Engineering Design Synthesis

Springer-Verlag London Ltd. http://www.springer.de/phys/

Amaresh Chakrabarti (Ed)

Engineering Design Synthesis

Understanding, Approaches and Tools

Springer

Amaresh Chakrabarti
Associate Professor
Centre for Product Design and Manufacturing (CPDM)
Indian Institute of Science
Bangalore 560012
Karnataka, India

British Library Cataloguing in Publication Data
Engineering design synthesis: understanding, approaches and tools
1. Engineering design 2. Engineering design – Data processing 3. Engineering design – Computer programs
I. Chakrabarti, Amaresh
620′.0042

ISBN 978-1-84996-876-8 ISBN 978-1-4471-3717-7 (eBook)
DOI 10.1007/978-1-4471-3717-7

Library of Congress Cataloging-in-Publication Data
Engineering design synthesis: understanding, approaches and tools / Amaresh Chakrabarti (ed).
p. cm.
Includes bibliographical references and index.

1. Engineering design. I. Chakrabarti, Amaresh.
TA174.E545 2001
620′.0042 – dc21 2001038410

Originally published by Springer-Verlag London Limited in 2002.
Softcover reprint of the hardcover 1st edition 2002

Typesetting: Best-set Typesetter Ltd., Hong Kong

69/3830-543210 Printed on acid-free paper SPIN 10835033

Preface

This book is an attempt to bring together some of the most influential pieces of research that collectively underpin today's understanding of what constitutes and contributes to design synthesis, and the approaches and tools for supporting this important activity.

The book has three parts. Part 1 – Understanding – is intended to provide an overview of some of the major findings as to what constitutes design synthesis, and some of its major influencing factors. Part 2 – Approaches – provides descriptions of some of the major prescriptive approaches to design synthesis that together influenced many of the computational tools described in the final part. Part 3 – Tools – is a selection of the diverse range of computational approaches being developed to support synthesis in the major strands of synthesis research – composition, retrieval, adaptation and change.

In addition, the book contains an editorial introduction to the chapters and the broader context of research it represents, and a supplementary bibliography to help locate this broader expanse of work. With the wide variety of methods and tools covered, this book is intended primarily for graduate students and researchers in product design and development; but it will also be beneficial for educators and practitioners of engineering design, for whom it should act as a valuable sourcebook of ideas for teaching or enhancing design creativity.

The general idea of the need to bring together works of research in design synthesis, both manual and computational, had its seeds in the feeling that grew in me in the early 1990s while participating in design conferences. It seemed that conferences that were largely design(er)-centred had a great deal in common, in the goals pursued and means used, with those with a strong computational flavour; yet, there was little information exchange or synergy between the two. A synthesis of ideas developed in these two research communities seemed necessary. This culminated in an earlier attempt at bringing together functional representation and reasoning research in the form of an *Artificial Intelligence in Engineering Design, Analysis, and Manufacturing* special issue in 1996. Taking on a project of this breadth, however, required a closer feasibility study, and working out the modalities. A workshop in design synthesis in Cambridge in 1998, in which about 30 researchers from around the world participated, provided this, and I am grateful to Lucienne Blessing and Tetsuo Tomiyama who helped make that possible.

This book would not be possible without the many spirited discussions with Lucienne Blessing when the idea seemed far too ambitious and unclear to be pursued at all. When the idea eventually became expressible enough, Nicholas Pinfield, the then Engineering Editor of Springer London, gave the much-needed encouragement for the project to take off on a serious note. I am thankful to all the contributors for their enthusiastic response, without which there would not be a credible proposal

with which to proceed. In particular, I am thankful to Susan Finger for her excitement at the idea of this book, and her suggestions for its improvement. In the more advanced stages, Oliver Jackson of Springer London has been extremely helpful with editorial support. The Cambridge Engineering Centre, my employer until recently, and John Clarkson, its director, have been generous with the facility; in particular, Andrew Flintham, the Computer Associate of the Centre, has been a great help in sorting out the computational problems faced. I am also grateful to Rob Bracewell for lending a patient ear whenever needed.

I would also like to thank the Centre for Product Design and Manufacturing at the Indian Institute of Science, my present employer, for its logistic support during the copy-editing and proof-reading stages of this book.

Finally, I would like to thank Ken Wallace and Thomas Bligh of Cambridge University for their effort in creating an ambience that fosters discussion, collaboration and integration, with creativity as an emergent, natural consequence. This book is as much a product of the effort of individuals as it is of this collective ambience.

Amaresh Chakrabarti
Bangalore
October 2001

Editor's introduction

Amaresh Chakrabarti

Engineering design, a central part of product development, is distinguished from other areas of human endeavour by its creative aspects, generally termed synthesis, whereby novel products are conceived. Engineering design synthesis is, therefore, a central area of design research. Traditionally, there have been consistent efforts in behavioural sciences to identify what constitutes creativity, and how it manifests itself in various aspects of human endeavour. Systematic research into design synthesis is relatively new. However, in the last few decades, especially with the increasing realisation of the potential of systematic design methods in enhancing design competence, and the advancement of computers as a potential design aid, the area has seen unprecedented growth. Descriptive studies and experiments have been undertaken, often in conjunction with psychologists and sociologists, to understand better the factors that influence this complex aspect of design. Many approaches have been, and are being, developed in order to enable, assist or even automate aspects of design synthesis; some of these approaches are theoretical, others are empirical, some are manual and others are computational.

This book brings together some of the most influential pieces of research undertaken around the world in design synthesis. It is the first, comprehensive attempt of this kind, and covers all three aspects of design synthesis research. Part 1 – Understanding – provides an overview of some of the major findings as to what constitutes synthesis and some of its major influencing factors. Part 2 – Approaches – provides a detailed description of some of the major prescriptive approaches to design synthesis, which together influenced many of the computational tools described in the final part. Part 3 – Tools – provides a selection of the diverse range of computational support techniques for synthesis in its major strands of research. It is to be noted that the parts have some overlap in content: the chapters in Understanding often propose approaches, and the chapters in Approaches and in Tools sometimes have well-developed theories that form part of the corpus of knowledge on which the current understanding of design synthesis is based. However, the part in which a chapter is placed signifies its main emphasis.

The chapters together provide an extensive coverage of the outcomes of design synthesis research in the last four decades: these include cutting-edge findings, as well as established, ready-to-use methods to help designers synthesise better ideas. The chapters are contributed by eminent researchers from four continents. Together, these chapters cover all major generic synthesis approaches, *i.e.*, composition, retrieval and change, and tackle problems faced in a wide variety of engineering domains and in many areas of application, including clocks, sensors and medical devices.

The rest of this article provides a summary of the chapters in the wider context of design synthesis research.

Part 1: Understanding

This part has five chapters. Together, these chapters provide insights into what constitutes and influences synthesis. Although all the chapters in this book are based, implicitly or explicitly, on some definition of synthesis, the first and last in this part are attempts to define and model the synthesis process. The other chapters in this part discuss the function and nature of knowledge necessary for synthesis.

In Chapter 1, Norbert Roozenburg provides an overview of the existing definitions of design synthesis, and their relationships to analysis. Synthesis, he argues, has taken two broad meanings in design research: as a distinct phase in designing, and as a part of the problem-solving process. Taking synthesis as the process of progressing from function to form, he analyses the logic of synthesis, and argues that certain kinds of synthesis cannot be attained by deduction alone and should require innovative abduction. This he terms innoduction, and defines as a reasoning process in which, given the intended function of a product, one must discover not only a form that can fulfil this function, but also the law that ascertains that the function can indeed be fulfilled by that form.

In Chapter 2, Michael French argues that insight into engineering science is the single most important influencing factor for good design synthesis. Drawing numerous examples from the history of designed artefacts, both industrial and household, and from both ancient and recent, he demonstrates that this engineering insight can often be encapsulated into a variety of "design principles". Research into, and use of, these principles should be very useful, he argues, but they are presently largely ignored and hardly researched.

In Chapter 3, Yoram Reich introduces the General Design Theory (GDT) of Hiroyuki Yoshikawa, which is one of the most mathematical of design theories. GDT is an axiomatic theory of design, which tries to establish the nature of knowledge necessary for engineering design in an idealistic sense, and the nature of designing given this knowledge. It also indicates how the nature of designing should change for existing engineering knowledge, which is far from ideal. Reich uses the domain of chairs as a simple example to explain GDT, and how designing is envisaged to proceed according to this theory.

In Chapter 4, Vladimir Hubka and Ernst Eder discuss their theory of technical systems and what it tells about synthesis. The theory of technical systems describes a technical system as one that fulfils a purpose using technical means, and proposes that it can be described at four levels of detail: process, function, organ and assembly. It prescribes synthesis as the process whereby these levels are achieved; in order to achieve these transformations, they suggest the use of various creative and systematic methods, such as brainstorming and morphological charts.

In Chapter 5, Tomiyama, Yoshioka and Tsumaya describe a model of the synthesis process developed by the "Modeling of Synthesis" project in Japan. Design is seen to be synonymous to synthesis; the relationship between the thought processes involved in synthesis and analysis are discussed, and synthesis is modelled in terms of knowledge and actions on knowledge. The theory is verified by developing a "reference model" from protocol data of designing sessions, and comparing the constructs of this model with that "predicted" by the model of synthesis developed.

Part 2: Approaches

This section has four chapters, each providing (the basis for) a prescriptive approach to design synthesis. It is interesting to note that most of these are based on theories of artefacts, although the nature and level of the approaches proposed vary considerably. Whereas the first three chapters provide outcome-based approaches of various degrees of detail, the fourth chapter provides a set of guidelines as to how areas of improvement can be found in a product, and how improvements can be effected. Together, the chapters provide guidelines as to how function and form can be developed.

Chapter 6 is by Claus Thorp Hansen and Mogens Myrup Andreasen and describes the domain theory of artefacts, which has been influenced by the theory of technical systems but which has evolved into one in which an artefact is described at three levels (called domains here): transformation, organ and part. Transformation between these is prescribed to take place using relationships that link functions to means, where each choice of a means leads to uncovering further functions and then to further means and so on, developing into a function–means tree. In this sense, synthesis of form for a given function could be seen, in a normative sense, as one of a bootstrapping process of developing means to fulfil a function and identifying functions required as a result.

In Chapter 7, Gerhard Pahl and Ken Wallace describe the function structures approach popularised by Pahl and Beitz. The function structures approach starts with the overall function necessary to be fulfilled by the intended product, and develops this into an assemblage of simpler subfunctions – a function structure. This is followed by a search for principles (means in Andreasen's terminology) that can fulfil each of these subfunctions, and combining them into concept variants. Using Krumhauer's generally valid functions as functional building blocks, they suggest the use of a morphological matrix to systematise the development of alternative concept variants, which, they argue, should lead to innovative designs.

Chapter 8, by Karlheinz Roth, describes another way of describing the various levels of an artefact description, but promulgates the use of design catalogues, with components made out of existing designs, for use by designers to achieve attainment of these levels. He argues that development and use of design catalogues, where each existing product or its components is described at multiple levels ranging from function through principles to form, should allow designers to reuse existing knowledge in an effective way. He uses several catalogues as examples to illustrate the variety and usefulness of design catalogues in designing.

In Chapter 9, Denis Cavallucci, an expert on the TRIZ approach developed by Genrich Altshuller of the former Soviet Union, introduces its basic components. Altshuller analysed a vast number of Russian patents to identify a set of "laws" that he believed were behind these patents. The laws are divided into three categories: static, cinematic and dynamic. Together, they help identify the areas in which an existing design can be improved and guidelines as to how this improvement can be pursued. Cavallucci also provides a comprehensive list of references on this approach, especially for the English-speaking reader.

Part 3: Tools

This part has ten chapters, which together exemplify all the major directions of research into computational (support to) synthesis of designs. Computational synthesis research has taken two major directions in the past: one is compositional synthesis, in which solutions are developed by combining a set of building blocks, and the other is retrieval of an existing design and its change for various purposes. The change effected may be to adapt the original design for the purpose at hand, or to modify it into other innovative designs.

The first two chapters are on automated compositional synthesis of concepts for fulfilment of a given function.

Chapter 10 is by Karl Ulrich and Warren Seering, and is one of earliest attempts at automated compositional synthesis of concepts. The area of application is sensors. The representational language is bond graphs, the algorithm is search, and the system developed is limited to synthesis of single-input single-output systems. Synthesis is performed at the topological level, and the resulting concepts are intended to be evaluated by the designer.

Chapter 11 is by Amaresh Chakrabarti, Patrick Langdon, Ying-Chieh Liu and Thomas Bligh, and is on the development of FuncSION – a multiple I/O concept synthesis software for mechanical transmissions and devices. The representation is based on systems theory and symbolic geometry, and the algorithm is search. Synthesis is performed at three levels: topology, spatial and generic physical. FuncSION has been tested using case studies, product compendia and patent catalogues. The designs synthesised are intended to be evaluated, modified and explored by the designer.

The next two chapters are examples of development of function into a function structure and support of compositional synthesis.

Chapter 12, by Rob Bracewell, is on the concepts underlying the Schemebuilder software for supporting design of mechatronic systems, involving mechanical, electrical and software elements. The representation is based on function–means trees and bond graphs. Using this system, a designer should be able interactively to develop the function and concept by a progressive proliferation of a function–means tree. The software has been tested using examples from several case studies.

Chapter 13 is by Ralf Lossack, and is for supporting the design of physical systems. The approach – DIICAD Entwurf – is a synthesis of systematic methodologies, and is based on the concept of a "working space" within which the design interacts with its inputs and outputs. Synthesis is done by designers selecting and concatenating means from a database. The software has been tested using several case studies and its use in student projects.

The next two chapters are examples of retrieval of existing designs.

Chapter 14 is by Tamotsu Murakami, and is on retrieval of existing mechanisms to fulfil a given, specified mechanical function. The representation used is based on qualitative configuration space, and the number of designs retrieved is one in each case. This has been tested using several cases, some of which are used as examples in the chapter. Retrieval is based on matching of the characteristics of intended function with that of the stored designs. The resulting designs are intended to be explored by the designer, but that is not currently supported within the framework.

Chapter 15 is by Lena Qian, and is on retrieval of mechanical, structural, hydraulic and software systems. The retrieval is done using analogy at three levels: function, behaviour and structure. The degree of similarity between the target and retrieved domains determines the choice of level used. Retrieved designs may be from a different discipline, and it is the task of the designer to transform the insight gained into an artefact appropriate for the domain in these cases.

The next two chapters are examples of changing retrieved designs for adapting to the current purpose.

Chapter 16 is by Sambasiva Bhatta and Ashok Goel, and is for adaptation using analogy. The current areas of application are electronic and mechanical controllers. The representation is based on logic and systems theory, and the adaptation mechanism is based on the use of design patterns with associated knowledge of what they can change into and how. The approach has been tested using several example cases.

Chapter 17 is by Boi Faltings on the FAMING system for adaptation of mechanisms. The software requires input from the designer for deciding the direction of modification and adapts the initial design using simple rules of replacement and envisionment. The representation is based on qualitative configuration space. This has been tested using several example cases, including those from architecture.

The final two chapters are on change from existing designs for generating innovative designs.

Chapter 18 is by Susan Finger and James Rinderle, and is on software that uses transformational grammar for changing a given intended behaviour or an existing design into new, behaviour-preserving designs. The current application is gear transmissions, and the representation used is bond graphs. This is one of the earliest papers that use grammars for generating designs, and is a precursor to much work on various generative grammars, not covered in this book.

Chapter 19 is by John Koza and is on software that uses genetic programming, which is based on the concept of genetic algorithms but uses programs that evolve in order to transform given designs to generate innovative designs with better performance in terms of the given criteria. The applications are electrical and electronic circuits and chemical reactions. The software has been tested using several case studies and patent catalogues.

Summary

Together, the chapters in this book provide a collection of views on the definition and nature of synthesis and some of its influencing factors, and a collection of approaches to synthesis. Below is a summary of these.

Definition of synthesis

There are five overlapping definitions of synthesis on which the chapters in this book are explicitly or implicitly based. These are:

- synthesis as designing;
- synthesis as problem solving;
- synthesis as design solution generation;
- synthesis as design problem and solution generation;
- synthesis as exploration.

According to the first definition (synthesis as designing), designing and synthesis are synonymous, as is propounded by Tomiyama and Yoshikawa. This appears to be used implicitly by Koza, who uses many cycles of generation and evaluation, operating at many levels of abstraction (topological, parametric, *etc.*) to develop a solution.

The second definition (synthesis as problem solving) means that one is operating at a particular level of abstraction, and uses a process involving both generation and decision (evaluation and selection) in order to develop a design solution at that level. In other words, synthesis is synonymous with problem solving. One example is the work of Finger and Rinderle, who use behaviour preservation as the evaluation process embedded in the algorithm to modify a given original graph representing an initial design to generate variants.

The third definition (synthesis as design solution generation) takes synthesis as a single part of the basic problem-solving process, which requires evaluation and selection in addition to this in order to complete the problem-solving cycle. In this sense, synthesis is synonymous with generation. Roozenburg mentions the ubiquity of this definition in design-process diagrams. This definition can be extended further to encompass generation of any design-related construction (synthesis as design problem and solution generation), if the problem-solving cycle is seen as cycling through at each level of design description through which a design develops: problem statement, requirements, functions, concept, embodiment, *etc.* It is the view taken in many engineering design methodologies not explicitly featured here, and is one way of describing the design process in practice [1].

The fifth view (synthesis as exploration) is different from the fourth in that it requires that synthesis be the process whereby clarity of the state of knowledge is increased. This is the definition implicitly used by the opportunistic strategy promulgated by Michael French, and many approaches described in this book try to support this process. Smithers [2], who takes this view of synthesis, gives a formal definition of exploration: it is the process by which a state of well-structured knowledge results from that of ill-structured knowledge.

Nature of synthesis and influencing factors

In designing, *designers* create an *artefact* by carrying out *activities* in an *environment* (settings, management, tools, *etc.*). Therefore, aspects of the human (designers, team), the artefact, design activities, and environment all affect design and its underlying synthesis process. Issues related largely to human and environmental aspects are not covered here. For human aspects, which include psychological studies of creativity, methods for enhancing idea generation, *etc.*, see among others Adams [3], Sternberg [4] and Frankenberger and Badke-Schaub [5]. For effects of environment, see Ottosson [6].

The chapters in the first two parts of this book cover some important aspects of the artefact, activities and underlying knowledge that make synthesis possible. Whereas Hubka and Eder, Hansen and Andreasen, and Roth highlight the necessity of artefactual knowledge and provide various views on the nature of this knowledge, Tomiyama *et al.* in particular present what they propose are the activities prevalent in design and syn-thesis. All chapters provide a viewpoint on the knowledge needed for synthesis. For instance, GDT (Reich) takes the view that this knowledge must lie in the relationships between entities and the functions that these entities are capable of performing. French claims that insight of engineering science is of essence, while TRIZ (Cavallucci) and other models provide various domain-neutral, procedural guidelines as to how these explorations may be carried out.

Between them, they propose three influences that are crucial for synthesis: (1) knowledge of artefact states; (2) knowledge of possible activities as progress from one state to another; and (3) knowledge of how these activities can be carried out.

Approaches to synthesis

Together, the chapters exemplify two major directions to synthesis: composition from scratch, and building on an existing design. Whereas compositional synthesis is often believed to enable generation of more innovative ideas, retrieval-based approaches are seen to be more efficient [7].

The essence of compositional synthesis is to bring the state of knowledge of the intended function of an artefact sufficiently close to that of the structural world such that a mapping between the two becomes possible. One way of doing this is to restructure the functional description such that each of its parts can be satisfied by composition of fragments of available artefacts. Another way of doing this is by decomposing the functional description using the functional descriptions of the existing artefacts themselves; this makes the generation process capable of being automated, with or without the intention of handing the resulting solutions to designers for exploration. The first two chapters, *i.e.*, Ulrich and Seering and Chakrabarti *et al.*, serve this purpose. The same can also be done by either decomposing the functional description sufficiently and then (composing and) replacing each with artefact fragments, thereby developing a composite artefact that fulfils the overall function. The chapters by Bracewell and by Lossack are intended to support this process.

Pure retrieval is seen as the most efficient way of developing a design, which requires no development at all. However, often the retrieved designs do not adequately fulfil the required functions, and need modification. The two chapters by Murakami and by Qian are primarily focused on retrieval, but both with the intention that the solutions retrieved should be modified, if necessary, by the designer to fulfil the requirements of the domain or the purpose. The issue of adaptation to fulfil the purpose is dealt with in the two chapters by Bhatta and Goel, and by Faltings. Once an initial design is retrieved (and adapted) for a given function, it can be used as a starting point for further modifications for generating other ideas either to produce variants or to optimise the design. Change is the theme of the last part, and is dealt with by the chapters by Finger and Rinderle, and by Koza.

The wider body of literature

Any anthology of this sort has to be indicative only of the body of literature at large, and cannot aspire to be exhaustive on any account. I mention some of the many interesting and useful studies, approaches and tools as pointers for readers who would like to delve into the wider body of literature beyond this book.

A number of researchers have developed theories of design and synthesis. Some notable ones are the knowledge level theory of designing by Tim Smithers [2], the situated model of design by Gero and Kannengiesser [8] and the reflection in action model of designing by Schön [9], and their implications on synthesis.

Many descriptive studies comment on the nature of synthesis in practice. See Fricke [10] and Ehrlenspiel *et al.* [11] for a case study where designers were observed, their attributes and design processes analysed and their solutions evaluated in order to measure success and success-promoting abilities. It was found, for instance, that balanced expansion of solution space and frequent evaluation of solutions are success-promoting factors. For an overview of descriptive studies with implications on synthesis, see Blessing [12].

A number of approaches to synthesis have been developed, for instance the functional reasoning approach developed by Freeman and Newell [13], and the prototyping approach developed by Gero and coworkers [14,15]. For a comprehensive review of synthesis techniques in various domains, see Flemming *et al.* [16].

Computational tools have been developed in a wide variety of domains and applications. For instance, several other researchers use compositional synthesis. Braha [17] uses adaptive search in his approach for finding optimal solutions in car configuration problems, Kota and Chiou [18] use search for mechanisms synthesis, Welch and Dixon [19] concatenate bond graph elements for synthesis of physical systems. Maher [20], Hundal [21], Umeda *et al.* [22], Malmqvist [23], and Alberts and Dikker [24] each developed an integrated framework for supporting synthesis of solutions, with goals broadly similar to Lossack and Bracewell.

Retrieval and repair has been a major theme of synthesis research, especially in case-based design [25]. For examples of (mainly) retrieval-based synthesis see Galletti and Giannotti [26] and McGarva [27], who use trial-and-error-based interactive selection of mechanisms from catalogues. For examples of retrieved designs see Sycara and Navinchandra [28], Madhusudan *et al.* [29], Joskowicz and Addanki [30], and Murthy and Addanki [31]. For mainly associative systems for innovative designs see the reviews by Navinchandra [32,33].

Changing existing designs for generation of new designs has been a continuing theme of synthesis research. Taura and Yoshikawa [34] use a metric space approach with adaptive search for this purpose. Grammar-based approaches use rules from a formal grammar to change designs. For examples of this see Shea and Cagan [35], Schmidt and Cagan [36], Heisserman [37] and Woodbury *et al.* [38], among others.

Most of the above references focus on the synthesis of solutions. However, the quality of the solution developed depends as much on the quality of solution synthesis as it does on the quality of problem finding. A number of interesting researches exist in development of support for identifying and representing requirements and functions, *e.g.*, see Wood and Antonsson [39] and O'Shaughnessy and Sturges [40].

For a more comprehensive coverage of articles related to design synthesis, the reader may find the following, by no means comprehensive, list of journals and conference proceedings useful: Research in Engineering Design (Springer); Design Studies (Elsevier); Journal of Engineering Design (Computational Mechanics); Proceedings of the International Conferences in Engineering Design (WDK); Proceedings of AI in Design Conferences (Kluwer); and Proceedings of the ASME Design Theory and Methodology Conferences (ASME).

References

[1] Blessing LTM. A process based approach to computer supported engineering design. Ph.D. thesis, University of Twente, Enschede, The Netherlands, 1994 [published in Cambridge by Blessing].

[2] Smithers T. Synthesis in designing as exploration. In: Proceedings of the 2000 Tokyo International Symposium on the Modeling of Synthesis, University of Tokyo, Japan, 11–13 December, 2000; 89–100.

[3] Adams JL. Conceptual blockbusting. 3rd ed. Reading (MA): Addison-Wesley, 1992.

[4] Sternberg RJ, editor. Handbook of creativity. New York: Cambridge University Press, 1999.

[5] Frankenberger E, Badke-Schaub P. Integration of group, individual and external influences in the design process. In: Frankenberger E, Badke-Schaub P, Birkhofer H, editors. Designers – the key to successful product development. London: Springer, 1998; 149–64.
[6] Ottosson S. Planetary organisations offer advantages in project work. In: Frankenberger E, Badke-Schaub P, Birkhofer H, editors. Designers – the key to successful product development. London: Springer, 1998; 196–201.
[7] Chakrabarti A. Towards hybrid methods for synthesis. In: International Conference on Engineering Design (ICED01), Design Research – Theories, Methodologies, and Product Modelling, Glasgow, 2001; 379–86.
[8] Gero JS, Kannengiesser U. Towards a situated function–behaviour–structure framework as the basis for a theory of designing. In: Smithers T, editor. Workshop on Development and Application of Design Theories in AI in Design Research, AI in Design'00 (AID00) Conference, Worcester, MA, July, 2000.
[9] Schön D. The reflective practitioner: how professionals think in action. New York: Basic Books, 1983.
[10] Fricke G. Successful individual approaches in engineering design. Res Eng Des 1996;8:151–65.
[11] Ehrlenspiel K, Dylla N, Guenther J. Experimental investigation of individual processes in engineering design. In: ICED '93, The Hague, 1993.
[12] Blessing L. Descriptive studies and design synthesis. In: Chakrabarti A, Blessing L, editors. Proceedings of the 1st Cambridge Workshop on Design Synthesis, Churchill College, Cambridge, 1999.
[13] Freeman P, Newell A. A model for functional reasoning in design. In: Proceedings of 2nd International Joint Conference in Artificial Intelligence, London, 1971; 621–40.
[14] Gero JS. Design prototypes: a knowledge representation schema for design. AI Mag 1990;(Winter): 26–36.
[15] Gero JS, Maher ML, Zhang W. Chunking structural design knowledge as prototypes. In: EDRC-12-25-88, Carnegie Mellon University, 1988.
[16] Flemming U, Adams J, Carlson C, Coyne R, Fenves S, Finger S, *et al.* Computational models for form–function synthesis in engineering design. In: EDRC 48-25-92, CMU, 1992.
[17] Braha D. Satisfying moments in synthesis. In: Chakrabarti A, Blessing L, editors. Proceedings of the 1st Cambridge Design Synthesis Workshop, Churchill College, Cambridge, UK, 7–8 October, 1999.
[18] Kota S, Chiou S-J. Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. Res Eng Des 1992;4:75–87.
[19] Welch RV, Dixon JR. Representing function, behavior and structure in conceptual design. In: Proceedings of the ASME Design Theory and Methodology Conference, DE-vol. 42, 1992.
[20] Maher ML. Synthesis and evaluation of preliminary designs. In: Proceedings of the International Conference on the Application of AI in Engineering, Cambridge, UK, July, 1989.
[21] Hundal MS. Use of functional variants in product development. In: ASME Design Theory and Methodology Conference, DE-vol. 31, 1991; 159–64.
[22] Umeda Y, Ishii M, Yoshioka M, Tomiyama T. Supporting conceptual design based on the function–behavior–state modeler. Artif Intell Eng Des Anal Manuf 1996;10(4):275–88.
[23] Malmqvist J. Computational synthesis and simulation of dynamic systems. In: ICED '95, Praha, 1995.
[24] Alberts LK, Dikker F. Integrating standards and synthesis knowledge using the YMIR ontology. In: Gero JS, Sudweeks F, editors. AI in Design '94. Kluwer, 1994; 517–34.
[25] Kolodner JL. Case-based reasoning. San Mateo (CA): Morgan Kaufmann Publishers, 1993.
[26] Galletti CU, Giannotti EI. Interactive computer system for the functional design of mechanisms. Comput Aided Des 1981;12(3):159–63.
[27] McGarva JR. Rapid search and selection of path generating mechanisms from a library. Mech Mach Theory 1994;29(2):223–35.
[28] Sycara K, Navinchandra D. Retrieval strategies in a case-based design system. In: Tong C, Sriram D, editors. Artificial intelligence in engineering design, vol. II. Academic Press, 1992.
[29] Madhusudan TN, Sycara K, Navinchandra D. A case based reasoning approach for synthesis of electro-mechanical devices using bond graphs. EDRC report, Carnegie-Mellon University, USA, 1995.
[30] Joskowicz L, Addanki S. From kinematics to shape: an approach to innovative design. In: Proceedings of AAAI-88, St Paul MN, 1988; 347–52.
[31] Murthy SS, Addanki S. PROMPT: an innovative design tool. In: Proceedings of the National Conference of the American Association for Artificial Intelligence, 1987; 637–42.

[32] Navinchandra D. Innovative design systems, where are we and where do go from here? Part I: design by association. Knowl Eng Rev 1992;7(3):183–213.
[33] Navinchandra D. Innovative design systems, where are we and where do we go from here? Part II: design by exploration. Knowl Eng Rev 1992;7(4):345–62
[34] Taura T, Yoshikawa H. A metric space for intelligent CAD. In: Brown DC, Waldron M, Yoshikawa H, editors. IFIP intelligent computer aided design. Elsevier/North-Holland, 1992; 133–61.
[35] Shea K, Cagan J. Generating structural essays from languages of discrete structures. In: Proceedings of Artificial Intelligence in Design AID 98, Lisbon, July, 1998; 365–84.
[36] Schmidt LC, Cagan J. GGREADA: a graph grammar based machine design algorithm. Res Eng Des 1997;9:195–213.
[37] Heisserman J. Generative geometric design. IEEE Comput Graph Appl 1994;14(2):37–45.
[38] Woodbury R, Datta S, Burrow A. Towards an ontological framework for knowledge-based design systems. In: Proceedings of the 6th AI in Design Conference (AID00), 24–26 June, Worcester, MA, 2000.
[39] Wood KL, Antonsson EK. A first class of computational tools for preliminary engineering design. Technical report, California Institute of Technology, 1992.
[40] O'Shaughnessy K, Sturges RH Jr. A systematic approach to conceptual engineering design. In: ASME Design Theory and Methodology Conference, DE-vol. 42, 1992; 283–90.

Contents

Preface v
Editor's introduction vii
Contents xvii
Contributors xxvii

Part 1: Understanding

1 **Defining synthesis: on the senses and the logic of design synthesis**
Norbert F.M. Roozenburg 3
1.1 Senses of synthesis 3
1.1.1 General meanings of synthesis and analysis 3
1.1.2 Synthesis and analysis as phases of the design process 4
1.1.3 Synthesis and analysis as functions of problem solving 6
1.1.4 Synthesis as assemblage of subsystems 7
1.1.5 Synthesis as integration of ideas 8
1.2 The logic of synthesis 9
1.2.1 Form and function 9
1.2.2 Reasoning from function to form 10
1.2.3 The pattern of reasoning of synthesis 12
1.2.4 Conclusions 16

2 **Insight, design principles and systematic invention**
Michael J. French 19
2.1 Introduction 19
2.2 The opportunistic designer 19
2.2.1 The opportunistic approach 20
2.3 Parallels with mathematics 20
2.3.1 Poincaré's sieve 22
2.3.2 Visual thought 22
2.4 Insight 23
2.5 Developing insight 24
2.5.1 Sufficient insight 25
2.6 Design principles 25
2.6.1 Kinematic design (least constraint) 25
2.6.2 The small, fast principle 26
2.6.3 Matching 27
2.6.4 "Prefer pivots to slides and flexures to either" 27
2.6.5 "Where possible, transfer complexity to the software" 28

2.7 Systematic synthesis ... 28
2.7.1 Clothes-peg example ... 28
2.8 Insight and systematic invention in power from sea waves ... 29
2.8.1 Background ... 29
2.8.2 An abstract view ... 29
2.8.3 Table of options ... 30
2.8.4 Embodiment ... 31
2.8.5 The checking of systematic design processes-link-breaking ... 32
2.9 Summary ... 32
Appendix 2A ... 33

3 Synthesis and theory of knowledge: general design theory as a theory of knowledge, and its implication to design
Yoram Reich ... 35
3.1 Introduction ... 35
3.2 The domain of chairs ... 36
3.3 GDT ... 38
3.3.1 Preliminary definitions ... 38
3.3.2 GDT's axioms ... 39
3.3.3 Ideal knowledge ... 40
3.3.3.1 Summary of ideal knowledge ... 42
3.3.4 Real knowledge ... 42
3.3.4.1 Summary of real knowledge ... 44
3.4 Contribution of GDT ... 45
3.4.1 Representation of design knowledge ... 45
3.4.2 Design process ... 46
3.5 Summary ... 47

4 Theory of technical systems and engineering design synthesis
Vladimir Hubka and W. Ernst Eder ... 49
4.1 Introduction ... 49
4.2 Design science and the theory of TSs ... 50
4.3 Designing – general ... 56
4.3.1 Starting designing – clarifying the problem – design specification ... 61
4.3.2 Designing – design procedure – novel products ... 62
4.3.3 Designing – design procedure – redesigned products ... 65

5 A knowledge operation model of synthesis
Tetsuo Tomiyama, Masaharu Yoshioka and Akira Tsumaya ... 67
5.1 Introduction ... 67
5.2 Related work ... 68
5.3 Design process modelling ... 69
5.4 A formal model of synthesis ... 69
5.4.1 Mathematical preparation ... 71
5.4.2 Analysis versus synthesis ... 73
5.4.3 Multiple model-based reasoning ... 74
5.4.4 Function modelling ... 76
5.4.5 A reasoning framework of design ... 77

5.4.5.1 Knowledge operations in design 77
5.4.5.2 A hypothetical reasoning framework of design 78
5.4.5.3 Modelling operations in the object-dependent models 78
5.4.5.4 Logical reasoning operations in the object independent level workspace 79
5.4.5.5 Formalising knowledge operations in design 79
5.5 The implementation strategy of the framework 79
5.5.1 Multiple model-based reasoning system 80
5.5.2 Thought-process model 80
5.5.3 Model-based abduction 80
5.5.3.1 Strategy for modelling of abduction operation 80
5.5.3.2 Model-based abduction 82
5.5.3.3 Algorithm of model-based abduction 83
5.6 Verification of the model of synthesis 84
5.6.1 Selection of data for verification 84
5.6.2 The reference model 85
5.6.3 Verification of the knowledge operation model 85
5.6.4 Vocabulary about design 86
5.6.5 Verification through implementation of the reasoning framework 87
5.7 Conclusions 88

Part 2: Approaches

6 Two approaches to synthesis based on the domain theory
Claus Thorp Hansen and Mogens Myrup Andreasen 93
6.1 Introduction 93
6.2 The domain theory 94
6.2.1 Systems theory 94
6.2.2 The domain theory 95
6.2.2.1 The transformation domain 95
6.2.2.2 The organ domain 96
6.2.2.3 The part domain 97
6.2.2.4 Visualising the domain theory 98
6.3 The function–means law 99
6.3.1 The function–means tree (F/M-tree) 99
6.4 Engineering design synthesis 99
6.4.1 The design object and its synthesis 100
6.4.2 Process-oriented synthesis 101
6.4.2.1 Problem analysis 105
6.4.3 Artefact-oriented synthesis approach 105
6.4.3.1 Utilising the F/M-tree 105
6.4.3.2 Developing a product model 106
6.5 Implications and conclusion 107

7 Using the concept of functions to help synthesise solutions
Gerhard Pahl and Ken Wallace 109
7.1 Introduction 109
7.2 Functional interrelationship 110

7.3 Handling the concept of functions in practice 114
7.4 Inappropriate use of the concept of functions 118
7.5 Summary of the approach and its advantages 118

8 Design catalogues and their usage
Karlheinz Roth 121
8.1 Purpose of design catalogues 121
8.2 Types and structure of design catalogues 121
8.2.1 Object catalogues 121
8.2.2 Solution catalogues 123
8.2.3 Operation catalogues 125
8.3 Requirements placed on design catalogues 127
8.4 Desirable forms of design catalogue 127
8.5 Use of design catalogues 128

9 TRIZ, the Altshullerian approach to solving innovation problems
Denis Cavallucci 131
9.1 The genesis of a theory 131
9.1.1 Introduction 131
9.1.2 Altshuller: evaluation of a life dedicated to others 132
9.1.3 Opening to the West gives TRIZ the opportunity to develop 133
9.2 An approach to classifying Altshuller's work 133
9.2.1 Introduction 133
9.2.2 Basic notions 134
9.2.3 Spotting Altshuller's original idea: the laws (or regularities) of developing technical systems 134
9.2.3.1 The "static" laws 134
9.2.3.2 The "cinematic" laws 136
9.2.3.3 The "dynamic" laws 136
9.2.3.4 Summary of the laws 137
9.2.4 Case study: improving the performance of an intake manifold [10] 137
9.2.4.1 Description of the problem 137
9.2.4.2 Positioning the manifold in relation to the laws of evolution 137
9.2.4.3 Interpreting the positioning 139
9.2.4.4 Findings of the study 139
9.2.5 Tools for breaking down the blockages of psychological inertia 140
9.2.6 Problem-solving tools 141
9.2.7 ARIZ, the algorithm for applying TRIZ 141
9.3 TRIZ's contribution to integration in the design process 142
9.3.1 Using TRIZ in an approach consisting of applying a series of tools 142
9.3.2 TRIZ as a "meta-method" 143
9.3.3 TRIZ as a component part of an existing method 143
9.3.4 The intuitive design model approach to methodological integration 143

9.4 Potential development of the theory in research 144
9.4.1 Contributions to integrating TRIZ in one or more existing methods 144
9.4.2 Contributions to the development of TRIZ itself 145
9.4.3 Contributions to other fields of activity 145
9.5 Orchestrating the work in Altshuller's wake 145
9.6 Conclusions 146
9.6.1 Industrial integration strategies 146
9.6.2 Creativity and innovation: the missing (or forgotten) link in the design process 147
9.6.3 An asset for product design 147

Part 3: Tools

10 Synthesis of schematic descriptions in mechanical design
Karl T. Ulrich and Warren P. Seering 153
10.1 Introduction 153
10.1.1 What is schematic synthesis? 154
10.1.2 Schematic synthesis of SISO systems 155
10.1.3 Importance of schematic synthesis 156
10.1.3.1 Reducing complexity 157
10.1.3.2 Decoupling functional and physical issues 157
10.2 Domain description 157
10.2.1 SISO dynamic systems 157
10.2.2 Representing schematic descriptions 158
10.2.3 Classifying the behaviour of a schematic description 158
10.2.4 Specifying a problem 160
10.2.4.1 Example specification 161
10.3 Solution technique 162
10.3.1 Generating candidate descriptions 162
10.3.1.1 Concept of a power spine 162
10.3.1.2 Connecting input to output 163
10.3.2 Classifying behaviour 163
10.3.3 Modifying candidate schematic descriptions 163
10.3.3.1 Transform the candidate design to a compact description 163
10.3.3.2 Based on domain knowledge, generate modifications 165
10.3.3.3 Reverse compacting transformation 166
10.4 A complete example 167
10.5 Discussion 170
10.5.1 Importance and utility of technique 171
10.5.2 Completeness of the technique 171
10.5.3 Extensibility 172
10.5.3.1 Extension within dynamic systems domain 172
10.5.3.2 Extension to other domains 173
10.5.4 Computer implementation 173
10.6 Related work 174

Appendix 10A ... 175
10A.1 Determining the type number from the system equations .. 175
10A.2 An explanation of isolated groups using bond graphs ... 176

11 An approach to compositional synthesis of mechanical design concepts using computers
Amaresh Chakrabarti, Patrick Langdon, Yieng-Chieh Liu and Thomas P. Bligh ... 179
11.1 Objective ... 179
11.2 Research approach ... 180
11.3 Synthesis approach: representation, reasoning and example ... 182
11.3.1 Develop theory from known design problems and solutions ... 182
11.3.2 Generate solutions to known problems and compare with existing designs ... 186
11.4 Evaluation of the synthesis approach ... 188
11.4.1 MAS project case studies ... 188
11.4.2 Hands-on experiments by experienced designers ... 188
11.5 Further developments ... 190
11.5.1 Resolving the first problem: managing the number of solutions generated ... 191
11.5.1.1 Improving efficiency of the synthesis procedure ... 191
11.5.1.2 Using additional constraints ... 192
11.5.1.3 Grouping solutions using similarity ... 192
11.5.2 Resolving the second problem: strategies for aiding visualisation ... 193
11.5.2.1 Embodiment at the generic physical level ... 193
11.5.2.2 Three-dimensional representation of the solution space ... 194
11.6 Conclusions and further work ... 194

12 Synthesis based on function–means trees: Schemebuilder
Rob Bracewell ... 199
12.1 Background ... 199
12.2 Key concepts of Schemebuilder ... 200
12.2.1 Hierarchical schematic diagrams ... 200
12.2.2 Scheme generation by combination of alternative subsolutions ... 200
12.2.3 Function–means trees ... 201
12.2.4 Artificial intelligence (AI) support for design context decomposition and recombination ... 201
12.2.5 Computer support for simulation and evaluation of schemes ... 204
12.2.6 Bond-graph-based functional synthesis ... 205
12.3 Design synthesis example: telechiric hand ... 205
12.4 Implementation of function–means-based synthesis ... 209

13 Design processes and context for the support of design synthesis
Ralf-Stefan Lossack ... 213
13.1 Introduction and overview of the design process ... 213

13.2 Solution patterns 217
13.2.1 Artefact and process knowledge 217
13.3 Design working space 220
13.4 The DIICAD Entwurf design system 224
13.5 Conclusion 224
13.6 Future work 225

14 Retrieval using configuration spaces
Tamotsu Murakami 229
14.1 Introduction 229
14.2 Mechanism library 229
14.2.1 Mechanism and configuration space 230
14.2.2 Kinematic behaviour and configuration space 231
14.2.3 Additional behavioural information description 232
14.3 Required behaviour as retrieval key 233
14.3.1 Required behaviour description 233
14.3.1.1 Timing charts of input/output motions 233
14.3.1.2 Types of input/output motion 234
14.3.1.3 Motion speed dependence 234
14.3.2 Required locus pattern generation 234
14.4 Locus pattern and configuration space matching 234
14.4.1 Locus along region boundary 235
14.4.1.1 Motion by object contact 235
14.4.1.2 Compliance 236
14.4.2 Locus along range limit 236
14.4.3 Locus through free region 237
14.4.4 Generation of entire locus from segments 237
14.4.5 Check additional conditions on motion 238
14.5 Implementation and execution examples 238
14.5.1 Mechanism library 238
14.5.2 Specifying required behaviour 239
14.5.3 Example 1: mechanism for shutter release 239
14.5.4 Example 2: mechanism in sewing machine 241
14.6 Conclusions and discussions 242

15 Creative design by analogy
Lena Qian 245
15.1 Introduction 245
15.2 Knowledge representation for design retrieval based on analogy .. 246
15.2.1 Structure 246
15.2.1.1 Primitive element and structural element 248
15.2.1.2 Attribute 248
15.2.1.3 Relationship 248
15.2.1.4 Operation and process 249
15.2.1.5 Static and dynamic structure 249
15.2.2 Behaviour 250
15.2.3 Function 253
15.2.4 Qualitative causal knowledge 256
15.2.5 Design prototype 257
15.3 An ABD model 258

15.3.1 Design retrieval process 259
15.3.2 Analogy elaboration process 260
15.3.3 Mapping and transference 260
15.3.4 Analogy evaluation process 263
15.4 Design support system using analogy 263
15.5 An example of designing a new door by behaviour analogy 264
15.6 Conclusion 267

16 Design patterns and creative design
Sambasiva R. Bhatta and Ashok K. Goel 271
16.1 Background, motivations and goals 271
16.2 MBA 272
16.2.1 SBF models of devices 273
16.2.2 Design patterns 275
16.3 Acquisition of GTMs 276
16.4 Analogical transfer based on GTMs 277
16.5 Evaluation 282
16.6 Related research 282
16.7 Conclusions 283

17 FAMING: supporting innovative design using adaptation – a description of the approach, implementation, illustrative example and evaluation
Boi Faltings 285
17.1 Introduction 285
17.1.1 Model-based design 286
17.1.2 Prototype-based design 287
17.1.3 Case-based design 287
17.1.4 Annotating cases with functional models 287
17.1.5 Case adaptation using SBF models 288
17.1.6 Innovation in case-based design 288
17.1.7 FAMING: an interactive design tool 289
17.2 Qualitative SBF models used in FAMING 289
17.2.1 Structure: metric diagram 290
17.2.2 Qualitative behaviour 290
17.2.2.1 Qualitative motions 290
17.2.2.2 External influences 291
17.2.2.3 Place vocabulary 291
17.2.2.4 Behaviour = envisionments of kinematic states ... 292
17.2.3 A language for specifying function 292
17.2.3.1 Quantitative constraints on behaviour 294
17.3 Inverting the FBS model 294
17.3.1 Matching behaviour to functional specification 294
17.3.2 S–B inversion 295
17.4 Case adaptation 295
17.4.1 Case combination 296
17.4.2 Modification operators 297
17.4.3 Discovering and satisfying compositional constraints 298
17.5 Conclusions 299

18 Transforming behavioural and physical representations of mechanical designs
Susan Finger and James R. Rinderle 303
18.1 Introduction 303
18.2 Related work 304
18.2.1 A brief introduction to bond graphs 305
18.2.2 Representation of function and behaviour 305
18.2.3 Grammars for representation of geometry 306
18.2.4 Configuration design 307
18.3 Representation of behaviour of specifications and components 307
18.3.1 Representation of design specifications 307
18.3.2 Representation of behavioural requirements of mechanical systems 308
18.3.3 Representation of behavioural characteristics of components 309
18.3.4 Representation of designs 309
18.4 Transformation of specifications into physical descriptions 310
18.4.1 Behaviour-preserving transformations 310
18.4.2 Component-directed transformations 311
18.5 The shaft matrix 312
18.6 Conclusions 316

19 Automatic synthesis of both the topology and numerical parameters for complex structures using genetic programming
John R. Koza 319
19.1 Introduction 319
19.2 Genetic programming 320
19.3 Automatic synthesis of analog electrical circuits 322
19.3.1 Lowpass filter circuit 324
19.3.1.1 Preparatory steps for lowpass filter circuit 324
19.3.1.2 Results for lowpass filter circuit 326
19.3.2 Squaring computational circuit 329
19.3.2.1 Preparatory steps for squaring computational circuit 329
19.3.2.2 Results for squaring computational circuit 329
19.4 Automatic synthesis of controllers 331
19.5 Other examples 333
19.6 Conclusions 335

Index 339

Contributors

Mogens Myrup Andreasen
Section of Engineering Design and Product Development
Department of Mechanical Engineering
Technical University of Denmark
Lyngby
Denmark

Sambasiva R. Bhatta
Verizon Communications
White Plains
USA

Thomas P. Bligh
Engineering Design Centre
Department of Engineering
Cambridge University
UK

Rob Bracewell
Engineering Design Centre
Department of Engineering
Cambridge University
UK

Denis Cavallucci
Ecole Nationale Supérieure des Arts et Industries de Strasbourg
France

Amaresh Chakrabarti
Centre for Product Design and Manufacturing
Indian Institute of Science
India

W. Ernst Eder (Professor Emeritus)
Department of Mechanical Engineering
Royal Military College of Canada
Canada

Boi Faltings
Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
Lausanne
Switzerland

Susan Finger
Department of Civil and Environmental Engineering
Carnegie Mellon University
Pittsburgh
USA

Michael J. French (Professor Emeritus)
Engineering Design Centre
Lancaster University
UK

Ashok K. Goel
College of Computing
Georgia Institute of Technology
Atlanta
USA

Claus Thorp Hansen
Section of Engineering Design and Product Development
Department of Mechanical Engineering
Technical University of Denmark
Lyngby
Denmark

Vladimir Hubka (Professor Emeritus)
Swiss Federal Institute of Technology
Zurich
Switzerland

John R. Koza (Consulting Professor)
Department of Electrical Engineering
School of Engineering
Stanford University
USA

Patrick Langdon
Engineering Design Centre
Department of Engineering
Cambridge University
UK

Yieng-Chieh Liu
Engineering Design Centre
Department of Engineering
Cambridge University
UK

Ralf-Stefan Lossack
Institute of Applied Computer Science in Mechanical Engineering
University of Karlsruhe
Germany

Tamotsu Murakami
Department of Engineering Synthesis
The University of Tokyo
Japan

Gerhard Pahl (Professor Emeritus)
Fachbereich 16 Maschinenbau
Technische Hochschule Darmstadt
Germany

Lena Qian
Canon Information Systems Research Australia
North Ryde
Australia

Yoram Reich
Department of Solid Mechanics, Materials and Systems
Faculty of Engineering
Tel Aviv University
Israel

James R. Rinderle
Department of Mechanical and Industrial Engineering
University of Massachusetts
Amherst
USA

Norbert F.M. Roozenburg
Faculty of Design, Engineering and Production
Technical University of Delft
The Netherlands

Karlheinz Roth (Professor Emeritus)
Institute for Engineering Design, Machine and High-Precision Elements
Braunschweig Technical University
Germany

Warren P. Seering
Department of Mechanical Engineering
Massachusetts Institute of Technology
USA

Tetsuo Tomiyama
Research into Artifacts, Center for Engineering (RACE)
The University of Tokyo
Japan

Akira Tsumaya
Collaborative Research Center for Advanced Science and Technology
Osaka University
Japan

Karl T. Ulrich
Operations and Information Management Department
Wharton School
University of Pennsylvania
USA

Ken Wallace
Engineering Design Centre
Department of Engineering
Cambridge University
UK

Masaharu Yoshioka
Rsearch Center for Information Resources
National Institute of Informatics
Tokyo
Japan