of potential users, grading of criteria and evaluation (pricing)	Overview of all concepts, text and animation describing use,
Rezayat [2]	searching existing good system solutions, digital engineering notebook,	A database of old solutions, VRML models illustrating solid model, notebook
Charlton & Wallace [4]	A broker for finding components suiting a function.	Clicked VRML-object generates part alternatives
Szykman [5]	Design repository, support for design, finding existing knowledge generated from previous design projects	
Hietikko [6]	Tool to improve communication in a smaller team	Common database. pdf, html, VRML documents
Nidamarthi [8]	An internet workbench and a VRML viewer. Aspects of communication.	Common database. VRML assembling several models.
Abramovici et al. [7]	Visualization of product structures, combining CAD and PDM structure.	VRML linking,

Dahan makes an important contribution, bridging market surveying with daily design work. Rezayat, Charlton and Szykman contribute with rule-based product structuring systems. Abramovici et al. explore ways of visualizing relations between parts. Hietikko and Nidamarthi focus how their tools are used by the team. None of the contributors had tried using the CAD tools for modelling design knowledge. It is often argued that they restrict the designer to think in solutions instead of demands [6], and on components instead of the product as a whole. Moreover, this paper argues that it is not the CAD tool used that restricts, it is the way it is used. The research project aims at pushing the possibilities of using vr techniques in daily design work. This implies that there could be other easier ways to use the computer for communicating design rationale, for example two-dimensional sketches (form and text) with attached links to documents. Generating and evaluating such alternatives is beyond the frames of this work though.

3 Approach

3.1 Product model

Product models based on the Theory of Technical Systems and the function-means-tree help to fill some logic gaps between intended behavior on one hand and geometric part models on the other, and therefore they constitute a suitable modelling base for this design work. Efforts have been made by the author to bring these models into a computer environment [9], not with a geometric context though.

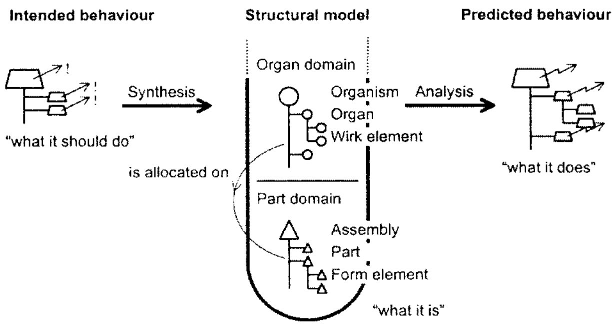


Figure 1 Jensen´s structural mapping [10] used in the product model.

As presented by Jensen [10], *intended behavior* (product specification, problem descriptions, user scenarios etc.) has a logical mapping to the *structural model* through the intended functions relating to so called organs (function carriers, German: Funktionsträgern). *Predicted behavior* is the result from tests, calculations, benchmarks etc and then mapped to the structure once again through the organ–"predicted function" relation. An organ is either allocated over several parts or within a part, then called wirk elements to make a distinction. Parts and form elements (form features) show how the product is constructed, with the distinction that form elements are not separable parts (a rugged surface, a flange, a set of holes for example). Functions are in this article not only technical, but also user-related (semantic, ergonomic, aesthetic) as proposed by Warell [11].

Figure 2 Examples of semantic and ergonomic functions.

Information based on figure 1 is modeled together with design issues of the team. This model is the base for all information gathered during the design activities. Some further explanation to figure 3 is given below:

Issue	Team tasks and problems, conflicts to be solved. An issue could also directly be a function to be solved.
is influenced by	A description of how issues are related
Function	Describes in what behavioral context the function shall be fulfilled and specifies properties related to that function.
is solved by	Description of test results: how the function is fulfilled, unexpected behavior...

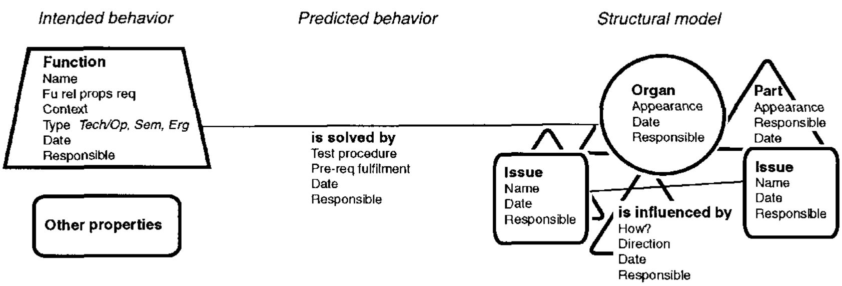


Figure 3 Gathering of information related to the product model

3.2 The three-dimensional interface

Ideally, communication would be held directly in the surface modeller, making extra file export (and thereby iteration loops) unnecessary. The drawbacks with cad systems in common though are the still poor real-time rendering capabilities and the limitations of file distribution. Also, vr formats usually give the opportunity to add extra functionality. In this application, a surface modeller is used as much as possible, complemented by a vr model. The aim is to make the designer do as much manipulation directly on the original model in the surface tool as possible, keeping the number of versions down. The computer environment consists of one surface modelling tool (Alias AutoStudio [13]), one vr modelling tool (Cycore Cult 3D Designer [12]), one web-editor (Netscape Composer) and other text editors (MS Word, MS Powerpoint). The geometry is created using the normal surface modelling procedure, grouping and naming surfaces and curves, leading to a hierarchic tree structure beside the geometry.

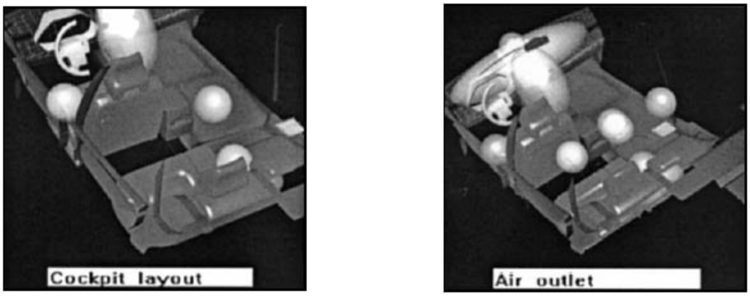


Figure 4 In the left picture, the observer has made the design issues for a car interior visible. The right picture shows overlaid organ and issue structures. A left-click on each object displays the name of the object, and a right-click opens the attached document.

In this approach, structures of organs and issues are added, also in the three-dimensional space, overlaid the part structure. This "over-laying" is believed to be very useful and effective, since the organ-component mapping is illustrated directly and spatial conflicts between organs are made visible. Issues to be treated by the team are placed with the geometry in positions were they are felt to be most relevant. The meeting agenda will thereby be focused around the product. The visibility of the organ structures is toggled to avoid the model becoming too muddled. Also, form elements are only highlighted if they belong to an issue which is being left-clicked with the mouse. More details regarding an objects are shown in separate vr models included in text files (see section 4.1). A small part of the issues and organs can not be placed three-dimensionally, and are reserved a special space. Geometry, materials, animations and the tree structure are therefore exported directly into the vr tool and kept untouched. The figure above shows the vr model, made in the Cycore Cult3d software

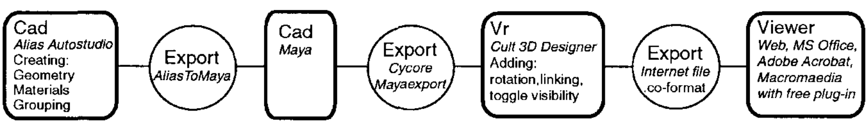


Figure 5 The somewhat lengthy export-procedure.

Unfortunately, there is no Cult export from Alias, so one has to go through Maya [13], without any quality loss though. Possibilities to toggle visibility, show object names and link to files are then added in the vr program. The vr object and its functionality are then saved in a project, leaving the possibility to update the object without changing functionality. For this to work, the top of the tree structure and the linked file names must remain untouched. This automation saves loop time and enables several people to update the vr file. After adding functionality, the vr model is transferred into a compressed format which can be imported to several document types (web, MS Office, Acrobat Reader).

The mayor drawback with the Cult 3D format compared to VRML (used by Rezayat, Szykman and more) is that you can only create the files from Wavefront Maya and Autodesk 3D Studio MAX. The easy-to-use vr modelling program and the good real-time performance on low performance workstations speak for its advantage though.

High-end programs such as Opus [14] and dvise [15] give opportunities for cybergloves, cubes and other peripherals giving higher degree of immersion, where the low-ends above can be used together with back-projected screens at the most. The high-end programs are not suited for the accessibility which this work demand. Focus is therefore set to desk-top monitors and projected screens with mouse and keyboard interaction. Animations in communication (used by Dahan, not here) usually give more accurate material, geometry and light than vr.

3.3 The overall interface structure

The interface is web-based with all project information gathered around the team home page and virtual reality models. The aim has been that everyone in a team should be able to read all files without having to put down large effort or use expensive software. Each person is responsible for updating their own files. All need a web-browser with the free viewer and MS Office. The persons creating CAD models do their own continuous conversion to vr and need the software for doing so.

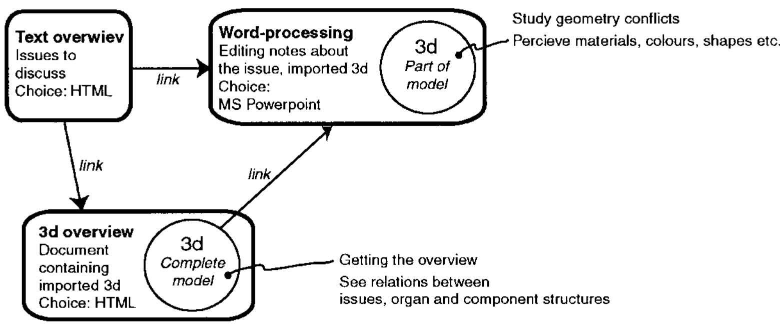
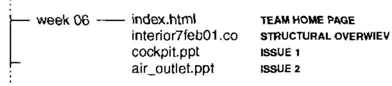


Figure 6 Relations between files used in the team communication.

The "3d complete model" and the "Text overview" give the overview and spatial relations between different structures. The "3d part of model" contains higher accuracy in geometry and material having the same real-time performance, since it shows less geometry. A major drawback is that different types of light can not be accomplished, which leaves out light setting from real situations (use during daytime and night-time, first impression from real situation versus exhibition hall). Animations are here an alternative, but are not tried out in this project. Rendered images are used though within the word-processor.

4 Implementation

Implementation is made at the beginning of an on-going pre-development project, which aims at improving convenience in cars. The task is split up in different areas in the interior, for example *new cock-pit layout* and *new type of air distribution*, based on ideas that have come up during the daily work of the participants. On an introductory stage, contacts are held per phone and e-mail mainly between the team leader and others. The team includes about ten persons, from which one is an industrial designer and the rest are engineering designers mainly. The scenario below highlights typi



Its Monday in week 6. The topic "new cock-pit layout" is stored as an issue-document and placed by the industrial designer in the vr-model on an old interior geometry common for the team (see Figure 4). The issue-responsible engineer and industrial designer communicate ideas adding scanned sketches to cockpit.ppt with additional text. Both chosen and some rejected routes to follow are documented. The industrial designer is in this case the geometry modeler. The organs "distribute air", "generate air flow", "generate music" and so on, are placed in with the geometry as symbols or as forms showing the occupied space. The "generate air flow" organ has in this case a carry-over concept, which occupies space. Supplier, performance and other properties are further described in the attached file. The designers wish to change the position of the air flow organ and exports geometry to the text document, showing alternative positions. The day after, a weekly team information meeting is held. Discussion is now held regarding the position of the air flow organ with the group who designs the new air distribution system. The vr alternatives are here hopefully easy to understand for others in the team, who can give valuable comments. After the meeting, a "week 07" folder is created and files copied. Organs and geometry are gradually replaced with more detailed organs and geometry in progress.

Investigations now focus on following questions:

How much time do one need to spend on a change (in component geometry, in adding issues, adding functions, links etc.)? Can the change be made "on the fly" or do one need to leave the room? How much, by whom and for what occasion is the tool used when researcher is not present? Is use declining or increasing? What type of decisions regarding especially materials and tolerance can be based on an CAD-rendering versus a vr-simulation?

5 Conclusions (and discussion)

This paper focuses on two major issues. First, a different way in which designers can use cad-tools is presented, to increase the understanding of the intended design. Secondly, the work shows that virtual reality techniques are mature for daily, informal design work, and when using a suitable constellation, easy enough to use without any extra knowledge or equipment. The implementation is on-going and has so far shown an eagerness among the team members for using the vr simulation. Further product modelling will also consider hierarchic function-means structures (china-box) and breakdown of criteria.

References

[1] Perry, M. and Sanderson, D. Co-ordinating joint design work: the role of communication and artefacts, *Design Studies*, no 19, pp 273-288, 1998

[2] Rezayat, M. Knowledge-based product development using XML and KCs, *Computer Aided Design*, no 32, pp. 299-309, 2000

[3] Dahan, E and Srinivasan, V. The predictive power of internet-based product concept testing using visual depiction and animation, *Journal of Product Innovation Management*, no 17, pp. 99-109, 2000

[4] Charlton, C. T. and Wallace, K. M. A web broker for component retrieval in mechanical engineering, *Design Studies*, no 21, pp.167-186, 2000

[5] Szykman, S et al. A web-based system for design artifact modeling, *Design Studies*, no 21, pp. 145-165, 2000

[6] Hietikko, E. and Rajaniemi, E. Visualized data-tool to improve communication in distributed product development projects, *Journal of Engineering Design*, Vol.11, Issue 1, p95, 7p, 2000

[7] Abramovici, M., Leszinski, C. New visualization techniques for product structures in product development, ICED' 99, vol 3, pp. 1841-1844, 1999

[8] Nidamarthi, S. Observations from multidisciplinary, internet-based collaboration a practical design project, ICED' 99, vol 2, pp. 709-714, 1999

[9] Schachinger, P. and Johannesson, H. L. Computer modelling of design specifications, *Journal of Engineering Design*, vol. 11, Issue 4, 2000

[10] Jensen, T. *Functional modelling in a design support system – Contribution to a designer's workbench*, Doctoral thesis, Department of Control and Engineering Design, Technical University of Denmark, IKS publication 99.44 A, ISBN 87-90130-20-0, Lyngby, 1999

[11] Warell, A. Introducing a use-perspective in product design theory and methodology, *Proceedings of the 1999 ASME DETC*, Las Vegas, 1999

[12] www.cult3d.com (Accession date 2001-02-08)

[13] www.aliaswavefront.com (Accession date 2001-02-08)

[14] www.opticore.com (Accession date 2001-02-08)

[15] www.ptc.com (Accession date 2001-02-08)

Peter Schachinger, Lic. tech. in Machine Design, BFA in Industrial Design, Chalmers University of Technology, Teknisk design, SE-412 96 Göteborg, SWEDEN, Telephone: +46-31-772 13 74, Fax: +46-31-772 58 58, E-mail: pesc@mvd.chalmers.se

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

CHOOSING THE BEST VISUALIZATION TOOL IN ENGINEERING DESIGN – COMPARING HIGH-IMMERSIVE VR WITH DESKTOP-VR

P Johansson

Keywords: virtual reality, visual simulation, product development, digital mock-up

1 Introduction

When speaking about *virtual reality* to design engineers, visions of head-mounted displays (HMDs), CAVEs or the use of strange glasses enabling to view stereoscopic images in 3D come to their mind. Enhanced visualisation tools, as for example VR, are becoming frequent in product development in industry and a lot of research effort is put into the development of the technology and knowledge around it.

According to a recent market study, Visual Simulation/Virtual Reality is valued at $24 billion in 2000, and is expected to grow at a rate of more than 50 percent annually [2]. The report estimates that the top VR business applications will include virtual prototyping, design evaluation and architecture. However, important engineering design questions about *benefits* and *usability* are too often put aside [1]. The most important research made within the field does not cover engineering design tasks [2].

In industry, digital mock-ups often replace traditional prototyping in the never-ending struggle for the reduction of cost and lead-times and the improvement of product quality and customer value [5]. Product representations have been found valuable for finding technical malfunctions [9]. In these kinds of tasks - do we always need to strive for full immersion or has a lower degree of immersion the same (or even higher) benefit for the engineers? VR-systems are expensive to buy and use - can we obtain good enough results with desktop-VR, i.e. systems that are less expensive? To answer these questions an experimental study has been set up, enabling a comparison of three different visualisation techniques.

The overall aim with this research is to provide a base for the development of guidelines focusing on the use of virtual reality in the early phases of product development. It might appear that this is about CAD Manipulation but CAD is only a mean to achieve the results, from which the guidelines will be developed. The specific aim of this study is to find out what criteria that are the most essential or crucial ones in selecting the visualisation equipment for engineering design tasks. The emphasis is on user performance, cost of equipment and pleasure for the users.

2 Related work

Research within the area of virtual reality is quickly growing, but questions of an effective balance of performance, cost, immersion and amusement have not been treated in engineering design research. Existing engineering design procedures [4] [8] [10] do not cover VR-issues.

Looking beyond the sphere of engineering design, there is research done in the area of abstract information visualisation in cognitive science. Results from software design and psychological perspectives are interesting; work in that area shows that sense of presence, subjectively rated ease of use, and preference were all affected by degree of spatial cueing, but objective performance was not [6]. The most spatially cued virtual world produced slightly poorer performance compared to an equivalent hypertext interface, but was rated more enjoyable.

In social sciences, different visualisation systems have been used in order to examine how groups work together in multi-user virtual environments [1], while in the field of HMI (human-machine-interface) research focuses on task performance.

3 The study

3.1 Design of experiment

The experiment was decided to take place in a laboratory setting, to get a better control over the parameters. The experimental design was a *within-subject* design. Well-known methods are used in an uncommon combination, not

In order to get a statistically counterbalanced experimental design, possible errors, e.g. user experience and spatial ability, order effects as carryover and fatigue etc, must be eliminated or randomised. Initially four factors (*visualisation technique*, *design error*, *order of the experiments* and *respondent*) were manipulated in a 3×3 Greek-Latin square design [7]. The rows of the Greek-Latin square were permuted in a 3×3 Latin square design in order to add two additional factors: *order of design errors* and *observer*. There were three levels of the visualisation technique factor, six levels of the design error factor and three observers. For this, we needed 18 respondents doing six trials each, i.e. two rounds of three experiments each.

3.2 Observers and respondents

Observers were selected among master students from the Media Technology programme at Linköping University since they were familiar with the equipment and could therefore provide appropriate support to the respondents. Observers were remunerated.

Respondents were senior undergraduate engineering students at Linköping University. Engineering students were selected as adequate respondents since they should have reached enough technical maturity to understand the product functionality. An invitation was sent out by email to all students at the department and a selection among the replies was made. Factors for selection were age, education and sex, and respondents were selected to get as broad range as possible within these factors. Respondents were remunerated.

3.3 Products

The products studied were a stacker crane and a transfer terminal used for materials handling in warehouse storage systems. 3D CAD data (AutoCad) from the company were imported to the VR-modelling tool (WTK) and programme code was added for the input devices.

3.4 Technical description

Three different visualisation systems were used, see Table 1. This gives an experimental design with systems ranging from a kind of augmented reality to high-immersive VR. The latter one is connected to tracking systems (i.e. systems tracking the user's position, orientation and movements) and advanced input-devices (providing possibilities for navigation in the virtual environment and interaction with the objects) - with the effort to transport the viewer into the virtual design world. An SGI Onyx2 Matrix with eight processors (250 MHz MIPS R10000) and three InfiniteReality pipes powered the visualisation systems. This means that all visualisation systems provided real-time interaction.

Table 1. Equipment used in the experiments

Visualisation system	Size & type	Resolution	Input-device
A. Vortex with stereoscopic glasses (ChrystalEyes) with tracking	67" Back projected screen, CRT projector	1024×768 pixels	Magic Wand (InterSense), keyboard
B. Desktop-VR with stereoscopic glasses (ChrystalEyes)	21" monitor	1024×768 pixels	Keyboard, mouse
C. Desktop	21" monitor	1024×768 pixels	Keyboard, mouse

4 Performing the experiments

4.1 Pilot study

A final preparation for the data collection was the conduct of a pilot study. The pilot study contained experiments with three respondents and helped to refine the data collection plans by revealing a few errors in the experimental design. The errors were mostly related to inferior instructions to respondents and observers.

Experimental sessions were conducted in the VR-lab at Campus Norrköping, Linköping University, Sweden, over a period of one week. Before starting, participants were introduced during 15 minutes to the experiments and the functionality of the stacker crane and transfer terminal. For this purpose another type of product representation was used – a coloured screen dump from the CAD-environment, size A3.

An error, i.e. a geometrical modelling/design error leading to technical malfunction, was introduced into the digital product representation. The task was to identify the error as fast as possible and then explain it to the observer. Every respondent made six tests, two on each display and with a new modelling error each time. A time limit was set to 15 min. Respondents not being able to identify the error within this time aborted the search and the error was disclosed to them before they proceeded to the next station.

4.2 Data collection

A decision was taken to prepare for both qualitative and quantitative data. Time for solving the given problem was selected as a base for the quantitative data collection, and qualitative data was collected by tape-recording of respondents, a questionnaire and direct observation by the researcher and the observers. Using multiple sources of evidence, i.e. using different data collection methods, improves the probability of convergence of evidence and enhances reliability and validity of the study.

Each respondent was given a questionnaire, which was filled in after the experiments. The questionnaire contained 22 questions with 3 open questions and 19 tick-box questions. Most of them with ordinal scale. The first part concerned the respondent 'profile' and former experience of digital product representations and visualisation techniques. The second part dealt with the respondents' experience from, and feelings about, the different techniques.

In this study a company is involved, but only to deliver the CAD-model. This means that the study stands free from pressure from industry and other possible stakeholders.

4.3 Analysis

Data from questionnaires and measured times were inserted to a spreadsheet (MS Excel). Different calculations were made for each respondent, observer and visualisation technique, e.g. maximum, minimum and average times. Moreover, the number of successful experiments, i.e. number of experiments where the design error was found within the time limit, was calculated and analysed. The results were coupled to age, sex, user experience etc.

The tapes were registered, transcribed and condensed according to normal procedure and analysed together with observation protocols from observers and the researcher.

5 Results and discussion

The sample of respondents was as follows:

- 2/3 (12 persons) male and 1/3 (6 persons) female.
- 1 person 20 years old, 14 persons 21-25 years old, 2 persons 30-40 years old and 1 person older than 40 years.
- 78 % had no former experience of stacker cranes, 11 % had a little experience and 11 % had some experience of these cranes.
- All respondents had high experience of working with ordinary monitors. As regards working with the Vortex, 44 % of the respondents had no former experience, 28 % had little experience, 22 % had some experience and 6 % had a lot of experience. Note that this concerns only the visualisation technique - not the navigation system.
- Most of the respondents had little or some experience from looking on or working with digital product representations. Only one respondent was not familiar with digital product representations.

Results from studies on objective performance are presented in Table 2. Since the product and the environment were unfamiliar to the respondents, it is even more interesting to make a comparison of the visualisation techniques divided in first and second set of experiments of the study (Table 3), looking upon the first set as a training sequence. We can clearly identify learning during the tests, which is also a result seen in the questionnaire; 67 % of the respondents claimed that they understood the functionality of the products fairly well before the tests and 78 % very well after the tests. The functionality of the products was better understood by the act of working with the digital mock-up, according to respondents' subjective rating.

Table 2. Mean times and no. of successful experiments with different visualisation techniques.

	N° of successful experiments	Mean-time	Subjective rating of support, 'most helpful'
Total	71 (66 %)	4:45	-
A. Vortex	22 (61 %)	5:54	11 %
B. Desktop-VR	23 (64 %)	**3:41**	22 %
C. Desktop	**26 (72%)**	4:39	**67 %**

Table 3. Comparison of mean times and no. of successful experiments between 1st and 2nd set of experiments.

	First set of experiments		Second set of experiments	
	N° of successful experiments	Mean-time	N° of successful experiments	Mean-time
Total	29 (54 %)	6:55	42 (78 %)	3:31
A. Vortex	8 (44 %)	9:16	14 (78 %)	4:34
B. Desktop-VR	10 (55 %)	**5:03**	13 (72 %)	**3:01**
C. Desktop	**11 (61 %)**	6:56	**15 (83 %)**	**2:59**

The objective performance in the second round was found to be almost the same for the Desktop and the Desktop-VR solutions, whereas the subjective rating from the survey claimed the Desktop application to be most helpful. Largest difference between first and second rounds was found with the Vortex. Apparently the Vortex being the most immersive equipment, did not provide sufficient help to respondents this time, but we can clearly identify the improvements between the two sets of experiments and a fast learning. The difference between Vortex and Desktop is not very large in the end and considering that this was the first meeting with Vortex for 45 % of the respondents, it is possible that Vortex-times will shorten with more training/experience. "If I could work more with the Vortex, I'd probably get much better results", and other similar comments from participants were frequent in the questionnaires.

Most of the respondents expressed their joy of working with the equipment and especially working with the Vortex provided most pleasure (Table 4), but what is the relative value of *pleasure* versus *task performance*? HMI research has, as stated above, focused on task performance, but enjoyment has potential advantages for user motivation and focus on task.

Table 4. Number of highest rated techniques on the questions of intuition, usefulness and amusement.

	Which visualisation technique was the most ...		
	... self-instructive/ intuitive?	... useful?	... amusing?
A. Vortex	6 (33 %)	2 (11 %)	**15 (83 %)**
B. Desktop-VR	2 (11 %)	3 (17 %)	2 (11 %)
C. Desktop	**9 (50 %)**	**10 (56 %)**	1 (6 %)

All participants had experience from working with ordinary computer screens, but 44 % had no experience from the Vortex. What happens in the long-time perspective, when there is no novelty left and when users are getting accustomed to all of these systems? Will amusement decrease and usability and performance increase? Results of this study point in this direction.

Moreover, the study showed no clear relationship between respondent profile and preferred technique. Nor could a relation between age and performance be found. Concerning gender and performance, the female participants succeeded in finding more errors than their male colleagues, but used more time on average. Since the aim of the study was not to compare male & female performance, no further analysis was made.

Figure 1. Pictures from the experiments. From left: a) Vortex, b) Desktop-VR and c) Desktop.

Comments/results from questionnaires and observations show that:

- Desktop-solutions were preferred to Vortex due to sharper image on the screen. This can be explained by the fact that the resolution was the same on the different screens, making the pixels larger on the Vortex.
- Participants preferring Desktop-VR liked the 3D-stereo view, but, as above, disliked the somewhat more blurred image on the Vortex.
- Some respondents did not like wearing stereoscopic glasses, claiming the glasses being "to heavy", "irritating", "disturbing", etc. Therefore they preferred the Desktop-solution.
- 16 respondents claimed to have *developed a method* for more efficient searching during the tests. Two respondents did not do this, one searched randomly during all tests whereas the other one claimed to be using the same method all the time. The strategy, as observed, was almost the same in every experiment; first getting an overview of the product, then trying the main functionality and then finally looking on details.
- Most stated difficulties in solving the task concerned unfamiliarity with the navigation system, i.e. how to interact with the product models. Even though some participants had used the WAND before, the navigation model in this application was a new experience to all respondents.
- Respondents said that the change from Desktop to Vortex and vice versa was difficult - they had to change the way of thinking, since the two navigation models built on different navigation-principles. A result from this is the proposal of using similar navigation principles in the VR-application as in the designers' ordinary CAD-tool to shorten time for adaptation. Since there are a number of different CAD-tools on the market, this will result in that all designers will not use the same navigation model.

Along the experiment, respondents became more and more successful in their attempts of error-identification. Time to solution decreased and the number of identified errors increased (Table 5). The subjective statements also confirm this - participants found error searching more easy as the experiments continued.

The aspect of learning can be divided into different aspects; understanding the model, understanding the product functionality etc. According to subjective ratings, respondents learnt more easily how to navigate with the computer mouse than with the WAND. Important to notice is that the mouse was used on two stations (4 experiments per respondent) while the WAND was used only together with the Vortex (2 experiments each).

Table 5. Mean-times trough the six trials show increased capability of error identification.

Try	1	2	3	4	5	6
N° of successful experiments	8 (44 %)	10 (56 %)	11 (61 %)	13 (72 %)	14 (78 %)	**15 (83 %)**
Mean time	07:35	08:31	04:00	**03:07**	03:28	03:56

Exposure order was totally randomised but no significance was found to performance or other factors.

As regards time and cost spent for this work in the view of the designer, the majority of time is spent on preparations, e.g. modelling and programming. This means that the actual work of the designer is rather limited in time and cost but is preceded by a costly task.

Finally, one of the most important factors when arguing for reducing costs in product development is still the cost of equipment and software. Not including the cost of the computer or the software, but just comparing the visualisation systems, costs are shown in Table 6.

Table 6. Cost of the different visualisation systems, the SGI Onyx2 Matrix is not included.

	Visualisation system	Approx. Cost [Euro]
A.	Vortex and tracking system ChrystalEyes (InterSense)	90 000 € 900 €
B.	Desktop-VR ChrystalEyes (InterSense)	1100 € 900 €
C.	Desktop	1100 €

6 Conclusions

A first conclusion is that the study design worked well in terms of task performance. Nearly 50 % of the participants were beginners with these kinds of systems; nevertheless they succeeded in carrying out the engineering task. The task was suitable for identifying differences in support from the visualisation techniques.

The major result from the study is that the Desktop and Desktop-VR applications were the most efficient ones for the error-searching task, both in objective and in subjective terms. Experiments with the most immersive system, the Vortex, took 50 % longer time. The same number of successful attempts, i.e. result, was achieved. The Vortex was rated highest on pleasure/amusement and ease of use. Moreover (complementing results in [6]), participant performance improved with task learning. The improvements were greatest on the Vortex.

There were large differences between respondents in task performance, which implies that the answer to the introductory question: "Criteria for selecting visualisation equipment" is more complex than imagined at the beginning. When selecting visualisation techniques (as well as devices and available modes of feedback), it is important to adapt them for the specific users and their experience, abilities and needs. No strong correlation was found between task performance, joy, age and gender. However, age and gender is important in terms of experience and technical adeptness. Moreover, cost of computers, devices, hard- and software as well as technical support have to be considered together with factors as resolution, frequency of use, existing CAD-environment but also "softer" questions of pleasure and amusement.

Future work needed in the area includes a more generalised study beyond error finding tasks. A wider population, including designers with more experience of engineering design and high-immersive VR-equipment would be fruitful.

Acknowledgements

The author would like to thank Transnorm Systems AB for co-operation. Thanks also to all the members in the Scientific Visualisation Group at Campus Norrköping for providing help whenever necessary. This research project is part of the ENDREA research programme.

References

[1] Axelsson, A-S. et al., "Collaboration and Communication in Multi-User Virtual Environments: A comparison of Desktop and Immersive Virtual Reality Systems for Molecular Visualization", Proceedings of the Sixth UKVRSIG Conference, Salford UK, 1999, pp 107-117.

[2] CyberEdge Information Services, Inc. (CEIS), "The Market for Visual Simulation/Virtual Reality Systems", Market study report, New York, 3 ed, 2000.

[3] Gabbard, J. and Hix, D., "A Taxonomy of Usability Characteristics in Virtual Environments", Department of Computer Science, Virginia Polytechnic Institute and State University, 1997, http://csgrad.cs.vt.edu/~jgabbard/ve/taxonomy/

[4] Hubka, V. and Eder, W. E., "Engineering Design: General Procedural Model of Engineering Design", Heurista, Zurich, 1993.

[5] Johansson, P., "Comparing digital and physical mock-ups", Proceedings from NordDesign Conference, Copenhagen, 2000, pp 235-244, ISBN: 87-90130-28-6.

[6] Modjeska, D., "Hierarchical Data Visualisation In Desktop Virtual Reality", Doctoral Thesis, University of Toronto, 2000.

[7] Montgomery, D., "Design and analysis of experiments", Wiley, New York, 4 ed, 1997.

[8] Roozenburg, N. F. M. and Eekels, J., "Product Design: Fundamentals and Methods", Wiley, Chichester, England, 1996.

[9] Söderman, M., "Product Representations: Understanding the product in the Design Process", Licentiate Thesis, Chalmers University of Technology, Göteborg, 1998.

[10] Ulrich, K. T. and Eppinger, S. D., "Product Design and Development", McGraw-Hill Inc., New York, 2000. ISBN: 0-07-116993-8.

Per Johansson

Division of Machine Design, Department of Mechanical Engineering, Linköping university, 581 83 LINKÖPING, Sweden, Tel: +46 13 281123, Fax: +46 13 285670, E-mail: perjo@ikp.liu.se, URL www.machine.ikp.liu.se

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

INNOVATION USING THE DIGITAL PRODUCT – THE USE OF VIRTUAL REALITY IN PRODUCT DEVELOPMENT PROCESSES

A M Kunz and M Meier

Keywords: collaborative design tools; methodology; information technologies; product development process; design teams; computer supported collaborative work

1 Abstract

The technique of virtual reality has been developed so far that it isn't a privilege of large industries anymore but also smaller enterprises can take advantage of it. In this case the main item is the Digital Product which is used for visualization tasks. The scope of this paper is to demonstrate how the Digital Product and its visualization is used in different enterprise processes and which benefits arise from that.

2 Introduction

Starting with the market investigation up to the market entry the product development process has to be carried out in an increasingly shorter time. Nowadays the time-to-market of a product is critical, only a few weeks delay can decide considerably about success or failure of a new product. In order to solve these problems the method of concurrent engineering is used. However, this is not done completely. Many processes still are serially or are removed from the product development process. The reason for this is that the contemporary product development process is still based on a paper-based method which has been slightly modified in order to use it in a team.

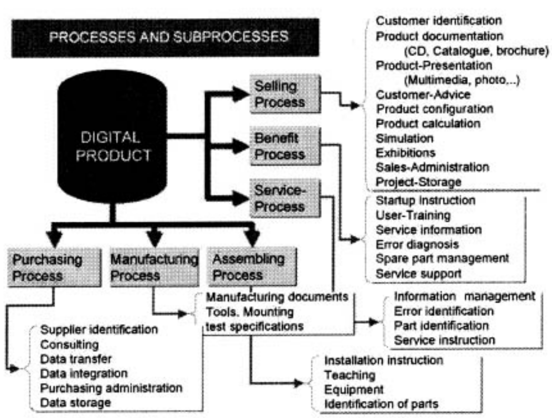


Figure 1. The Digital Product - the "entirety of all product data"

Every part of the product development process has its own data world (Fig. 2), clear boundaries exist and only a document exchange is done. The responsibility of the technical department is limited to the preparation of perfect production data sheets. The sales department generates catalogues (on paper or CD) without basing them authoritatively upon the data of the Digital Product. Faced with this current problems the future way is drawn: The next step will be to use state-of-the-art means and tools to overcome these boundaries. The product data and thus the digital product become the integral and strategic turntable of the whole enterprise.

3 Motivation

Information and/or data arise from the entire development stage and the entire profit stage of a product. The entirety of all relevant product data, that are generated and administrated consistently, represent the real product and are named "Digital Product". Product development processes use the data of the digital product, add new data or modify data of the Digital Product.

Dissecting these processes it becomes clear that in most cases a person is the producer or the receiver of these data. He uses specific services which allow him to carry out these processes based the digital product. Such services are for example:

visualize: The content of the Digital Product and/or parts of it is made accessible to the customer. In this case it is not distinguished between a graphical and a textual edition. In the case of visualizing functional aspects are in the center of interest.

simulate: The Digital Product or parts of it are used to simulate unknown properties. The simulation can be used to find out functionalities of the product or of the processes.

archiving: Data of products or processes and data coming from the customers feed the Digital Product. The archiving of these data allows an instantaneous access to them.

documenting: The content of the Digital Product is used for a textual or graphical documentation of the product or the processes.

transferring: The content of the Digital Product or parts of it are accessible over a network in order to be used by other services or processes.

presenting: Parts of the Digital Product are extracted, processed in textual or graphical form and represented in a comprehensible way. Optical aspects are in the center of interest.

integrating: Parts of the digital product are integrated into the databases of other Digital Products in order be used by foreign processes and services.

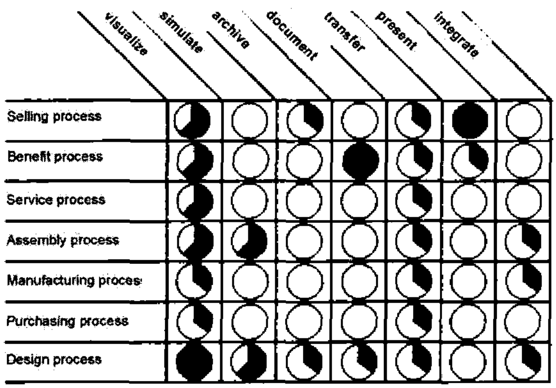


Figure 2. Assignment of the services to the enterprise processes

Not all enterprise processes have to use all services (Fig.2). The services guarantee the connection of the processes and sub-processes to the Digital Product. In particular the services "Visualization" and "Data transfer" are of central importance. The task of virtual reality will be to represent complex data structures to the persons in an acceptable form. Furthermore virtual reality must enable the persons to interact with the Digital Product in an effective and ergonomic way. In this manner all enterprise-processes will be enabled to use the data of the Digital Product in order to achieve the desired shortening of time-to-market.

4 Contributions

Virtual reality and with that the Digital Product gain more and more importance within industrial enterprises. In future an essential task of virtual reality will be the replacement of the physical preproduction model in the primary development phase. The physical preproduction model must be distinguished into the following functional types:

Design-Prototype: It is used for the inspection of the design draft concerning aesthetic, optical and ergonomic aspects; mechanical qualities are not of any importance; mostly a functionality is not integrated (Fig. 3).

Geometrical Prototype: It is used for the inspection of the form accuracy and the accuracy of fit. Only the geometry is in the center of interest but not the material itself.

Functional-Prototype: The functionality of components is tested with this prototype. This inquires an identical material as the one for the series production.

Technical Prototype: All functional aspects of the system are checked.

The scope of this paper is to give some examples how the digital prototype and the Digital Product is currently used in enterprises and which efforts arise from this. A discussion is given whether the Digital Product is able to replace the physical prototype.

The essential task of virtual reality in the product development process is the visualization of objects and functionalities. Today geometry and design questions of new products already can be completely solved with the geometry visualization of the Digital Product without the need to create a design preproduction model (Fig.3). In accordance with the tasks in a product development process (Fig.1) the services of visualization and communication can be essentially supported by this digital preproduction models.

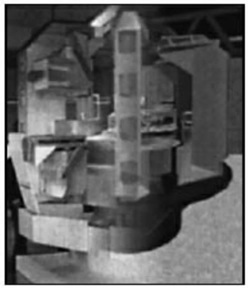


Figure 3. Virtual reality replaces the geometrical preproduction model

The advantage of digital geometrical preproduction models for the companies is obvious: the costs for the preparation of a real preproduction model drop in the same way as its manufacturing time. Thus it is possible to achieve a higher product quality considerably earlier and to influence the time-to-market significantly (Fig. 4).

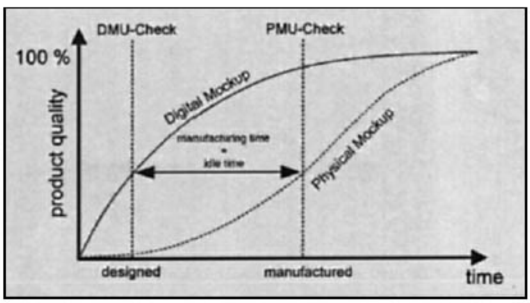


Figure 4. Shortening of the time-to-market with the digital preproduction model

The advantages of digital preproduction models become clear in particular if constructive changes must be made and visualized afterwards. Making changes in physical preproduction models is very time consuming because of the additional manufacturing time. In this case the digital preproduction model allows it to represent the changes tridimensionally after a very short time.

Another example of the promising usage of virtual reality in the product development process is given by the FMEA, the failure mode and effect analysis. FMEA is a means to recognize potential design and construction mistakes within an early phase of the design process. The FMEA is made in order to prevent faulty constructions from being manufactured or from being sold. Thus this method is useful to save high costs that are caused by faulty products on the market. In cooperation with a large power tool manufacturer the existing FMEA-method was basically analyzed. One of the main advantages of this method is the systematic scrutinizing of the design. Thus it will be possible to consciously integrate participants from other fields of an enterprise. However this is also one of the most problematic points because of the underlying two-dimensional drawings. Only persons with practice in reading these drawings have a contiguous illustration of the part on which the discussion is about.

Based on the data of the Digital Product an additional visualization of the geometry was integrated into the FMEA (Fig. 5). This allows to carry out the methodical analysis considerably more efficiently and thus also less expensive. Using VRML (virtual reality modeling language) as a description language for the objects guarantees a low-cost usage of these visualization possibilities on standard computer systems. The geometrical data as a component of the Digital Product are visualized simultaneously during the FMEA in addition to the form to be filled. All participants synchronously obtain the same visual information and thus a moderation of the session is facilitated. A standard personal computer is used in order to realize a low-cost solution for small and medium enterprises. Since it is necessary for the realization of the FMEA to visualize simultaneously both the form and the object, two projectors are used. They are connected to a personal computer by two separate graphics channels.

Figure 5. Product visualization within the FMEA

A simple navigation and visualization tool is used during the FMEA. The moderator can carry out a simple navigation, i.e. swiveling, displacement and so forth. Thus the relevant part can be investigated in detail. The Digital Product is used in order to allow the services visualization, engineering and communication.

In the case of very complex objects that have to be designed, as for example during the planning of bigger plants or machines, it is desirable to integrate the customer into the product development process more closely. In order to realize this demand a new interaction system was developed. Some persons can use this system for a simultaneous planning of complex objects. The involvement of the customers requires an easy use of the system without any training phase (Fig. 6).

Figure 6. Working at the Buildit-System

"Sit down and collaborate" - could be the motto because everybody can become operable in a few seconds without prior knowledge of this system. The positioning of machines, the navigation in the room as well as adding new elements is done by the displacement of an interaction brick. The results of the interaction are immediately visible both in the outline and in a tridimensional view. With the aid of this system and thus the Digital Product the team elaborates the solution of a planning problem.

A further important area of application for virtual reality is the representation of functional relations. In a very early stage of the product development process instructional materials and assembling instructions can be performed with the use of the Digital Product (Fig. 7).

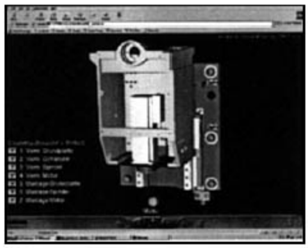


Figure 7. Virtual assembly instruction

The representation allows it to train the assembling of a product interactively. Next to the geometrical representation textual complements supply additional information. The units to be examined are not static, but they can be turned, rotated or zoomed by the user. Thus it becomes possible to examine geometrical limiting conditions and functional properties and to learn about the product before it is manufactured.

Large products as for example plants are unsuitable for exhibitions since they usually have very high transportation and build-up costs. The only way to show these products to the customers consists in the use of virtual reality. In order to check whether an exhibition can be done only by the use of a digital mockup a complete asphalt factory was modeled (Fig. 8).

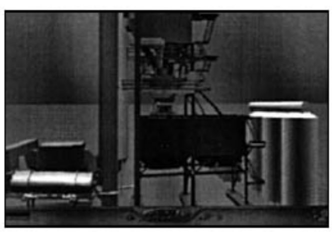


Figure 8. Modeled asphalt factory

The practical test of such big VR-models showed that the computer performance is yet not sufficient enough for a fluid visualization. Additional problems came up with the projection of these models during the visualization on booths. Within the bright environment of a booth the projectors must be very bright and therefore very expensive in order to realize an appealing representation of the new product. In addition the clientele in mechanical engineering does not show acceptance for a complete digital exhibition of products at the present time.

5 Conclusions

The examples from the above show that virtual reality and thus the Digital Product becomes increasingly important for the industry. New low-cost technologies will allow to use the data of the Digital Product intensively in the different enterprise processes. In future the product data won't be used only for manufacturing but also for other enterprise processes:

- Increase of the product quality: The visualization of the geometry with the data of the digital product allows to recognize potential sources of errors early and to achieve a higher quality in a shorter time. Further efforts in this field will allow a step-by-step approximation to the digital functional preproduction model. Next to the geometry visualization this prototype has to simulate also physical qualities in order to enable an initiation of a machine on a virtual basis. Also extreme test runs at a machine become possible that up to now led to a destruction of the physical functional preproduction models. The digital product already includes the necessary data sets but up to now an effective linking is still missing. If this succeeds satisfactorily a machine or plant including their control and their physical behavior can be simulated. The amount of time from the idea up to the finished product can be significantly shortened.

- Involving the customer during the product development process: Future products are not branded articles anymore but they are manufactured especially to the orders and needs of the customer. This can be done best by involving the customer very early into the product development process. The customer specifies "his" product without deeper knowledge into its manufacturing. He participates in the generation of the Digital Product; "his" product can be visualized early and possible changes can be done without additional costs.

- Improvement of the internal communication: If the common data set of the Digital Product is consistently used it is guaranteed that the data keep up-to-date. Already existing visualization also can be used in the manufacturing process.

- Employee and customer education: The techniques of virtual reality will allow to explore the products interactively, to inspect them and to gain first experiences before the first specimens of this product go into production. Thus it will be possible to train persons on complex products in a very early stage. Before a new product comes onto the market experienced personnel is available (Fig. 9). Also the customer can gain experience with his future product so that he is operationally immediately after the delivery of the product.

Figure 9. Education with the virtual model

- Marketing: Exhibitions and product presentations are one of the most expensive parts within the life-cycle of a product. The companies are endeavored to represent an comprehensive product range to the customer on big booths and in extensive product catalogs. Printing costs, postal charges, exhibition costs and haulage are very high. If only a small part of the product presentation could be done with the use of virtual reality there will be the possibility to save a large amount of money. In addition it will be possible to present in the same booth size a considerably larger product palette. Since the represented geometries originate from the digital product these can be made available to the customers. This will allow to generate a brochure that fits especially onto the questions of the prospective customer.

- Service: Especially in the case of custom-built products an individual service is necessary. The digital product and its visualization will help to prepare service tasks in order to carry out a maximally effective maintenance on the customers products.

6 Future work

Future work will handle the use of the virtual reality in medium and small enterprises. This will be an optimization of the data access of the Digital Product by the means of virtual reality, for example a simplified generation of the virtual illustrations from the existing CAD-drawings. In addition new areas of application are supposed to be opened for the visualization, for example in the product configuration or in the preparation of virtual assembly or operating instructions.

References

[1] F. Dai, P. Reindl; „Enabling digital mock-up with virtual reality techniques"; ASME Design Technical Conference and Computers in Engineering Conference; August 1996, Irvine, CA, USA

[2] M. Meier; „ETH-Know-how für den Werkzeugmaschinebau"; Technische Rundschau Nr.20, 1998

[3] C. Lämmle, Y.M. Wright; „Virtual Reality"; ETH-internal research project; 1998

[4] O. Niedermann; „Nutzenanalyse des Einsatzes der Virtual Reality Technologie in der Failure Mode and Effect Analysis"; ETH-internal research project; 1999

[5] S. Frei; „Erstellung eines FMEA-Software-Prototypen mit integrierter Visualisierung und Datenbankankopplung"; ETH-internal research project; 2000

[6] Y. Rieffel, D. Hautle; „Entwicklung eines virtuellen Messestandes"; ETH-internal research project; 2000

[7] T. Holmer; "Brainstorming mit Hypertext"; Diploma thesis; Carl-von-Ossietzky-University Oldenburg; 1997

Mr Andreas Kunz, Mr Markus Meier
Centre of Product Development, Swiss Federal Institute of Technology Zurich, Tannenstrasse 3, 8092 Zurich, Switzerland, Tel: +41 (1) 632 5771, Fax: +41 (1) 632 1181,
e-mail kunz@imes.mavt.ethz.ch,

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

EXCHANGING ADMINISTRATIVE PRODUCT DATA IN THE AUTOMOTIVE INDUSTRY USING STEP

K Pagenstert, H Axtner, D Lange, and P Deasley

Keywords: Product data management, configuration modelling, STEP, product architecture, automotive engineering.

1 Objectives

Considerable experience exists in the automotive industry in exchanging graphical information between CAD systems. Next to no experience, however, exists in exchanging BoM (Bill of Material) and product configuration related information. The characteristics of the automotive product - high product variety and complexity, together with high build-to-order volumes – require the use of complex semantic and logical rules in BoM systems. These pose a particular challenge to the management and exchange of product data. As the degree of business integration between automotive companies and suppliers grows, the ability to exchange BoM-related information is increasing in importance [1] (Figure 1).

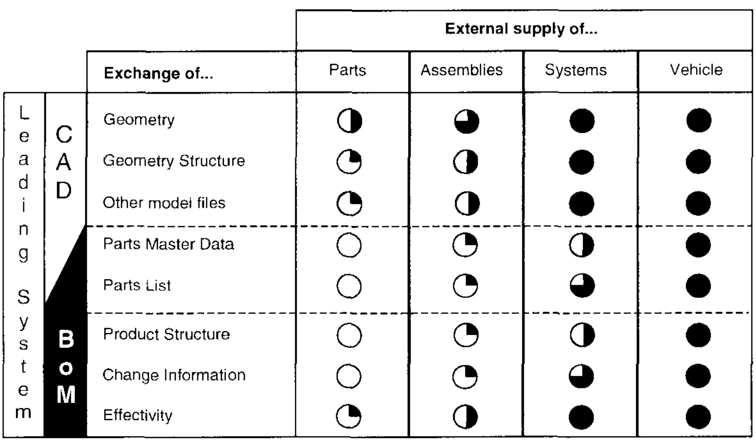


Figure 1: Data Exchange significance for Automotive Industry

Recently, key elements of the functionality required for the representation of automotive BoMs have been considered within the ISO STEP [2] standard. The objective of the STEP standard, to achieve *"an unambiguous representation of computer interpretable product information through-out the life of a product"* [3] promises the following for the sharing and exchange of BoM information:

- First, that the framework of design methods and implementation techniques that STEP provides would facilitate a software development process targeted at developing a BoM exchange application.

- Second, that the data functionality contained in the industry-oriented STEP AP data schemas ("Application Protocol") would cover the functionality requirements of the automotive BoM.

Among the STEP application protocols there is one that explicitly addresses the design process of the automotive industry: AP 214 "Core Data for Automotive Mechanical Design Processes [ISO-10303-214]", also referred to as "STEP AP 214" [4]. The BMW Group therefore decided to investigate and assess the potential of the STEP AP214 standard for the exchange of BoM-related information with other automotive companies and suppliers.

2 Method

2.1 Investigative Project

To investigate how well STEP, and in particular the AP214 schema, could be used for the automotive BoM, a project was defined with the objective to

- Investigate the quality of fit of BMW's and Rover's engineering BoM in the STEP AP214 schema
- Investigate the STEP implementation architecture and methods
- Investigate the feasibility of STEP as a translation enabler for engineering BoMs in the "real world"

by realising a STEP-based BoM translation application.

The scope of the project was focussed on two key elements of the engineering BoMs: parts and product coding management [5, 6]. The engineering BoMs were chosen so as to ensure that the most complex exchange requirements would be encountered, which would not be the case with single-structure design (CAD) BoMs. Because BMW and Rover have fundamentally different approaches to managing administrative product data [7], this was seen as representing a suitably generic test case. The development experience gained during the project and the resulting application was to provide the answer to the following questions:

1. How well can the BMW and Rover BoMs, specifically the product coding as the key to the BoM, be represented in the STEP AP214 data models ?
2. Can the STEP AP 214 data model be used to reference the BMW and Rover BoMs ?
3. What are the advantages and disadvantages of using STEP as an interfacing technology between BMW and Rover BoMs compared to a bespoke solution ?
4. What IT issues need to be considered in implementing STEP in terms of performance, software and hardware requirements etc. ?

To ensure that the functionality provided by the application allowed for a representative assessment of the STEP standard, particular attention was paid to ensuring the adherence to the ISO development guidelines (data modelling, application of appropriate schemas, etc.).

The final application was then tested with a selection of automotive BoM information from BMW and Rover product ranges to investigate data quality and performance issues.

3 Results

3.1 Software architecture

A software application was developed with the following basic architecture:

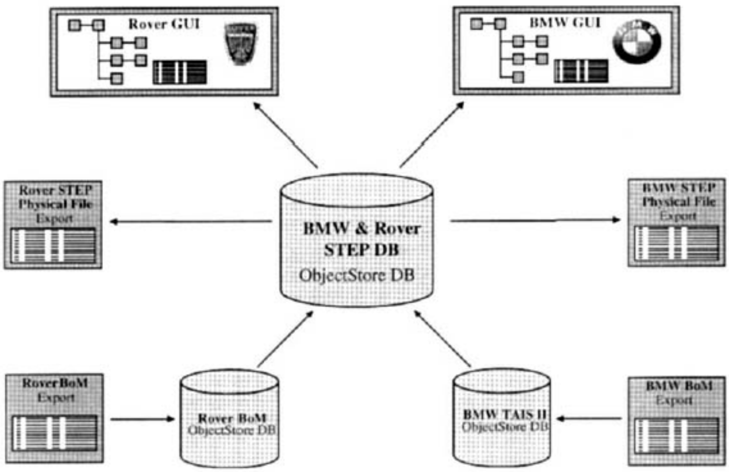


Figure 2: Application architecture

Within the application three data schemas are used: one to import the BMW BoM, one to import the Rover BoM, and one representing the BoM-specific subset of the STEP AP214. All three schemas are described in EXPRESS, the formal description language used in the STEP standard. Because of intricacies of the STEP development methodology, the ARM was used instead of the AIM of AP214 (AIM - Application Interpreted Model, ARM - Application Reference Model). The major reason for this was that the ARM is closer to user terminology, and is less complex to instantiate than the AIM. Using the ARM does not result in a loss of expressive capabilities since the AIM only is a more generic representation of the same constructs. The functional model is described with four "mapping" models, which comprise the import processes of the native data exports from the BMW and Rover BoM systems to the STEP database. Both the BMW and the Rover native data are mapped to one AP214 STEP database, which is capable of exporting appropriate STEP physical files. It uses the identified AP214 subset as a basis for the physical data schema. An OODBMS ObjectStore database is used as the database management systems for all three databases.
The BMW GUI is connected to the BMW STEP database, which contains BMW data, as well as the mapped Rover data in the BMW interpretation of the STEP format. The Rover GUI is connected to the Rover STEP database, which contains Rover data as well as the mapped BMW data in the Rover interpretation of the STEP format. The BoMs can be exploded from both the BMW and the Rover engineering BoM STEP instantiations.

3.2 Software functionality

The application is able to display a vehicle structured using the BMW and Rover engineering BoM approach (Figure 3). The application allows the user to navigate through the BMW and Rover product structure down to the lowest level of the product structure. The user is able to carry out the key BoM product structuring functions, such as the selection of interior/exterior colour combinations, country specific options and special packages. The application is able to validate the selections carried out by the user, and will report illegal combinations of options. Significantly, the application allows the user to switch between a BMW and Rover view of a specific configured vehicle i.e. the STEP AP 214 data model can be used to reference between the BMW and Rover BoMs.

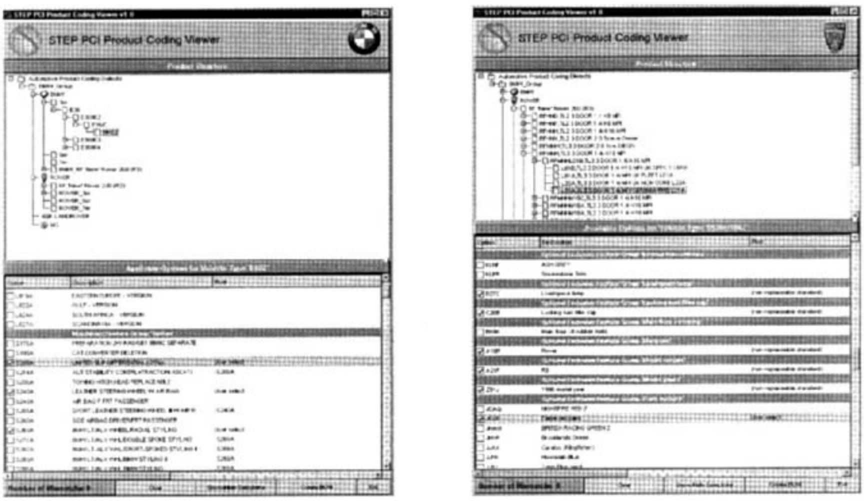


Figure 3: STEP-based BoM translator – BMW and Rover views on product coding

If the option selection comprises a 'valid vehicle', the user may create a BMW and/or Rover BoM view on the vehicle-specific parts list (Figure 4). This creates the list of only and all those parts which correspond to a specific car configured according to the options selected by the user.

4 Evaluation

4.1 Functionality of STEP-based Application

It was possible to realise the defined functional scope in the application in conformance with the STEP AP214 standard, both for the BMW as well as the Rover engineering BoM approach. It was possible to realise the key "main stream" elements of an automotive engineering BoM, including product coding, parts management and demand calculation. The exchange of BoMs between BMW and Rover could be carried out. Switching between a Rover view on the product structure of a Rover and a BMW, as well as a BMW view, is possible. A STEP AP214-compatible export of BoM data is possible.

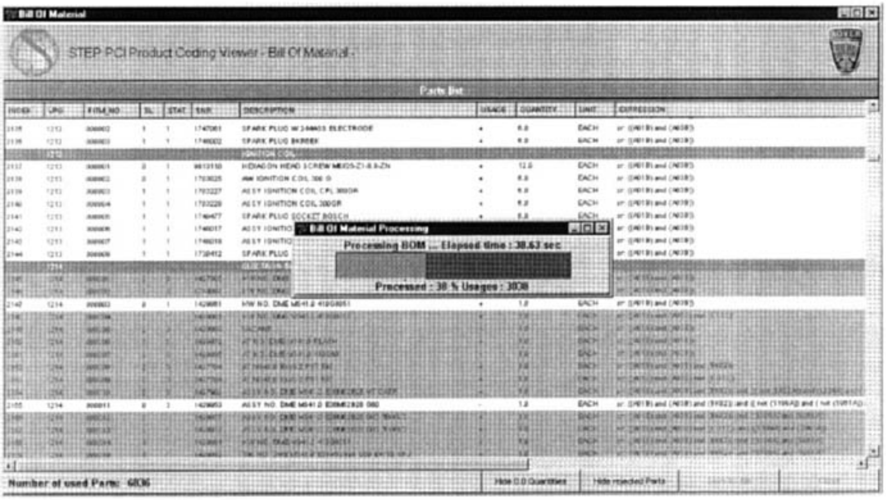


Figure 4: Creation of an order-specific parts list (Rover view – BMW view similar)

4.2 Quality of fit of BMW and Rover BoMs in the STEP AP214 Schema

Both selected subsets of BMW and Rover's Product Model Data can be mapped to AP214 using Conformance Classes (CC) 6 and 8, mostly using the Units of Functionality (UoF) S1, S3 and S7 [8]. Whilst the product coding logic of both BMW and Rover can be represented with a good level of conformance to STEP AP214, the schema's capability to represent complex products is clearly biased towards a Feature based approach like that of Rover. A Rover Exclusive Feature Group can be simply mapped to the entities '*specification_category*', and a Feature will result in a AP214 '*specification*'. Availabilities can then be represented by 'effectivities' as understood by the STEP AP214 schema. The mapping of a BMW type rule-based Product Coding is not as straight forward. It results in complex instantiations of the entities '*class_condition_association*'(s) with '*specification_expression*'(s) for every basic operation within the Boolean expression that conditions the part, and assigns them to a Model Type. This generated a huge number of instantiations that need to be followed whenever the underlying logic needs to be resolved.

4.3 Performance of STEP-based BoM Translators

The instantiation issue mentioned above is manageable for applications that do not have major performance requirements on data, timing or throughput. Systems within the productive backbone of an automotive manufacturer, however, are usually under an immense performance pressure. Timing windows for basic processing operations do not allow for a data and database model that requires tens of entity instantiations for a single, simple rule attached to a single part. This inherent issue has a crucial, detrimental impact on run-time performance.

4.4 Implementation experience

Efficient Development Environment

One of the key benefits the project team could identify in the use of STEP as the basis for an application was the implicit data modelling techniques and the standardised vocabulary. The advantages this offers, especially to international, multi-location development teams, should not be underestimated. The ISO 10303 vocabulary is reasonably well defined, and minimises the context-based, company background or language-related misconceptions which contribute to the immense waste that occurs in many software development teams. The standardised STEP environment defines a number of tools - and with them, procedures - that are to be used, especially in the critical early development stages.

Customisation of established company development models required

The complete development process itself - from the initial idea to the product deployment and customer care concept - is not covered by STEP. There no publicly available ISO 900x guidelines either, so in-house software development models will usually need to be adopted to the new methods and tools that accompany an AP 214 software creation process (comparable to the difference between an OO and standard development cycle).

Development support tools have weaknesses

The tools which are at the moment available in the STEP environment are in the experience of the project not optimal for the development of software used under productive conditions. Even the most widely available "STEP-tool", an EXPRESS(-G) modeller, can only be bought from a very small number of suppliers, and there are few productive users (and level of support). The AP 214 has just become stable as a schema, and the available tool-kits have a distinct "academic" bias - they lack performance, scalability, and often even stability. Some tools are not available at all e.g. an automatic testing environment (although one can employ tools supporting the selected implementation language). While the typical STEP development architecture works well for a rapid prototype approach, an interpreted language like Tcl would not be the priority choice for an application running under production load.

Support of distributed applications limited

Data access and navigation on shared STEP repositories is not covered by ISO 10303 itself, and while there are first approaches to e.g. migrate STEP over CORBA buses, these are more technology trials than efforts to establish industry wide best practices or readily available tool-boxes. In view of the rising importance of distributed computing and scalability, and the fundamental objective of STEP to represent a unified data integration environment, this is a key weakness within the standard.

Availability of external resources limited

There are very few STEP developers available in Europe today - and even less with a good conceptual grasp of BoM-related issues (most of the know-how is focused in the CAD area). While there are some significant suppliers actively supporting STEP AP 214 (like SAP or debis), the amount of skilled resources they can offer to a customer is very limited, both because their resource base in this area is low, and because the existing resources are already well employed.

4.5 Fundamental requirements of exchanging BoM data

The complex semantic functionality and company-specific aspects of most automotive BoMs

mean that the exchange of BoM data poses a different challenge than the exchange of CAD information. CAD data is expressed in terms of geometrical relationships, which are universally defined and understood. Exchanging a CAD file with a circle, for example, poses less of a problem. BoM data, however, is expressed using semantic and logical structures which are largely company-specific i.e. not only does a common understanding need to be created concerning a specific entity of information, but the information itself also needs to be interpreted within the context of a company to be instantiated correctly i.e. to make sense.
For example, a typical BoM entity found in many automotive BoMs is the entity "feature". This can mean any number of things, depending on the company in which it is used. Just equating features between two BoM environments will obviously lead to misunderstandings. Exchanging and sharing BoM information therefore at its most fundamental level depends on being able to achieve a common understanding as to the significance and context of the information which needs to be shared. This is largely independent of the "technology" used. The STEP environment (tools, procedures, data schemas) can minimise the required effort, but it cannot replace this activity.

5 Conclusion

The application developed by the project team allows the exchange of BoM-related information between BMW and Rover, two companies that have fundamentally different engineering BoM approaches. The key functionality of the BoM product coding could be covered using the STEP AP214 schema. Nevertheless, there are areas where the schema needs to be improved. A number of attributes that are present in the BMW and Rover BoM can not easily be mapped to appropriate AP 214 entities - not due to a lack of suitable entities, but because there is no agreed approach within the industry as to the use of the entities i.e. a recommended practice. The overall vehicle configuration and the parts master data aspects of the automotive BoM, for example, need to be specified more precisely. There are also areas of the AP 214 in which the schema is not generic enough to cover the requirements of the whole automotive industry. A number of attributes and constructs have been defined in a restricted fashion for no apparent reason (e.g. a '*specification*' must belong to only one '*specification_category*' thus '*specifications*' can not be mapped 1:1 to features as they are widely used in the automotive industry). A number of issues where identified which indicate that compared to bespoke solutions, applications which are based on a complete STEP-based implementation are not suitable for on-line, transactional type BoM applications. STEP should however be considered as a foundation for:

- Applications which require the sharing of data, especially when the integration of sophisticated information (such as the BoM) is required between internal and external development and manufacturing partnerships

Because the number of partners in any development effort in the automotive industry is increasing significantly, and the role of suppliers in the engineering process is growing to have the same significance as in-house engineering units, the automotive OEMs need to find a way to open and integrate their bespoke and standard CAx, PDM or Configuration Management applications. A common data backbone within and without the company can allow the seamless integration of automotive processes, for example through the use of transparent STEP gateways into bespoke PDM & BoM systems.

- Long-term data storage solutions

For liability reasons the automotive OEMs are expected to preserve data for 30+ years. Due to the nature of the IT business, it is likely to be very difficult to reactivate bespoke BoM systems in the future. A STEP-based approach will allow the present data to be largely vertically up-gradable, due to the standardised nature of STEP.

For a potential applicants of STEP-based technology this means:

- the standard STEP schemas should be employed wherever feasible
- if there is a level of freedom in choosing a data representation, then the one employed in STEP should be chosen, as this will ensure maximum compatibility once this data needs to be transferred to/from an internal or external partner
- the level of granularity as documented in STEP AP214 should be chosen wherever feasible, as this will avoid or at least reduce the difficulty of the complex mapping processes required to instantiate the considered information properly
- a neutral method of documentation should be employed, not a system specific one i.e. the data model and application logic should be documented, and not only the (application-specific) implementation
- A number of issues need to be considered in any STEP development which do not in themselves have anything to do with STEP itself (e.g. lack of skilled resources)

References

[1] Rose, B.: "Milliardenverluste bei Automobilbauern" in VDI Nachrichten, Nr 51/52, December 1996

[2] STEP stands for: STandard for the Exchange of Product Model Data. The term is used to refer to the ISO Standard Nr. 10303.

[3] ISO Reference series: "Industrial automation system and integration - Product data representation and exchange", 1994

[4] ISO Reference series: "Application protocol: Core Data for Automotive Mechanical Design Processes", 1995

[5] Grupp, B.: "Aufbau einer optimalen Stücklistenorganisation", Technische Akademie Esslingen, Expert Verlag (1995)

[6] Mather, H.: "Bills of Materials", Dow Jones-Irwin, American Production and Inventory Control Society (1987)

[7] Pagenstert, K. et al: "Description of Project Product Coding with STEP", PDMI2 Work-Group 13 Report, ProSTEP, 1997

[8] CCs and UoFs are segmentations to an AP that are defined in ISO 10303-1.

Konrad Pagenstert

BMW AG, TK-201, 80788 Munich, Germany, Tel: +49-(0)89-382-33152, Fax: - 44602

This project would not have been possible without the kind support from the Rover Group.

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21–23, 2001

INTEGRATED DEVELOPMENT OF GEAR UNITS – PRODUCE- AND WORKFLOW MODELLING

A Dyla, B-R Höhn, and K Steingröver

Keywords: integrated product design, gears, gear units, product model, product architecture, product data management, STEP, ISO10303

1 Introduction

The intention of this contribution is to outline the scientific research and development results of a computer based workplace for the development of gear units (figure 1) and its implementation in industrial processes. A unique product model combined with Product-Data-Management (PDM) functions enables a seamless data- and workflow during all stages of product development.

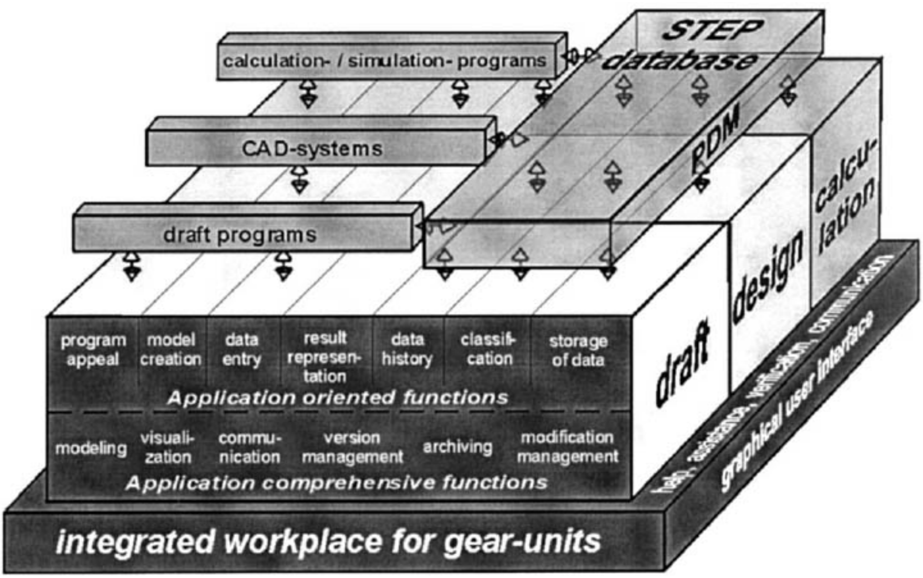


Figure 1. Workplace for the integrated development of gear units

The paper includes the purpose and procedure of product- and workflow modelling and the implementation in the existing area of well established calculation, simulation and modelling (CAD) programs which are connected together by a continuous data flow. The whole data flow is managed by PDM functions which are implemented within the working place. The paper also renders the experiences during the research project, starting with the process modell, its implementation and necessary changes in industrial product development. It is dedicated to show the usage and acceptance of a project specific prototypical computer based workplace for gear units, too.

2 Motivation and Aims

During the product development process a lot of different computer systems are needed. They can be categorized in calculation/simulation and design (CAD) systems. Figure 2 shows the usage of some programs over the time. Although the functionality and data base of these programs overlap each other, their data bases are totally separated. In addition there is no interaction between calculation and modelling (CAD) during product development, except by using interfaces (IGES, DXF, ...), that are mostly limited to geometric data.

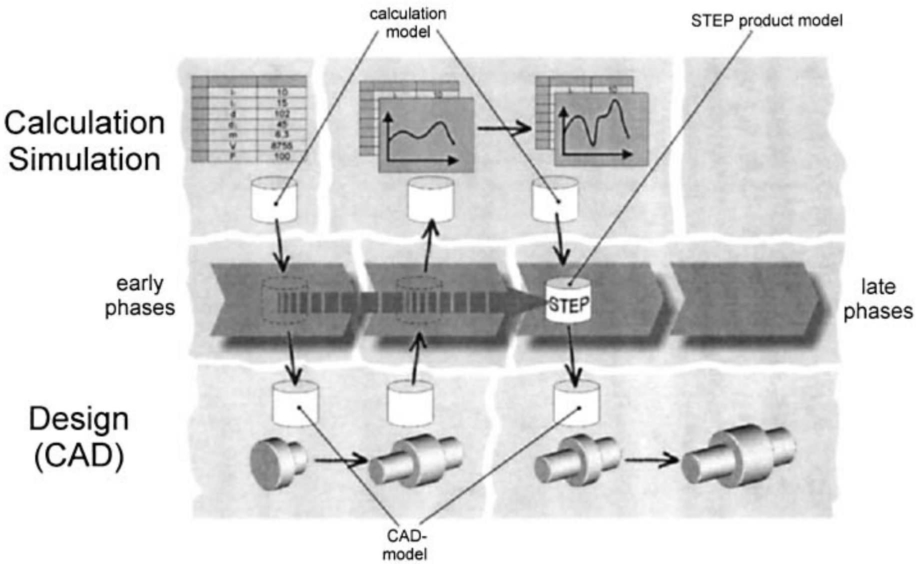


Figure 2. Workflow between design and calculation during product development

But this approach was no longer accepted in modern industry. To improve this a german research program "Innovative computer aided engineering: integration of modelling and calculation/simulation" has been founded in 1995 within the course of the DFG (german research foundation). More than 30 Universities discussed the combination of CAD and calculation/simulation in different projects [10]. This contribution shows results and perceptions of one project that is dedicated to describe a modern technique to develop a standard product.

The project meets the following objectives:

- Development of a digital workplace for the integrated design of gear units.
- Central STEP [8] product model in combination with PDM to enable a continuous data flow and to avoid data redundancy.

- Integration and bi-directional communication of calculation/simulation programs for the development of gear units with other applications, e.g. CAD, PDM.
- Complete representation of gear units as a virtual product.
- Transparent and easy integration of other applications and adaption to company-specific environment.
- Easy usage and comfortable GUI

3 Basic Research

The basic research and development of the workplace and its components (product model, PDM, GUI, ...) has been done since 1995 within the course of the DFG research program.

The focus of this contribution is to outline the work in the area of developing the product model and the data- and workflow management with PDM-functions. Further information about the architecture and the graphical user interface (GUI) of the digital workplace (figure 1) are described in references [3] [4] [5] [6].

3.1 Product Model

The product model is the unique data base during the whole product development process.

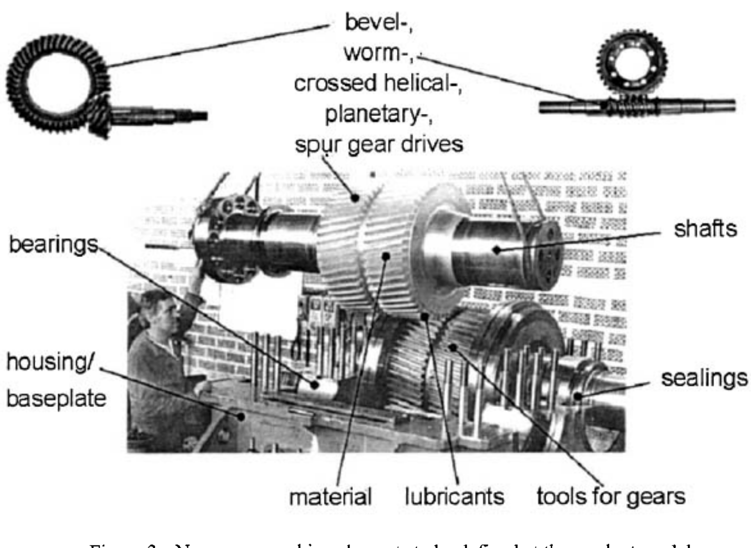


Figure 3. Necessary machine elements to be defined at the product model

This leads to the fact that all data from the development process are stored in this product model. Therefore the modelling mechanism must be equipped with efficient functions. The suitability of established product models has been checked for the modelling of gear units. As a result STEP [8] includes all necessary methods to define not only the geometry but also administrative and calculation data.

The product model includes products of manufacturers and suppliers of gear units. These products can be parts, assemblies and tools. All gear-drive elements that are taken into account of the product model are shown in figure 3.

The reproduction of all real gear elements to a digital product model is realized with a specification of STEP-AP214 [8]. Figure 4 shows all elements out of AP214 that can be associated with gears. But these elements are insufficient compared to the needs of gear units. The standardized AP214 product model was therefore enlarged with gear elements and properties. AP214 offers all necessary methods to define not only the geometry but all product data that are needed through the development process of gear units.

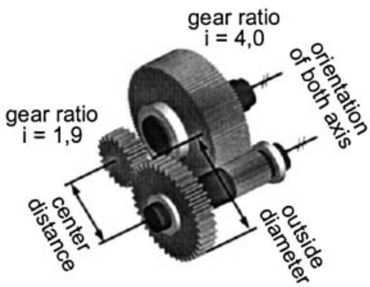


Figure 4. STEP AP214 gear entities

The following gear drive properties are specifized by the enlarged STEP product model and all of them can be assigned to the gear elements in figure 3:

- Environment-, operation- and application data.
- Geometry and tolerances, detailed geometry, 3D-topology (tooth-flank modifications, ...), geometry oriented data.
- Load capacity data (wear, pitting, tooth breakage, scuffing).
- Power loss and efficiency.
- Dynamic data.
- Load, torque, tensions, stresses, deformations.
- Technology data, quality, sets of characteristic curves.
- Tool- and manufacturing data.
- All STEP-AP214 properties [8].

In summary the STEP product model for gear units is an application specification of the standardized STEP Application Protocol 214 [8]. The installed mechanism is an object oriented structure of new gear entities in combination with their properties. The extension of the STEP standard is restricted to the level of the Application Reference Model (ARM). The new mapping for gear units leads to the same Application Interpreted Model (AIM) as standardized within STEP AP214. This assures a 100% compatibility to STEP AP214 during the exchange process. Not only CAD-systems but all other programs, such as PDM-systems can be integrated very easily due to the STEP conformance.

3.2 Workflow Modelling

The workflow during product development shown in figure 2 has to be managed with PDM functions. All these functions are embedded in the graphical user interface of the workplace and they have to meet the following requirements:

- Administration of all product data.
- Check mechanism of data import/export with access authority.
- Correlation of values of programs (calculation, simulation, CAD, ...) with product model based values.
- Administration of simulation/calculation programs and their variables.
- File management and design history archiving.

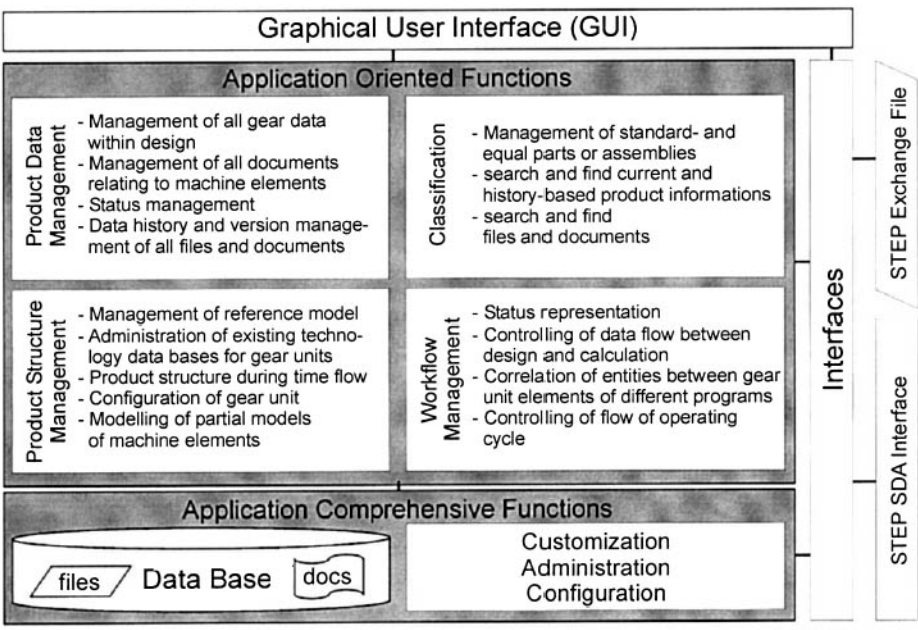


Figure 5. PDM functions used for gear unit development process

The task of the PDM-System is the workflow management in connection with the STEP data base. The implemented PDM architecture is shown in figure 5. It consists of base functions such as customization, administration and configuration and of application oriented functions that are needed to link all programs correctly and to assure a proper data flow. They are shown in figure 5. The realization of workflow and the usage of the PDM functions is described in the following chapter.

4 Implementation to Industry

Since 2000 the development of the product model and the adaption of the working place into industry is accompanied by the "Forschungsvereinigung Antriebstechnik" (FVA). The FVA is a german association of more than 130 companies in the field of transmission technology and automotive suppliers. The conformance to the STEP Standard and planned standardization is assured by the participation at DIN (German Industry Standard) workgroup NAM 430.4 "transfer and archiving of product defining data". The ProSTEP membership enables a coordination with other significant projects, e.g. the mechatronic project "MechaSTEP" [2].

On behalf of the FVA a lot of simulation programs (e.g. STplus, KNplus, RIKOR, WTplus, SNESYS) to investigate gear drives have been developed since more than 25 years. But these programs aren't linked either to each other nor with CAD-Systems. This was the starting point to use the STEP product model for gear units as a central data base at FVA companies.

Figure 6 illustrates the state of the art development process of gear units.

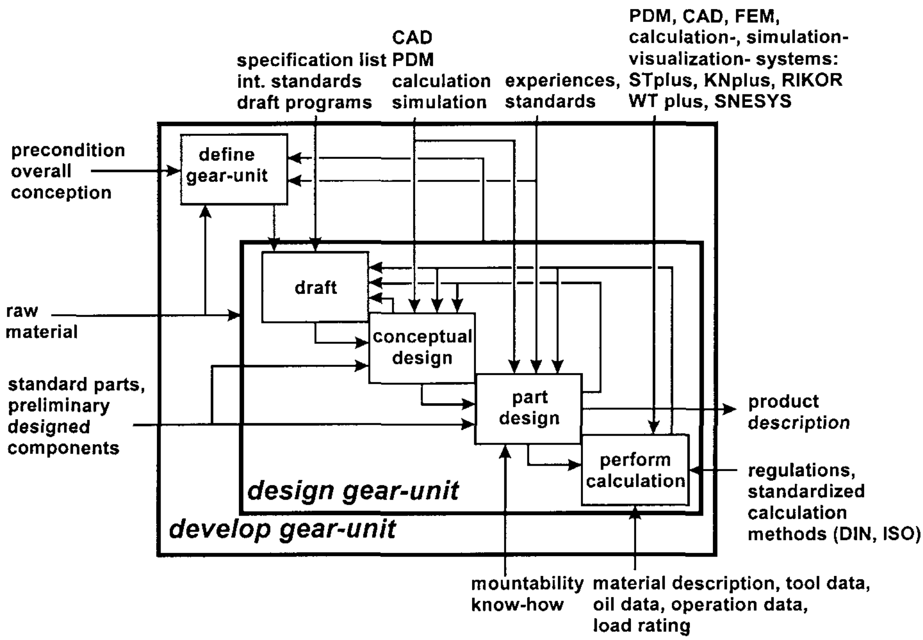


Figure 6. Development Process for gear-units (idealized Application Activity Model)

The idealized AAM (Application Activity Model) diagram shows how many systems and programs are participating in the development process. As a consequence a lot of data and files are to be managed and administrated. Therefore the usage of PDM functions is necessary. Figure 7 shows the coupling between CAD-System and calculation/simulation program as the most important task of the PDM system.

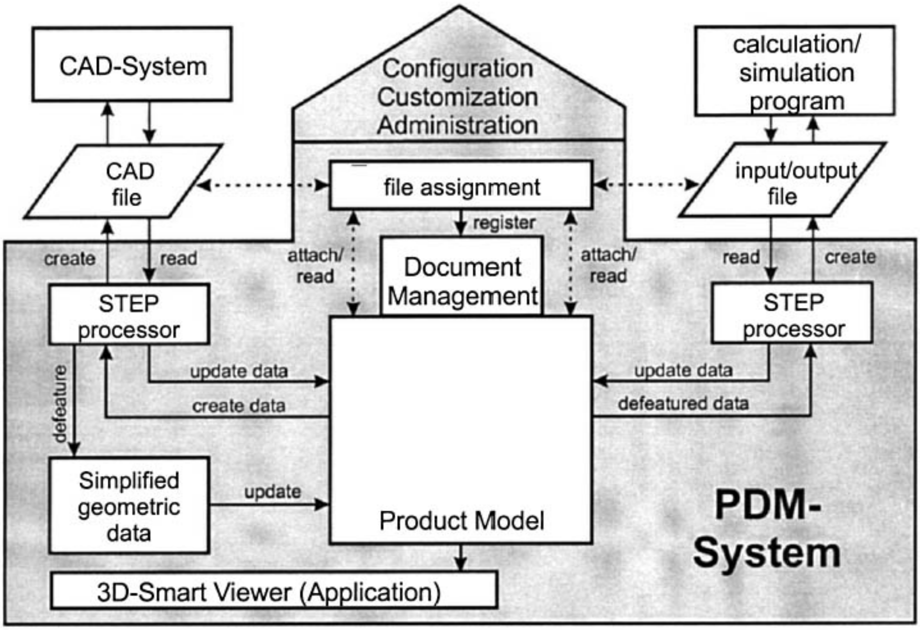


Figure 7. Link between modelling (CAD) and calculation/simulation

In support of the product development process the data flow between CAD and calculation/simulation must be enabled in both directions. This so-called bi-directional exchange process is passed through a lot of cycles during product development. This supports the importance of a proper data transfer, data storage and history recording. The realization is done with a product model to store all data in combination with a document management system to administrate all files. All exchange files are compatible with STEP AP214 and they are created and read by processors that are integrated in the PDM system.

5 Conclusions and View

The paper describes two important components of a digital workplace to design gear units. A central product model and PDM functions are responsible for a totally digital development process.

In conclusion the STEP product model is the data core of the workplace. In addition to the geometry the product model administrates all other data from calculation and simulation of gear units. All connections between the product model and the applications are managed and monitored by PDM functions. The implementation of the workflow model in the existing IT-periphery of FVA companies may require a reorganization of reasoning and the current processes because the product development strategy is significant different than before.

For the design of a product model a lot of possibilities exist. The usage of STEP assures a compatibility with all other STEP applications such as standard CAD systems. But even inbetween STEP it is necessary to differentiate the methods for the modelling of products. The chosen mechanism is a mapping on the level of the application reference model. This assures a 100% conformity to STEP AP214.

The product model for gear units is going to be documented as a Public Available Specification (PAS) at the DIN (German Industry Standard). It's publication is planned for 2001. The document is the first example how to specify STEP AP214 and use it for a special application. The documented mechanism can be used as a template for other STEP specification projects such as mechatronics or electronics.

In the future it is planned to expand and connect the product model to all data and processes which deal with the transmission technology. The digital workplace will then be the central station to design and administrate complex products.

References

[1] Anderl, R., Trippner, D.: Standard for the Exchange of Product Model Data. Teubner Verlag 2000

[2] Donges, C., Krastel, M.: Das MechaSTEP-Projekt – ein Statusbericht. ProSTEP Produktdatenjournal, No. 1, 2000, pp.13-14

[3] Höhn, B.-R., Steingröver, K., Dyla, A.: Rechnerbasierte Produktentwicklung-Das systemneutrale Produktmodell als Ergebnis von innovativem CAE. Journal Konstruktion, No. 1-2, 2000, pp. 36-40

[4] Höhn, B.-R., Steingröver, K., Dyla, A.: Systemneutrale CAE Methoden für die innovative Produktentwicklung. Journal CAD-CAM Report, No. 3, 2000, pp. 34-40

[5] Höhn, B.-R., Steingröver, K., Dyla, A.: Computer Aided Product Development. Proceedings Design2000, Dubrovnik, Croatia, 2000, pp. 307-312

[6] Höhn, B.-R., Steingröver, K., Dyla, A.: Product Model based Workplace for the Integrated Development of Gear Units. Proceedings ProSTEP Science Days 2000, Stuttgart, 13.-14.09.2000, pp. 64-73

[7] Grabowski, H., Anderl, R., Polly, A.: Integriertes Produktmodell, Beuth Verlag 1993

[8] ISO 10303-214: Core Data for Automotive Mechanical Design Processes, 2000

[9] Spur, G.; Krause, F.L.: Das virtuelle Produkt: Management der CAD-Technik. Carl-Hanser Verlag 1997

[10] VDI-Berichte 1487: Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung – Potentiale und Erfahrungen. VDI Verlag 1999

Dipl.-Ing. A. Dyla, Prof. Dr.-Ing. B.-R. Höhn, Dr.-Ing. K. Steingröver
Gear Research Centre (FZG), FVA-EDV-Centre, Technical University Munich,
Boltzmannstr. 15, 85747 Garching, Germany, Phone: +49 (0)89 289 15821,
Fax: +49 (0)89 289 15808, E-Mail: fzg@fzg.mw.tum.de, Internet: http://www.fzg.mw.tum.de

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21-23, 2001

PROVIDING SUPPLIER-COMPONENTS DATA AS PDM-OBJECTS IN WEB-BASED SYSTEMS

D Nötzke and H Birkhofer

Keywords: hypermedia and multimedia, classification and retrieval, information representation, product data management, web-based systems

1 Introduction

Supplier components play an important role in product design. So supplier components are often used to fulfil not only recurrent secondary functions, but also main functions. As a result they are of particular importance in the stage of making a layout. In this case making a layout is seen as the entirety of activities to establish all of the construction elements and their connections (structure). To orientate oneself to the sequence of operations according to VDI 2221, the use of supplier components is to be considered, particularly in steps 5.3 (rough design of remaining main function carriers) and 6.1 (rough design of remaining secondary function carriers) [1].

If the designer wants to use supplier components, he has to weigh benefits to efforts. The efforts are especially important because the designer is in the face of the vast information abundance by searching for supplier components. In this situation designers need support to manage the considerable information. The availability of all the pre-existing components in digital libraries, and their exchange or remote access through the network would drastically increase the efficiency and the quality of the design process of products using supplier components [2]. This shows that in the field of product innovation there is a growing importance of process innovation. Modern information and communication technologies support this trend.

The project POINT was presented on ICED 99. The objective of this project, which was promoted by the German Ministry of Economy and Technology, was to develop a procedure to realise an "associative integration" of online-catalogue systems provided by suppliers into virtual marketplaces [3]. POINT II takes advantage of the results of POINT to integrate components to customers' component database. For this purpose the use of standards is an important demand.

2 Standards

The ISO 13584 series provide a representation of parts library information. The standard provides a generalised structure for a parts library system and does not define a fully detailed implementable parts library system. ISO 13584 does not specify content of a supplier library. The content of a supplier library is the responsibility of the library data supplier. The library management system used in the implementation of the structure defined in ISO 13584, and any interface between this system and a user of the system is the responsibility of the library management system vendor and is not specified in ISO 13584 [4].

An important characteristic for integration in manufacturing and engineering is that the standard ISO 13584 is fully consistent with the ISO 10303 "Standard for the Exchange of Product (STEP)". It extends the STEP capabilities, mainly oriented towards explicit modelling of one particular product, to support implicit and global modelling of classes of similar products (part families) [5].

3 Data model and implementation

In ISO 13584 supplier components are divided into parts and assemblies. Following this model we distinguished components in size ranges and modular systems. Therefore it was necessary to build models of the various components. For each there is a separate model. Modular systems extend size ranges by a special configuration method. Up to now it has only been possible to describe size ranges to search for them by properties. In POINT II itself the model for modular systems, has been developed to make it possible to describe them by properties, too. Because of the complexity of modular systems, first we have to speak the same language. Figure 1 try to make the terms clear by using the modular system Zuko, a simple „robot" assembled by supplier components.

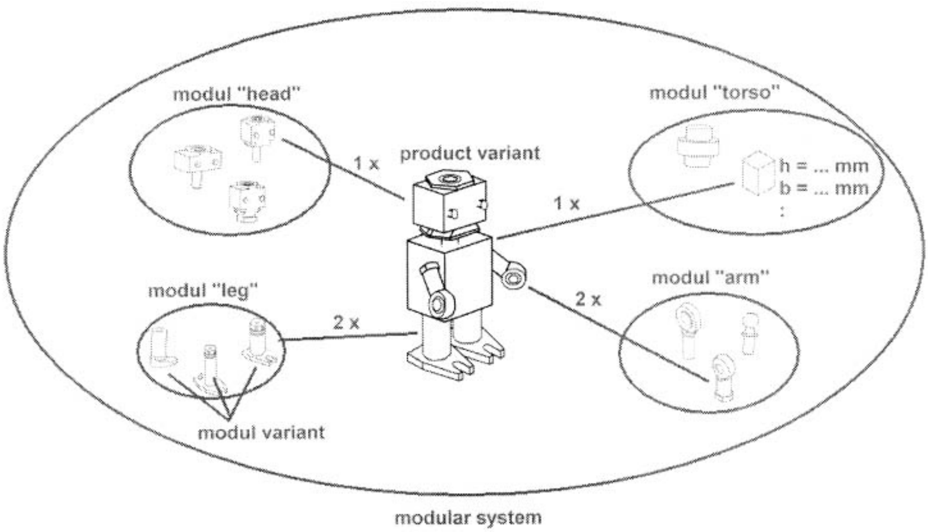


Figure 1: common expressions

In this case the modular system Zuko is made up of four modules: Head, body, leg and arm with 2-3 module variants each. With the help of four structure relations (intimated by the straight lines between the several modules and the product variant with the number of used modules each) and selection of one module variant each, a product variant can be configured. Nevertheless not all module variants have to be compatible with all the variants of another module. Such incompatibilities are often schematize in so-called compatibility matrizes. On the top of this is the fact that modules can be used in several different modular systems. People say it is a horizontally intertwined modular system. By contrast vertically intertwined modular systems are modular systems in modular systems. This means a modul is carried out as a modular system again.

The expressions mentioned cause us to suspect which complexity will have to be shown by the data model. Figure 2 shows the first approach of modelling modular systems in EXPRESS-G [6]. It also shows the differences to the model of size ranges.

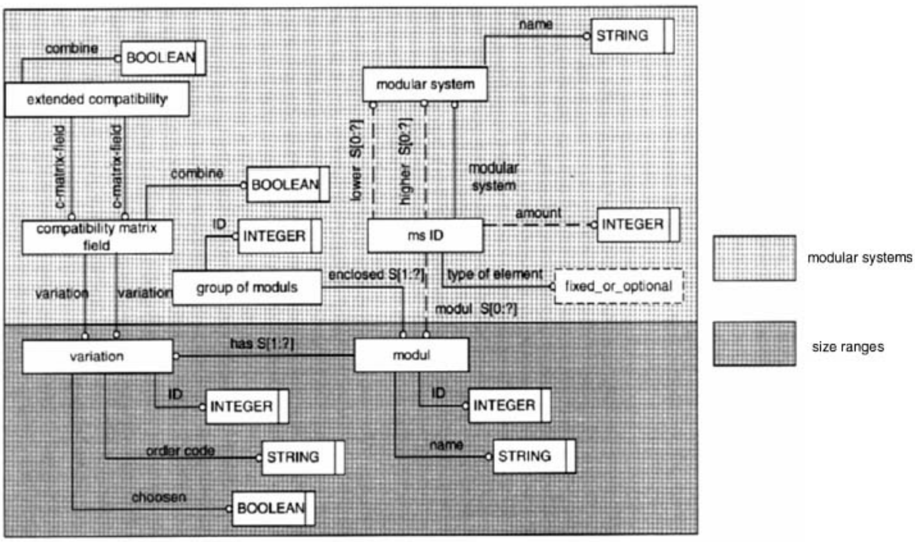
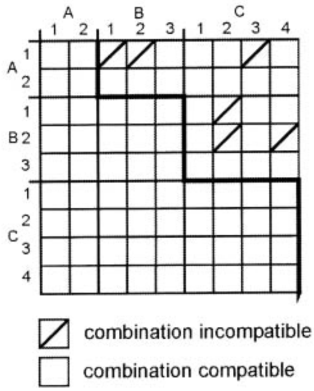


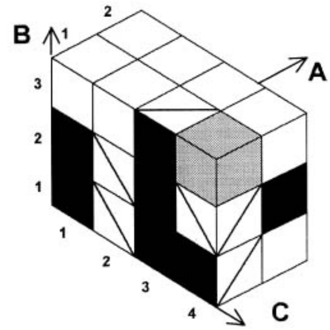
Figure 2: first approach of a supplier component data model

3.1 Compatibility of variations

The next paragraph shows a method to map compatibility in n-dimensional vectors. Up to this point, modules have been arranged in paires to compare them with each other. The so-called compatibility matrix, a two-dimensional ordering diagram, make it possible to compare all modules of a modular system in a methodical way[7].



a) compatibility matrix



b) compatibility cuboid

Figure 3: compatibility styles[7]

The disadvantage of a compatibility matrix is that only a comparison of pairs is possible. Another pattern to describe compatibility is indispensable.

The first approach is to transfer the model of the matrix to a 3-dimensional cuboid. Figure 3 shows three modules (A,B,C) and their variations (A1,A2,B1,B2,B3,C1,C2,C3,C4) in such a cuboid. We map the three submatrices to the surface of the cuboid. The incompatibility of the variations A1 and C3, i.e. causes the combinations $A1 \wedge B3 \wedge C3$, $A1 \wedge B2 \wedge C3$ and $A1 \wedge B1 \wedge C3$ to be incompatible. Therefore, because of this incompatibility, three of the 2*3*4=24 potential product variations cannot be built. To describe these relations mathematically we developed a vectored representation. The above shown incompatibility is represented by the vector:

$$\begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}$$

All incompatibilities resulting from this vector are described by the line equation:

$$\vec{x} = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} + \lambda \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

Representing by vectors makes it possible to build n-dimensional incompatibilities. A 3-dimensional example is the vector:

$$\begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}$$

It is shown in Figure 3b) by the grey cube. Coordinates unequal zero determines the degree of incompatibility.

So, the benefits of a vectorial representation are:

- Simple computer-internal representation

- Possibility to represent n-dimensional incompatibilities
- Possibility to exclude single product variants
- Oriented towards ISO 13584 level 3 parts concept

However, for the input of incompatibilities, matrices are a useful tool. The actual status of implementation will show this.

3.2 Implementation

For the *suppliers* we designed a tool to describe their components by characteristics. Both size ranges and modular systems are ordered in a product classification and are described by their properties. In this case a property is built by characteristic plus value. Birkhofer decided between structure properties to describe the internal architecture of an object, and extensive properties with an effect directed outwardly[7]. Properties integrated in module representation contain both types of properties. By the process of informing oneself, which is the center of our interest, somebody mainly accesses extensive properties[8]. The benefit of description by properties is that suppliers do not have to provide the complete product model data. Therefore they have the security that they will not lose their product know-how to competitors. On the other hand this information will be sufficient for customers.

In the software system we developed, the so-called property matrix serves to type in properties (Figure 4). In property matrices the user is able to allocate values once defined with the appropriate characteristics to size ranges, respectively modules, by marking these with a cross. As a result there is a considerable lower effort of data input.

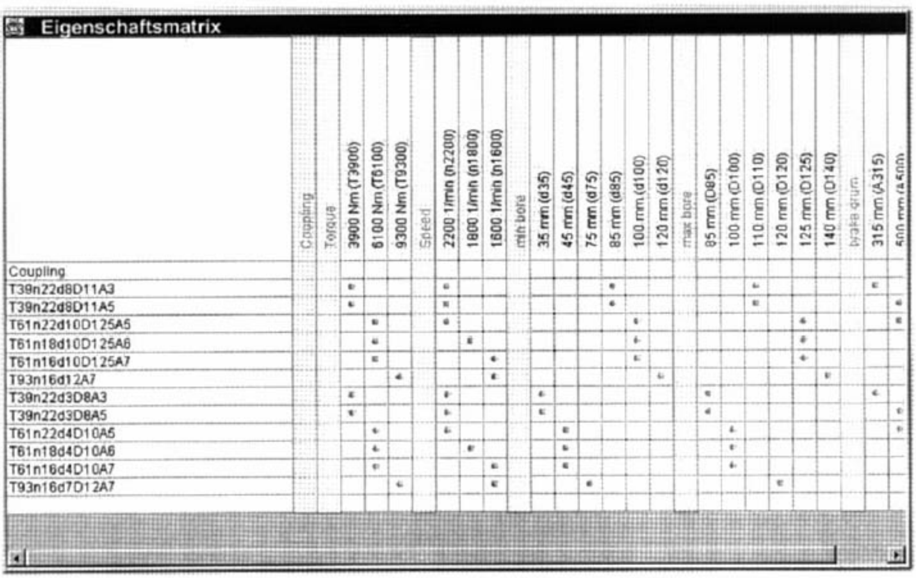


Figure 4: property matrix

If the supplier component is a modular system, the compatibility matrix as a simple instrument for data input is used in addition. The entries made in the compatibility matrix are transformed in the vectored representation mentioned above. N-dimensional incompatibilities are directly typed in as vectors.

Components described by suppliers in this way are able to be exported to a special description language. The description language to supply components on the internet is specified in XML.

The next step is to use the supplier information to build simplified product models. Therefore suppliers have to provide not only article characteristics but geometric and structural characteristics. The customer uses these characteristics and a generic model which has to be carried only once. After that only the characteristics will be updated. In customers PDMS these two components merge to one PDM object.

4 PDM-integration

Because of integration and handling of supplier components as PDM-objects respectively customers are able to make use of functions of the PDM-System (PDMS). These functions are i.e., management of versions, authorization or workflow. Integration is based on ISO 13584 mentioned in chapter two. Besides, integration increased the efficiency of using supplier information. Above all this is of importance because among technical criteria there are also criteria of the purchasing department such as availability or preferential suppliers. By using supplier components as PDM-objects, customers are able to integrate these components in their internal workflow and processes.

Among the development of a description model there are also results in the second part, the integration of supplier components as PDM objects. In several workshops and one survey the demands of *customers* are gathered. The key demands are listed below:

- type of interconnection (use of the system without considering the kind of internet interconnection; consideration of internet bandwidth)
- broken network interconnection (local working has to be possible)
- transparency of place (irrelevance where object data is physically stored)
- object identity (a global object identification is necessary)
- administrative sovereignty (the administrative sovereignty of master objects remains by the supplier; the administrative sovereignty of copies remain by the customers)
- update (customers have to be informed about updates)

Based on these demands catalogue information has to be integrated into customers tabular layout of characteristics. In this connection data remains by the supplier but the customer gets a copy. Thereby it is possible to provide supplier components to the designers in their own familiar environment. A global search is possible. In case of standard parts designers are able to include supplier specific intermediate sizes.

5 Conclusion

POINT II set the goal for making available supplier components data directly as Product Data Management (PDM) objects independent of a specific sector or particular kind of components although the focus is especially on modular systems.

For this we designed a data model oriented by ISO 13584, to support designers in special steps of design process by the inclusion of supplier components. Thereby the focus is especially on the data model of structured modular systems, that extends the data model of size ranges designed in POINT I. XML as a further standard to structure, and exchange data was chosen to be compatible with other projects on this area. In addition with the help of X3D, a 3D-format based on XML and VRML there is a chance to exchange parametric geometric in the future.

References

[1] H. Birkhofer: Produktentwicklung, Umdruck zur Vorlesung Wintersemester 1997/98, TU Darmstadt 1997

[2] E. Sardet, G. Pierra, Y. Ait-Ameur: Formal Specification, Modelling and Exchange of Classes of Components according to Plib. - A case study, Proceedings of the International Symposium on Global Engineering Networking Antwerp, Belgium, HNI-Verlagsschriftenreihe , Bd.21, Paderborn 1997

[3] How to use global supplier information efficiently – product modelling and associative integration in virtual market places, Proceedings of the ICED 99 Conference WDK 26 Vol. 2, p697, Munich 1999

[4] ISO-13584-1; Parts Library, Part 1 Title: Overview and Fundamental Principles; ISO/IEC Schweiz, 1997

[5] ESPRIT 8984: PLUS - Parts Library Usage and Supply; Summary of the PLUS project results; http://www.fast.co.at/Research/plus.htm

[6] ISO-10303-x; Product Data Representation and Exchange, Part 11: Description methods: EXPRESS language reference manual; ISO/IEC Schweiz, 1993-95

[7] H. Birkhofer: Analyse und Synthese der Funktionen technischer Produkte, Dissertation TU Braunschweig, VDI-Verlag, Düsseldorf 1980

[8] K. Büttner: Rechnerunterstütztes Konfigurieren von Baukastenprodukten, Dissertation TU Darmstadt, VDI-Verlag, Düsseldorf 1997

Dipl.-Ing. Dirk Nötzke
TU Darmstadt
Department for Machine Elements and Engineering Design
Magdalenenstraße 4
64289 Darmstadt
Germany
Phone: +49 6151 / 16-2155
Fax: +49 6151 / 16-3355
E-mail: noetzke@muk.tu-darmstadt.de

This page intentionally left blank

INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 01 GLASGOW, AUGUST 21 23, 2001

TOWARDS 'DESIGN DEFINITION MANAGEMENT'

S M Barker

Keywords: *Product Data Management, Digital Mock-Up, Feature Based Design, STEP.*

1. Introduction

This paper forms part of a larger, on-going work on creating an integrated design environment in the aerospace industry. A new aircraft design may take some 20 years to develop, involve several thousand staff in the prime contractor alone, and involve the use of several hundred software applications.

Designing an aircraft is not a static problem, for which a well defined set of inputs lead to a finished product. Even over the design lifecycle, major changes in requirements, technologies and supply chain can be expected. Such changes impact cost and schedule, as well as fit, form and function, and must be controlled through configuration management (CM) mechanisms.

However, current CM standards, such as ISO 10007 [1] or EIA 649 [2] are driven very much from the customer viewpoint, and look only at controlling the standard of the delivered product. They fail to even mention the primary role of CM, which is to control of the dynamics of the design process. Furthermore, the control model they imply is obsolete, and fails to account for developments in concurrent engineering and Integrated Product Teams (IPT).

Design Definition Management (DDM) is an extension of the CM control model. It aims to support the IPT in managing the growth in maturity of a particular product. The object of this paper is to define the role of DDM and to differentiate it from that of the other control processes.

2. Control Context (Figure 1)

The approach arose from the AIT cluster of Esprit projects, where there was a need to differentiate the problem domain covered by the Digital Mock-Up project cluster from that of Configuration Management (later the ADCOMS project) [1]. The "Four Level Architecture" was developed further within BAE SYSTEMS to describe and analyse the way product development is controlled.

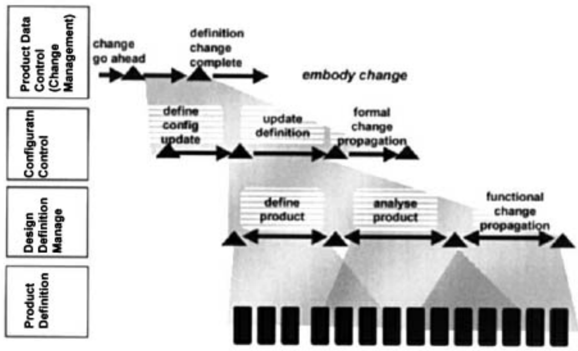


Figure 1: Control Context

It is based on a normative breakdown of the engineering design process into four layers of control:

- Project/Change control - the top level of processes considered by the Four Level Architecture (4LA). It starts with customer requirements and ends when these are embodied in a delivered product, although the 4LA is only concerned with that part of the process that creates or changes the product design.

- Configuration Management - a set of processes spawned by Project Management, which ensures the elements of the product design are fully identified, that their status is known and that they are actually used in the right products.

- Design Definition Management - processes spawned by a configuration change form a newly emerging area which supports the Integrated Product Team (IPT) in managing the growth in maturity of a particular part of a product design.

- Design - activities of individuals in creating the design data.

Note: creating a new design is treated as the same *control* problem as altering an existing design (as a change from nothing to something).

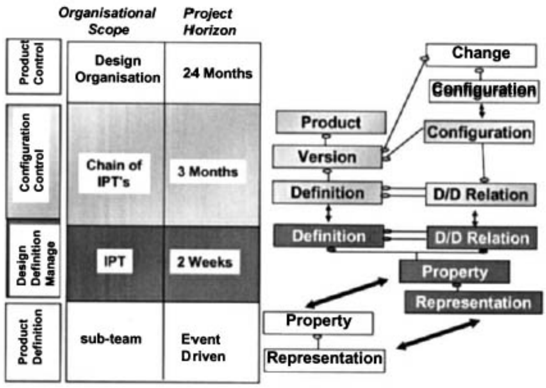


Figure 2: Integrated Approach in Four Layers

3. Integrated Approach (Figure 2)

Although described above in process terms, the architecture integrates process, organisational, project and product views. This is based on models of decision making from the University of Bordeaux's GEM methodology [2], and, in effect, treating the design organisation as a design factory.

At the DDM level, the organisation is a single IPT - a team of up to 30 people from all disciplines - who are allocated specific design problems with a duration, say, of two to three months. Typically this will be the development of both the physical and manufacturing view of some subsystem of the product. Project planning problems include the availability of team members (holidays, other projects, etc.), finding that a proposed solution will not work, and co-ordinating with other IPT's

The 4LA approach leads to a corresponding hierarchical architecture for a federated PDM environment. This further maps across to the STEP data model [3], with each layer being competent to create and control particular entities in the model. This federated PDM environment will be needed to cope with integrated design/analysis systems, with software tools based around databases and with structured file applications such as the Web.

The 4LA approach has already been used to analyse and develop the Product Data Management (PDM) strategy of a major project, and is being used to help structure the main design processes for military aircraft. It is also now being used to develop integration links between engineering analysis domains.

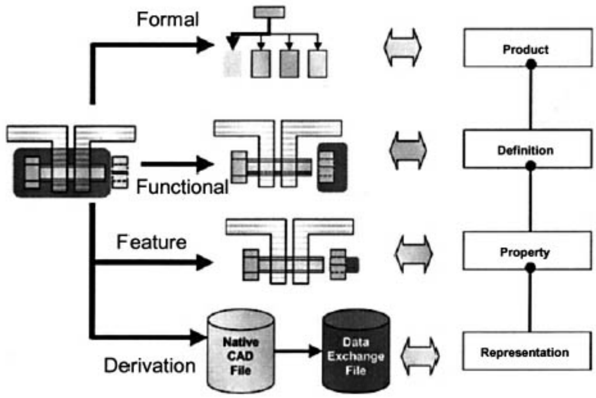


Figure 3: Types of Change Propagation Problems

4. Types of Change Propagation Problems (Figure 3)

The fundamental problem of design is to propagate a change from its source to a complete design. The first step is to classify the change propagation problems and allocate them to particular levels of the 4LA:

- Formal change propagation acts at the Configuration Control level to follow the effects of reissue of configuration items - if a bolt is redesigned, then the assembly that uses it needs to be reissued.

- Functional change propagation identifies that a part is functionally related to another - such as mating parts- but leaves the engineer to assess and effect any necessary change. E.g. if the material of a bolt is changed, how does this affect the nut? Since this follows relations between different items in a product structure and between product structures (i.e. it follows definition/definition relations), it is a DDM problem

- Feature based propagation allows one design to directly update another though design knowledge of the feature. E.g. changing a bolt thread changes the thread of the nut. Because it uses design knowledge, this is a design not a design management problem.

Direct derivation converts one representation to another, such as a CAD model to a data exchange file. This is a physical data management problem, but is grouped with DDM domain by virtue of DDM's role in keeping data consistent across a team.

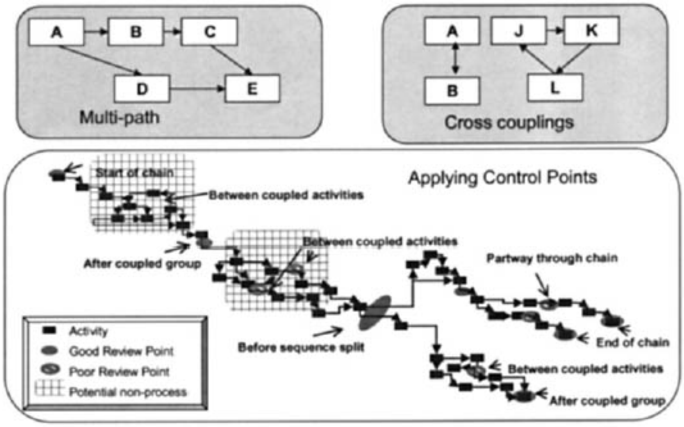


Figure 4: Problems in Functional Change Propagation

5. Problems in Functional Change Propagation (Figure 4)

In the restricted domain of functional change propagation (the DDM problem) there are some significant open issues, including:

- Multi-path and cross-coupling between design activities
- Relations between multiple product structures
- Design decision trees and recording design rationale

The multi-path problem occurs when a change triggers two or more separate chains of knock-on effects, which intersect again at some common part. The problem is to ensure that the part definition is updated only once, taking into account all the knock-on effects, rather than redesigning as each knock-on effect reaches the part.

A variant on this problem is caused by cross coupling. Cross coupling occurs when, say, the output of design activity A is input to activity B, and the output of B is input to A. There are a number of ways of converting a chain of cross coupled activities into process (a process sequences activities in time). These include spiral design, prototyping and "non-processes", where a non-process uses the team organisation to negotiate the design between the activities, rather than defining a sequence for them.

The "Design of Design Processes" is a topic closely related to DDM. For example, reviews - the control points of a DDM process - should not be sited between cross-coupled activities, since cross couple activities tend to iterate, and in consequence the reviews will need repeating.

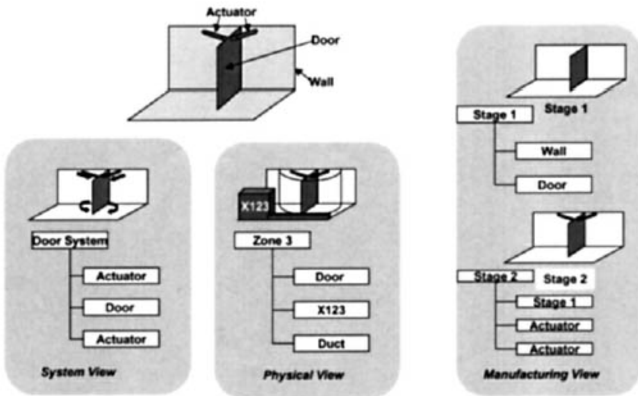


Figure 5: Multiple Product Structures

6. Multiple Product Structures (Figure 5)

A product structure is a discipline specific breakdown of the product, based on creating a comprehensible hierarchy of emergent behaviours. For example, a door system may have two identical actuators to open and close the door. "Open" and "close" are properties of the system as a whole - one cannot look at an actuator in isolation and ask "does this open or close the door?"

The behaviours of interest to a *systems engineer* are "open" and "close". However, the *layout designer* must ensure than the space needed by each function is consistent with all the other functions - not only the physical volume taken up by a part, but also the volume used by the door mechanism, the access paths for service personnel, etc. At best there will be a correspondence at the component level but intermediate (system) nodes of the product structure are unlikely to correspond.

However, the intermediate nodes are the points at which configuration control is applied, that is, they are the starting point for any change propagation. (Components change as a consequence of changing the system). It would be possible to build direct links between the nodes in one structure and the nodes in another. However, the cost of building and maintaining them in the face of change may be excessive. It is thought that some rule based approach to link inheritance would help.

Given the links exist, there will then be a problem of following them without getting lost in loops, or of following every possible link and creating many false alerts. A rule based approach will be needed here.

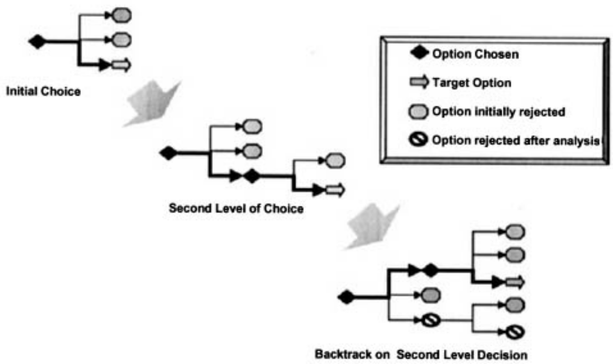


Figure 6: Design Rationale

7. Design Rationale (Figure 6)

Design decisions are always made for a series of reasons. The reasons may include past experience or feedback from a downstream process, e.g. Supportability analysis shows that repairing this part is not practical in the field.

Design decision trees follow the development of a product in terms of the design commitments at each stage of the process. As work proceeds downstream, it may show that upstream commitments are not feasible. This leads to backtracking up the decision tree and creating a new branch with different commitments.

The tree structure provides a home for recording design commitments and their rationales. It also helps expose the set of decisions consequent on each decision, and hence the impact of changing that decision. Viewing rework as backtracking - rather than feedback in a process - should also give a more precise view of the impact of the revision and hence better project control.

It is speculated that design decision trees may be used as a way of moderating the behaviour of systems which trace functional relations in and between product structures. For example, "fillet radius set to 3mm for standard cutter" is a clear indication not only that there is a functional relationship between the part design and the manufacturing process plan, but also that a decision was made based on the manufacturing environment. Following the link to manufacturing process planning would therefore be a priority in looking at the effects of change.

8. Conclusions

- The Four Level Architecture provides a normative analysis of the design process into four levels of control.
- The process view has corresponding organisation, project and product views and an associated software architecture.
- Design Definition Management is one of the levels of the 4LA, corresponding an Integrated Product Team managing design maturity in a Concurrent Engineering environment.
- DDM covers the problems of Functional Change Propagation, particularly:
- Design activity sequencing, include coupled activities
- Relating different product structures
- Capture of design rationale

DDM is an emerging - but as yet ill defined - control level in the overall design process. The Challenge is to create a *practical* system.

References

[1] ISO 10007 "Quality Management - Guidelines for Configuration Management".

[2] "National Consensus Standard for Configuration Management", EIA/IS 649, Engineering Industries Association, Aug 95.

[3] Barker S. et al,"Harmonisation of DATA ORGANISATION in DMU and in PCM" ESPRIT Project 7704 Deliverable D.6.3.1.1.1.3, Dec 95

[4] Doumeingts, G., Vallespir, B., Zanettin, M., & Chen, D. "GIM, GRAI Integrated Methodology - A methodology for Designing CIM systems" LAP/GRAI University of Bordeaux I, France

[5] ISO 10303 "Industrial Automation Systems and Integration:Product Data Representation and Exchange" (aka STEP - in many parts)

Author: S. M. Barker
BAE SYSTEMS, Advanced Technology Centres - Sowerby, P.O. Box 5, FPC 267 Filton, Bristol BS34 7QW. Sean.barker@BAESYSTEMS.COM
Tel: +44 (0)117 936 3375 Fax: +44 (0)117 936 3733

Authors' Index

A

Anderl, R 421–428
Andreasen, M M 3–10
Araujo, C S 573–580
Arcidiacono, G 45–52
Armeanu, A 93–100
Aßmann, G 493–500
Axtner, H 701–708

B

Barker, S M 724–732
Baumgärtner, C 189–196
Baxter, D H 605–612
Bender, B 173–180
Bender, B 197–204
Bercsey, T 21–28
Birkhofer, H 717–723
Blessing, L T M 173–180, 181–188, 189–196, 197–204, 557–564
Bligh, T P 237–244
Bojčetić, N 581–588
Bongulielmi, L 653–660
Borg, J C 445–452
Bracewell, R H 181–188, 557–564
Braham, D 117–124
Brix, T 613–620
Brüderlin, B 613–620
Bryant, C R 501–508

C

Camilleri, K P 445–452
Campatelli, G 45–52
Cantamessa, M 29–36
Chakrabarti, A 237–244, 379–386
Chan, B 485–492
Chan, X 485–492
Chang, T K K 637–644
Chase, S C 621–628
Clarkson, P J 141–148, 181–188, 371–378, 557–564
Cury, J E R 517–524

D

de Negri, V J 517–524
Deasley, P 701–708
Dekovic, D 275–282
Dew, P 669–676
Dierssen, S 653–660
Döring, U 613–620
Dorst, K 205–212, 245–252
Duffy, A H B 213–220, 315–322, 453–460
Duhovnik, J 53–60
Dyla, A 709–716

E

Earl, C 149–156
Eckert, C M 149–156, 371–378
Eder, W E 69–76
Ehbets Müller, R 395–402
Eppinger, S D 283–290
Erden, A 291–298
Erden, Z 291–298
Eynard, B 509–516

F

Farrugia, P 445–452
Fauroux, J-C 549–556
Fauvel, O R 261–266
Field, B 109–116
Finger, S 485–492
Fiorentino, M 645–652
Fitchie, M R 669–676
Flath, M 541–548

G

Gabriele, G A 605–612
Gausemeier, J 541–548
Gero, J S 339–346
Gerst, M 371–378
Giannini, F 445–452
Grote, K-H 661–668
Gruenenfelder, M 133–140
Gu, P 525–532
Güroglu, S 291–298

H

Hacker, W 173–180, 197–204, 461–468
Hansen, C T 101–108
Hendriks, D J H C 205–212
Hofer, A P 133–140
Höger, K 403–410
Höhn, B-R 709–716
Höhne, G 613–620
Holmqvist, T K P 413–420
Horváth, I 13–20, 469–476, 565–572
Hubka, V 69–76

I

Igoshi, M 533–540
Ito, Y 61–68

J

Jandeleit, M 589–596
Jansson, J 565–572
Jarrett, J 141–148
Johansson, P 685–692
Johnson, J 149–156
Juster, N P 669–676

K

Kammerer, U 173–180, 197–204
Käosaar, J 299–306
Kaymaz, I 629–636
Kennel, T 387–394
Kimura, I 661–668
Koch, M 437–444
Krastel, M 421–428
Kuczogi, G 565–572
Kumar, B 157–164
Kunz, A M 387–394, 693–700
Kurfman, M A 501–508

L

Labrousse, M 597–604
Lafon, P 509–516
Lan, R 485–492
Langdon, P M 181–188, 557–564
Lange, D 701–708
Lange, M W 77–84
Lauche, K 267–274, 395–402, 403–410
Laurikkala, H 165–172
Lebbon, C S 221–228
Lee, B S 453–460
Lee, K 629–636
Leonardt, U 653–660
Lewis, W 37–44, 109–116
Liew, P 621–628
Lim, S 453–460
Lindemann, U ..... 347–354, 371–378, ... 461–468, 477–484
Lipson, H 45–52
Lordan, M M 355–362

M

Macdonald, A S 221–228
Mack, P 21–28
Marjanović, D ..... 275–282, 429–436, ... 581–588
Marxt, C 323–330
Maxfield, J 669–676
Mbiti, K 395–402
McAdams, D A 501–508
McMahon, C A 629–636
Meerkamm, H 437–444
Meier, M 653–660, 693–700
Möhringer, S 541–548
Monno, G 645–652
Mulet, E 363–370
Müller, S 387–394
Murakami, T 229–236
Muscat, J 445–452

N

Nidamarthi, S 237–244
Nomaguchi, Y 229–236
Nötzke, D 717–723

O

O'Donnell, F J 315–322
Opiyo, E Z 469–476
Opran, C 93–100
Ottosson, S 331–338

P

Pache, M 461–468
Pagenstert, K 701–708
Pajarre, E 165–172
Pavkovic, N 275–282
Pavlic, D 429–436
Pietzcker, F P 173–180, 197–204
Pulm, U 477–484
Puustinen, E 165–172

R

Reich, Y 117–124
Römer, A 461–468
Rusák, Z 565–572

S

Sakita, K 533–540
Salminen, V K 125–132, 283–290
Samuel, A 37–44
Santos, E A P 517–524
Sartor, M 549–556
Schachinger, P 677–684
Shea, K 181–188
Sim, S K 213–220
Sinha, S 157–164
Steingröver, K 709–716

Stetter, R 493–500
Stone, R B 501–508
Storga, M 429–436
Strohmeier, K 589–596

T

Takeda, H 229–236, 307–314
Tang, V 125–132
Tanskanen, K 165–172
Taylor, S 669–676
Thompson, G 355–362
Thomson, A I 157–164
Tomiyama, T 229–236, 307–314
Tsumaya, A 229–236

U

Uva, A E 645–652

V

Vajna, S 21–28, 589–596
Valkenburg, R C 253–260
Vergeest, J S M 469–476, 565–572
Veron, P 597–604
Vidal, R 363–370
Viertlböck, M 493–500

W

Wahl, J-C 549–556
Wallace, K M 557–564
Wang, C C L 637–644
Warell, A V 85–92
Warisawa, S 61–68
Weir, J 109–116
Willings, R M 61–68
Winkelman, P M 261–266

Y

Yoshioka, M 229–236, 307–314
Yuen, M M F 637–644

Z

Žavbi, R 53–60
Zeng, Y 525–532
Žeželj, D 581–588
Zhao, J 669–676
Zirkel, M 589–596