John S. Gero *Editor*

# Design Computing and Cognition '12

Springer

# Design Computing and Cognition ’12

John S. Gero
Editor

# Design Computing and Cognition ’12

Springer

*Editor*
John S. Gero
Krasnow Institute for Advanced Study
Fairfax, VA
USA

ISBN 978-94-017-9111-3 ISBN 978-94-017-9112-0 (eBook)
DOI 10.1007/978-94-017-9112-0
Springer Dordrecht Heidelberg New York London

Library of Congress Control Number: 2014940323

© Springer Science+Business Media Dordrecht 2014
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

# Preface

The first mention of design appears in the Code of Hammurabi around 1750 BC. This was an enunciation of the moral code of that time and the design part covered building design. The next mention appears to be around 50 BC in Vitruvius' *De architectura: Ten Books of Architecture* that outlined design knowledge in the form of rules, both prescriptive and performance. Vitruvius covered both machine design and building design. In 1452 Leon Battista Alberti published *De re aedificatoria: Ten Books of Architecture* that introduced the notion of design process as an intellectual activity. Designing as a human intellectual activity has its roots in human needs expressed by changing the natural world in order to address those needs and then by changing the world that includes designed artifacts. Since designing results in both economic and social benefit it is therefore surprising how little the design world has been studied compared to the physical world we inhabit.

Design research, largely started only 50 years ago, has started to provide some insight into both design processes and designed objects.

Design thinking, the label given to the unique act of designing, has become a paradigmatic view that has transcended the discipline of design and is now widely used in business and elsewhere. As a consequence, there is an increasing interest in design research and government agencies are gradually increasing funding for design research, and increasing numbers of engineering and computer science schools are revising their curricula to emphasize design. This is because of the realization that design is part of the wealth creation of a nation and needs to be better understood and taught. The continuing globalization of industry and trade has required nations to re-examine where their core contributions lie, if not, in production efficiency. Design is a precursor to manufacturing for physical objects and is the precursor to implementation for virtual objects. At the same time, the need for sustainable development is requiring the design of new products and processes, and feeding a movement toward design innovations and inventions.

This conference series aims at providing a bridge between the fields of design computing and design cognition. The confluence of these two fields continues to provide the foundation for further advances in each of them and to an increased understanding of this field whose influence continues to spread.

The papers in this volume are from the *Fifth International Conference on Design Computing and Cognition (DCC'12)* held at Texas A&M University, College Station, Texas, USA. They represent the state of the art of research and development in design computing and design cognition. They are of particular interest to researchers, developers, and users of advanced computation in design and those who need to gain a better understanding of designing.

In these proceedings the papers are grouped under the following nine headings, describing both advances in theory and application and demonstrating the depth and breadth of design computing and design cognition:

Design by Analogy
Design Cognition—1
Design Creativity
Design Cognition—2
Design Generation
Shape and Space
Design Knowledge
Design Function
Design Processes

There were 91 full paper submissions to the conference of which 34 were accepted and presented and appear in these proceedings. Each paper was extensively reviewed by at least three reviewers drawn from the international panel of 98 active reviewers listed on the next pages. The reviewers' recommendations were then assessed before the final decision on each paper was taken. Thanks go to them, for the quality of these papers depends on their efforts.

Mercedes Paulini and Pinelopi Kyriazi assisted in bringing the papers in this volume into a uniform whole, special thanks go to them.

John S. Gero
Krasnow Institute for Advanced Study
Fairfax, VA, USA

# Reviewers

Henri Achten, Czech Technical University, Czech Republic
Robin Adams, Purdue University, USA
Alice Agogino, University of California Berkeley, USA
Janet Allen, University of Oklahoma, USA
Tomasz Arciszewski, George Mason University, USA
Uday Athavankar, IIT Bombay, India
Petra Badke-Schaub, TU Delft, The Netherlands
Kirsty Beilharz, University of Technology, Sydney, Australia
Eric Blanco, Grenoble INP, France
Nathalie Bonnardel, University of Provence, France
Frances Brazier, TU Delft, The Netherlands
David C. Brown, Worcester Polytechnic Institute, USA
Winslow Burleson, Arizona State University, USA
Jonathan Cagan, Carnegie Mellon University, USA
Marco Cantamessa, Politecnico di Torino, Italy
Hernan Casakin, Ariel University Center of Samaria, Israel
Gabriela Celani, UNICAMP, Brazil
Amaresh Chakrabarti, Indian Institute of Science, India
Scott Chase, Aalborg University, Denmark
Henri Christiaans, TU Delft, The Netherlands
Per Christiansson, Aalborg University, Denmark
Mark Clayton, Texas A&M University, USA
Graham Coates, Durham University, UK
Nathan Crilly, University of Cambridge, UK
Andrew Crooks, George Mason University, USA
Umberto Cugini, Polytecnico Milan, Italy
Ellen Yi-Luen Do, George Institute of Technology, USA
Andy Dong, University of Sydney, Australia
Jose Duarte, Instituto Superior Técnico, Portugal
Chris Earl, Open University, UK
Claudia Eckert, University of Cambridge, UK
Thanos Economou, Georgia Institute of Technology, USA
Georges Fadel, Clemson University, USA
Gerhard Fischer, University of Colorado, USA

Christian Freksa, University of Bremen, Germany
Daniel Frey, MIT, USA
John Gero, George Mason University, USA
Ashok Goel, Georgia Institute of Technology, USA
Gabriela Goldschmidt, Technion, Israel
Ewa Grabska, Jagiellonian University, Poland
Kaz Grace, University of Sydney, Australia
Ning Gu, University of Newcastle, Australia
David Gunaratnam, University of Sydney, Australia
Andrés Gómez, ITAM, Mexico
Sean Hanna, University of College London, UK
Urs Hirschberg, Graz University of Technology, Austria
Hao Jiang, National University of Singapore, Singapore
Yan Jin, University of Southern California, USA
Leo Joskowicz, Hebrew University of Jerusalem, Israel
Yehuda Kalay, University of California Berkeley, USA
Jeff Kan, University of Sydney, Australia
Udo Kannengiesser, NICTA, Australia
Nick Kelly, University of Sydney, Australia
Yong Se Kim, Sungkyunkwan University, Korea
Terry Knight, MIT, USA
Ramesh Krishnamurti, Carnegie Mellon University, USA
Djordje Krstic, Hetzel Design Inc., USA
Pascal Le Masson, MINES ParisTech, France
Pierre Leclercq, University of Liege, Belgium
Noel Leon, ITESM, Mexico
Julie Linsey, Georgia Institute of Technology, USA
Mary Lou Maher, University of Maryland, USA
Anja Maier, DTU, Denmark
Peter Matthews, Durham University, UK
Janet Mcdonnell, University of Arts London, UK
Amitabha Mukerjee, IIT Kanpur, India
Yukari Nagai, JAIST, Japan
Douglas Noble, University of Southern California, USA
Rivka Oxman, Technion, Israel
Panos Papalambros, University of Michigan, USA
Morteza Pourmohamadi, Tabriz Islamic Art University, Iran
Lawrence Sass, MIT, USA
Linda Schmidt, University of Maryland, USA
Gerhard Schmidt, ETH Zurich, Switzerland
Jami Shah, Arizona State University, USA
Kristina Shea, TU Munich, Germany
Li Shu, University of Toronto, Canada
Vishal Singh, NYU, USA
Ian Smith, EPFL, Switzerland

Steven Smith, Texas A&M University, USA
Tim Smithers, Consultant, Spain
Ricardo Sosa, ITESM, Mexico
Martin Stacey, de Mountford University, UK
George Stiny, MIT, USA
Rudi Stouffs, TU Delft, The Netherlands
Markus Stumptner, University of South Australia, Australia
Hsien-Hui Tang, National Taiwai University of Science and Technology, Taiwan
Toshiharu Taura, Kobe University, Japan
Tetsuo Tomiyama, TU Delft, The Netherlands
Jan Treur, Vrije Universiteit Amsterdam, The Netherlands
Benamy Turkienicz, Universidade Federal do Rio Grande do Sul, Brazil
Barbara Tversky, Columbia Teachers College and Stanford University, USA
Andrew Vande Moere, KU Leuven, Belgium
Willemien Visser, INRIA, France
Chris Williams, VPI, USA
Maria Yang, MIT, USA
Seda Yilmaz, Iowa State University, USA
Robert Youmans, George Mason University, USA

# Contents

Part I Design by Analogy

Part II Design Cognition—1

## Part III Design Creativity

## Part IV Design Cognition—2

## Part V Design Generation

**Part VI Shape and Space**

**Part VII Design Knowledge**

## Part VIII Design Function

## Part IX Design Processes

# Contributors

**Kinda Al Sayed** University College London, London, UK

**Tomasz Arciszewski** George Mason University, Fairfax, VA, USA

**Serap Arikoglu** Grenoble-INP, Grenoble, France

**Mário Barros** Technical University of Lisbon, Lisbon, Portugal

**Deborah Benrós** University College London, London, UK

**Eric Blanco** Grenoble-INP, Grenoble, France

**Prasad Bokil** Indian Institute of Technology Bombay, Mumbai, India

**Jonathan Cagan** Carnegie Mellon University, Pittsburgh, PA, USA

**Matthew I. Campbell** University of Texas at Austin, Austin, USA

**B. M. Chaparro** Politechnic Institute of Tomar, Tomar, Portugal

**Chang Chen** University of Southern California, Los Angeles, CA, USA

**Pinzhi Chen** Carnegie Mellon University, Pittsburgh, PA, USA

**Yong Chen** Shanghai Jiao Tong University, Shanghai, China

**Hyunmin Cheong** University of Toronto, Toronto, Canada

**Ivey Chiu** Ryerson University, Toronto, ON, Canada

**Jia Cui** The Hong Kong Polytechnic University, Kowloon, Hong Kong

**Andreea Danielescu** Arizona State University, Tempe, AZ, USA

**Mahmoud Dinar** Arizona State University, Tempe, AZ, USA

**Andy Dong** University of Sydney, Sydney, NSW, Australia

**José Pinto Duarte** ADETTI-IUL and TU Lisbon, Lisbon, Portugal

**Chris Earl** The Open University, Milton Keynes, UK

**Julian R. Eichhoff** Furtwangen University, Furtwangen im Schwarzwald, Germany

**Tamer El-Khouly** University College London, London, UK

**Sara Eloy** ADETTI-IUL and TU Lisbon, Lisbon, Portugal

**Katherine Fu** Carnegie Mellon University, Pittsburgh, PA, USA

**John S. Gero** Krasnow Institute for Advanced Study, Fairfax, VA, USA

**Michael W. Glier** Texas A&M University, College Station, TX, USA

**Ashok K. Goel** Georgia Institute of Technology, Atlanta, GA, USA

**Ning Gu** University of Newcastle, Newcastle, NSW, Australia

**Gregory Hallihan** University of Toronto, Toronto, Canada

**Sean Hanna** University College London, London, UK

**Michael E. Helms** Georgia Institute of Technology, Atlanta, GA, USA

**Noe Vargas Hernandez** University of Texas at El Paso, El Paso, TX, USA

**Jian Huang** Shanghai Jiao Tong University, Shanghai, China

**Hao Jiang** National University of Singapore, Singapore, Singapore

**Yan Jin** University of Southern California, Los Angeles, CA, USA

**Iestyn Jowers** Technische Universität München, Garching, Germany

**Julie Jupp** University of Technology Sydney, Sydney, Australia

**Udo Kannengiesser** Krasnow Institute for Advanced Study, Fairfax, USA

**Maryam Khorshidi** Arizona State University, Tempe, AZ, USA

**Corinna Königseder** Technical University of Munich, Munich, Germany

**Kenneth Kotovsky** Carnegie Mellon University, Pittsburgh, PA, USA

**Gul Okudan Kremer** Pennsylvania State University, Philadelphia, PA, USA

**Gül Kremer** Pennsylvania State University, University Park, PA, USA

**Djordje Krstic** Hetzel Design Inc., Calabasas, CA, USA

**Mukund Kumar** University of Texas at Austin, Austin, USA

**Pat Langley** Arizona State University, Tempe, AZ, USA

**Ju Hyun Lee** University of Newcastle, Newcastle, NSW, Australia

**Chun-Yu Lin** Pennsylvania State University, Philadelphia, PA, USA

**Julie S. Linsey** Texas A&M University, College Station, TX, USA

**Fernán Acevedo López** Universidad Politécnica de Valencia, Valencia, Spain

**Wolfgang Maass** Saarland University, Saarbrücken, Germany

**Chris Maclellan** Arizona State University, Tempe, AZ, USA

**Jorge Alcaide Marzal** Universidad Politécnica de Valencia, Valencia, Spain

**Daniel A. McAdams** Texas A&M University, College Station, TX, USA

**Scarlett Miller** Pennsylvania State University, University Park, PA, USA

**Manikandan Mohan** Arizona State University, Tempe, AZ, USA

**Sumit Narsale** Arizona State University, Tempe, AZ, USA

**Alan Penn** University College London, London, UK

**Morteza Pourmohamadi** The University of Sydney, Sydney, Australia

**Franck Pourroy** Grenoble-INP, Grenoble, France

**Shilpa Ranade** IDC, Indian Institute of Technology Bombay, Mumbai, India

**Filippo A. Salustri** Ryerson University, Toronto, ON, Canada

**Somwrita Sarkar** University of Sydney, Sydney, NSW, Australia

**Kinda Al Sayed** University College London, London, UK

**Linda C. Schmidt** University of Maryland, College Park, MD, USA

**Chiradeep Sen** Oregon State University, Corvallis, OR, USA

**Jami J. Shah** Mechanical Engineering, Arizona State University, Tempe, AZ, USA

**Kristina Shea** Technical University of Munich, Munich, Germany

**Sue Sherratt** University of Newcastle, Newcastle, NSW, Australia

**L. H. Shu** University of Toronto, Toronto, Canada

**Joshua D. Summers** Clemson University, Clemson, SC, USA

**Ming-Xi Tang** The Hong Kong Polytechnic University, Kowloon, Hong Kong

**Simon de Timary** University College London, London, UK

**Christine Toh** Pennsylvania State University, University Park, PA, USA

**Joanna Tsenn** Texas A&M University, College Station, TX, USA

**Matthew Wood** Carnegie Mellon University, Pittsburgh, PA, USA

**Youbai Xie** Shanghai Jiao Tong University, Shanghai, China

**Ching-Chiaun Yen** National University of Singapore, Singapore, Singapore

**Robert J. Youmans** George Mason University, Fairfax, VA, USA

# Part I
# Design by Analogy

# Analogical Problem Evolution in Biologically Inspired Design

**Michael E. Helms and Ashok K. Goel**

**Abstract** Conceptual design typically entails co-evolution of the design problem and the design solution: initial problem formulations lead to preliminary solutions; incremental changes in the proposed solution lead to new insights into the design problem, and so on. In this paper, we describe a complementary process: problem evolution using analogies to already existing design cases. In particular, we present a case study in the context of biologically inspired design that inspects the evolution of an ill-defined design problem from inception to conceptual design. This case study demonstrates three important aspects of problem evolution from inception: first, significant problem evolution may occur independent of the generation of a new design solution for that problem; second, existing solutions to related problems serve as analogies that influence the way in which the problem is formulated; and third, the use of existing solutions from different domains, for example from existing biological solutions to engineering design problems, generates value not only by offering both potentially innovative solutions but also by changing the formulation of the problem itself.

## Background, Motivation and Goals

Conceptual design typically is characterized in terms of evolution of both the design problem and the solution, often described as a process of co-evolution. This characterization distinguishes design from routine problem solving: in problem solving, the problem remains fixed and only the solution to the problem evolves; in design, the problem and the solution co-evolve. While aspects of problem solving are now understood well enough to be implemented in computers, the process of co-evolution of the problem and the solution in design is not yet understood equally well.

---

M. E. Helms (✉) · A. K. Goel
Georgia Institute of Technology, Atlanta, GA, USA
e-mail: mhelms3@gatech.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_1,

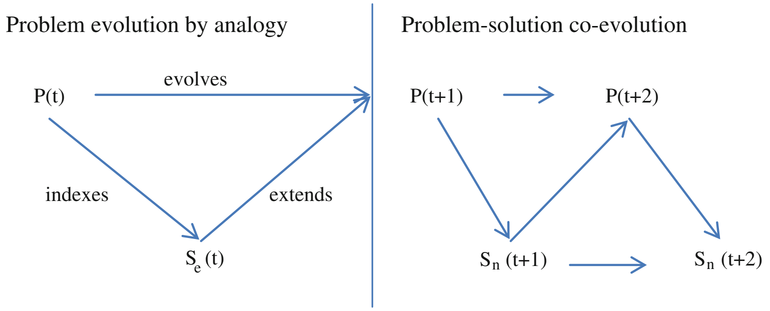


**Fig. 1** This figure contrasts our model of problem evolution by analogy (*left side* of the figure) with problem-solution co-evolution (*right side* of the figure, adapted from Maher and Tang). P represents a design problem, $S_e$ represents an existing analogical solution, $S_n$ represents a new design solution

Recently, Maher and Tang [1] and Dorst and Cross [2] have proposed computational models of the co-evolution of design problems and solutions. The right side of Fig. 1, which starts at time t + 1, illustrates Maher and Tang's model: A formulation of the design problem at time t + 1 focuses the search for a design solution. A design solution at time t + 1 may potentially lead to a revised problem formulation at time t + 2, which focuses the search for a new design solution, and so on.

We have observed a complementary process in our studies of biologically inspired design: evolution of design problems based on analogies to already known design cases. The left side of Fig. 1 illustrates this process: a formulation of a problem at time t, P(t), leads to an analogy to an already existing solution $S_e$(t) to a known problem. The existing solution then helps extend and expand the problem formulation into P(t + 1), which may result in the construction of a new solution or another analogy to another existing case. We call this process *analogical problem evolution*. Figure 1 as a whole indicates the process of problem–solution co-evolution including analogical problem evolution: at any step in the process, the formulation of the problem may lead to an analogy to an existing case or the generation of a new solution.

The context in which our observations are made, biologically inspired design (also known as biomimicry or bionics), espouses the use of biological systems as analogues for designing engineering systems [3–7]. In earlier work, we have reported on several findings from our studies of biologically inspired design: In Helms et al. [8] we described problem-driven design and solution-based design as two fundamental processes of biologically inspired design, and also presented a classification of design errors often made in the design processing. Similarly, in Helms et al. [9] we presented data indicating that the use of Structure-Behavior-Function models (SBF, [10]) of biological systems enhances reasoning as compared to textual descriptions as well as textual and diagrammatic representations of the systems. These findings informed the development of tools and techniques for teaching biologically inspired design [11–14]. Other researchers have followed similar research methodologies, coupling empirical studies with tool development (e.g. [15–17]).

Four basic questions in analogical design, including biologically inspired design, are *why*, *what*, *how* and *when* [18]: *Why* is knowledge transferred from a source case to a target problem; *What* knowledge is transferred; *How* is the knowledge transferred; *When* does the knowledge transfer occur. In Vattam et al. [12] we found not one, but several answers to the *why*, *what*, *how*, and *when* questions. In particular, we found that in biologically inspired design, biological designs are used not only for generating design ideas, but also for refining problem definitions, explaining proposed design concepts, and evaluating candidate design solutions. The work we describe here builds on this line of research. In particular, it examines the role of biological analogues in problem evolution from problem inception to conceptual design.

### *Tracking Problem Evolution*

To describe the process of *analogical problem evolution*, we need a scheme for consistently describing the design problem at a point in time so that we can identify the changes that occur over time. Most modern textbooks on design describe at least partial design problem representations (e.g. [19–21]) has noted, the scope and efficacy of the various problem representation schemes reflects the perspective adopted. In biologically inspired design, Dinar et al. [22] and Helms [23] have proposed two problem representation schemas. For this case study we use a simplified version of the problem schema developed in Helms [23] to represent the problem as a set of functions and constraints.

## Problem Schema

We use a problem schema that specifies design problem along two dimensions: (1) the functions desired of the artifact, and (2) the specifications and constraints of the artifact.

### *Functions Desired of the Artifact*

The literature on design contains several distinct and coexisting characterizations of a function. Erden et al. [24] review many functional descriptions in engineering. Borgo et al. [25] formally characterize several functional representation schemes. Carrara et al. [26] suggest that different notions of function coexist in the design literature, including function as the intended state or result of the system, function as a change to substances flowing through a system, and functions as actions the device must perform on an environment in the form of (subject, verb, noun) tuples. In this work, we adopt the last meaning of function above, where the subject is the to-be-designed artifact, the verb reflects the action in question, and the noun reflects the object on which the subject is acting, which we call function-object. In many cases, the noun is the same as the subject e.g. to move self, to clean self, etc. In other cases multiple function-objects may be required.

Functional hierarchy is prevalent in design theory [21, 27], representing a system/sub-system hierarchy in which the function of the sub-system contributes to the function of the larger system. Thus, system $S_1$ performing function $F_1$ is comprised of sub-systems $S_{1-1}$, $S_{1-2}$, … $S_{1-n,}$ which perform sub-functions $F_{1-1}$, $F_{1-2}$,…$F_{1-n.}$ Each sub-system can then recursively be defined by additional sub-systems and sub-functions. In this sense the functions can be seen as additive, or And-type conjunctions. In order to accomplish $F_1$, the system must perform all sub-functions $F_{1-1}$ And $F_{1-2}$ And…And $F_{1-n}$. In design thinking, designers also consider multiple alternative functions that can be represented as Or-type conjunctions. Chandrasekaran [28] provides an analysis of AND/OR function hierarchies, including the implications of such hierarchies for computational search.

## *Artifact Specifications and Constraints*

Artifact specifications designate properties and values, quantitative and qualitative, of the artifact being designed. Many artifact specifications are constraints, which designate properties and values of the designed artifact terms of inclusion or exclusion such as "must", "cannot", or "should". Other artifact specifications may denote options, which make explicit certain possibilities for properties and values that may be associated with the design artifact, expressed in terms like "could", "might", or "possibly" to name a few. Where a constraint is a statement such as "the design must use lightweight materials", an option may be "the design could use lightweight metal foam"; both talk about the properties of the material from which the artifact will be manufactured, however, one expresses an absolute condition to be met (albeit qualitatively), while the other provides an alternative to be considered.

Artifact specifications can also apply to either manufacturing or performance aspects of the problem. Additional sub-types include: time, shape, structure, material, energy, information, and cost.

## *Relationships Among Functions and Specifications in the Problem Schema*

Several kinds of relationships may exist among the functions and specifications. First, as described above, there may be function → sub-function relationships of both AND and OR types. Furthermore we commonly see function → specification relationships in which a particular function (e.g. propel self through air) implies certain constraints (e.g. material property must be lightweight).

### *Solution Schema*

Since in tracking problem evolution, we are also interested in describing the relationship between problem concepts and existing solutions, we also need a scheme for describing design solutions. Fortunately, there already exist many formal languages from which we may draw, including Functional Basis [29], SAPPhiRE [30], and SBF [10, 31]. In this study, we leverage SBF, a solution modeling schema already created and vetted in earlier work on biologically inspired design [11, 12, 32].

## Case Study

### *Study Context and Participants*

Each fall term since 2005, Georgia Tech's Center for Biologically Inspired Design has offered a senior-level, project-based interdisciplinary course in biologically inspired design (ME/ISyE/MSE/PTFe/BIOL 4740). Faculty members from Georgia Tech's Schools of Biology, Mechanical Engineering, and Industrial and Systems Engineering jointly teach the course. The course typically attracts 40–45 (mostly) undergraduate students every year. The class composition too is interdisciplinary: the 2009 class comprised of 15 biology students, 11 mechanical engineering students, and 13 students from a variety of academic disciplines.

The 2009 ME/ISyE/MSE/PTFe/BIOL 4740 course was structured into lectures, found object exercises, and a semester-long design project. The semester-long design projects group an interdisciplinary team of 4–6 students together based on similar interests. Instructors ensure that each team has at least one designer with a biology background and a few from engineering disciplines. Yen et al. [13, 14] describe the pedagogy in ME/ISyE/MSE/PTFe/BIOL 4740 in detail.

In 2009, each design team in the ME/ISyE/MSE/PTFe/BIOL 4740 class was tasked with a high-level problem related to building more sustainable homes. Topics included sensing, energy, environment, and resource management. Each team was asked to research their problem and design a solution based on one or more biological systems. Students were responsible for finding, understanding and applying biological systems relevant to their problem. Each team had one or more faculty as mentors who gave expert advice as and when needed. All teams presented their problem and initial design concepts during the middle of the term, then submitted final designs during the last two weeks of class along with a final design report.

The case study in this paper derives from one of the term-long design projects from the 2009 ME/ISyE/MSE/PTFe/BIOL 4740 class. The case was selected from an average performing but well functioning team as representative of a typical design project, with a straightforward design outcome. The design team was formed by the instructors based on student preferences, and consisted of one student each from biology, mechanical engineering, electrical engineering, math, and material science majors. The team was asked to focus on energy generation in the context of sustainable housing. The term-long design project led to a biologically inspired color changing cover for solar thermal water heaters to prevent overheating.

## *Protocol Study*

We analyze the case study using content-oriented protocol analysis [33–35]. Since our case study extends over a term-long trajectory, rather than verbal protocol transcripts, we use documents produced in the context of a class on biologically inspired design as our data source. These work documents are coded and analyzed according to the schema described previously. We refer to each coded element in the schema as a concept.

## *Data Gathered*

We gathered data at four stages to track the progression of the design problem over time. As part of their homework, each design team was required to provide their interpretation of the problem they were working on. The first homework assignment was due 2 days after the assignment of the problem topic. The design team had a single in-class discussion among themselves about their problem. Four of five students turned in a one- to two-page problem description. We used the text of the most comprehensive student description as our data at this stage.

The second stage was the midterm presentation delivered the students. Students were instructed in the midterm presentation to provide (1) an updated problem description, (2) five biological sources to serve as potential sources of inspiration, (3) to demonstrate their understanding of how each biological system worked, and (4) to show how they could apply each source to their problem. We used the design team's presentation slides and notes as the data at this stage.

The third stage occurred after students were provided feedback from instructors on their midterm presentation. The assignment was the same as the assignment used to collect the first data point; a one- to two-page text-only problem description submitted by each student. We used the text of the problem description of the same student as in the first assignment.

The fourth data point was the final presentation made by the design team. The assignment included the same elements as the midterm presentation, with the addition of a description of the final design and a qualitative analysis demonstrating the viability of the new design. Again, we used the presentation slides and notes as the data from this stage.

## *Analytical Methodology*

Only the text content, including bullet points, formulae, tables and text annotations, from each of the four design documents were considered in this study. Problem descriptions were provided in text only format, and were structured in complete sentences. Presentation material contained less structured text, including tables, bullet points, formulae and tables.

Text was divided into phrases, each of which encapsulated a single schema concept. Some concepts, such as referencing a biological or existing man-made solution, are short and straightforward, such as “the desert snail.” Other concepts such as “so that it is cooler within the shell than the outside air and ground” are more verbose, but encapsulate essentially a single concept: the degree to which the function “cool” must perform.

Relationships were inferred directly from text. If a solution concept was mentioned e.g. “the desert snail cools itself” with respect to a relevant problem concept e.g. “the designed system must cool itself”, the solution was tagged to the problem concept. In this case we say the solution “desert snail” is related to the problem concept, the function “system cools itself”. Another non-biological example is the phrase “we typically think of voltaic cells creating current”, in which voltaic cells are an existing solution, and creating current is a function of that solution (the phrase “we typically think of” is a meta-level design phrase, which is ignored for this analysis.)

As an example of the encoding, take the following text:

> The snail shell structure is stand alone and has the ability to passively dissipate heat by using the heat gradient so that it is cooler within the shell than outside the air and ground. This would be helpful for allowing the interior of a structure with solar panels to remain cool. Currently solar panels are rigid and typically pretty sensitive.

In Table 1 we provide a representative sample of text and the breakdown and coding for it. The details of the notation used for coding is not of much importance here, but should provide the reader with a firm grasp of the protocol used. Some cases of encoding text are ambiguous. For example, in Table 1 it is not clear that “stand alone” is indeed a function. This term has potentially many implications for additional specifications and functions. However, lacking explicit elaboration, we make our best guess about the designer‘s intent. The total number of concepts encoded for each of the four data points was stable, varying between 106 to 124 total concepts, with a total of 466 concepts encoded among the four stages. However, the types of concepts and the concepts themselves changed significantly from one state of processing to another.

**Table 1** Sample encoding from problem description 2

| Text | Comment | Encoding |
|---|---|---|
| The snail shell structure | A biological solution | Solution (biological): snail shell |
| It stands alone | Function of snail shell | Function: stand alone |
| And has the ability to passively | Modifies the function dissipate heat | * |
| Dissipate heat | Function of snail shell | Function: dissipate heat |
| By using the heat gradient | Principle applied to dissipate head | Solution principle: use heat gradient |
| So that it is cooler within the shell than the outside air and ground | Describes the degree to which cooling occurs | * |
| This would be helpful | Meta-Comment | * |
| For allowing | Function allow…to remain cool | Function: keeping cool |
| The interior of a structure | A location inside of the solar panels | * |
| With solar panels | An existing solution | Solution (existing): solar panels |
| To remain cool | Function: allow…to remain cool | Function: keeping cool |
| Currently solar panels | Same existing solution | Solution (existing: solar panels |
| Are rigid | A perceived deficiency | * |
| And typically pretty sensitive | A perceived deficiency | * |

*Neither function, nor specification/constraint. See future research section and reference [20] for schemas in development of more complete encoding

Of the 466 concepts, 35 were not directly related to the design per se, for example the meta-comment "this would be helpful" listed in Table 1. Of the remaining 431 concepts, 24 concepts could not be clearly encoded as a function, a specification/constraint, or neither. This accounts for roughly 5 % of the total number of concepts that were encoded. We consider an ambiguous encoding as any encoding that could reasonably be considered in one or more encoding categories. All encoding was conducted by the first author of this paper and the definition of whether a single encoding was ambiguous or not was at the author's discretion. Ambiguous concepts are included in the analysis that follows, categorized as the single concept most relevant in the author's opinion. Some concepts such as perceived deficiencies and performance criteria, were not represented in this encoding schema.

Figure 2 provides a complete visual representation of the encoding of a single design document (problem description 2). One can see from the visual representation the large number of functions relative to other concepts. Solid lines represent function/sub-function or system/sub-system relationships. Dotted lines represent relationships between either existing solutions, new solution concepts, or specifications/constraints and functions. Existing solutions with an asterisk (*) are biological.

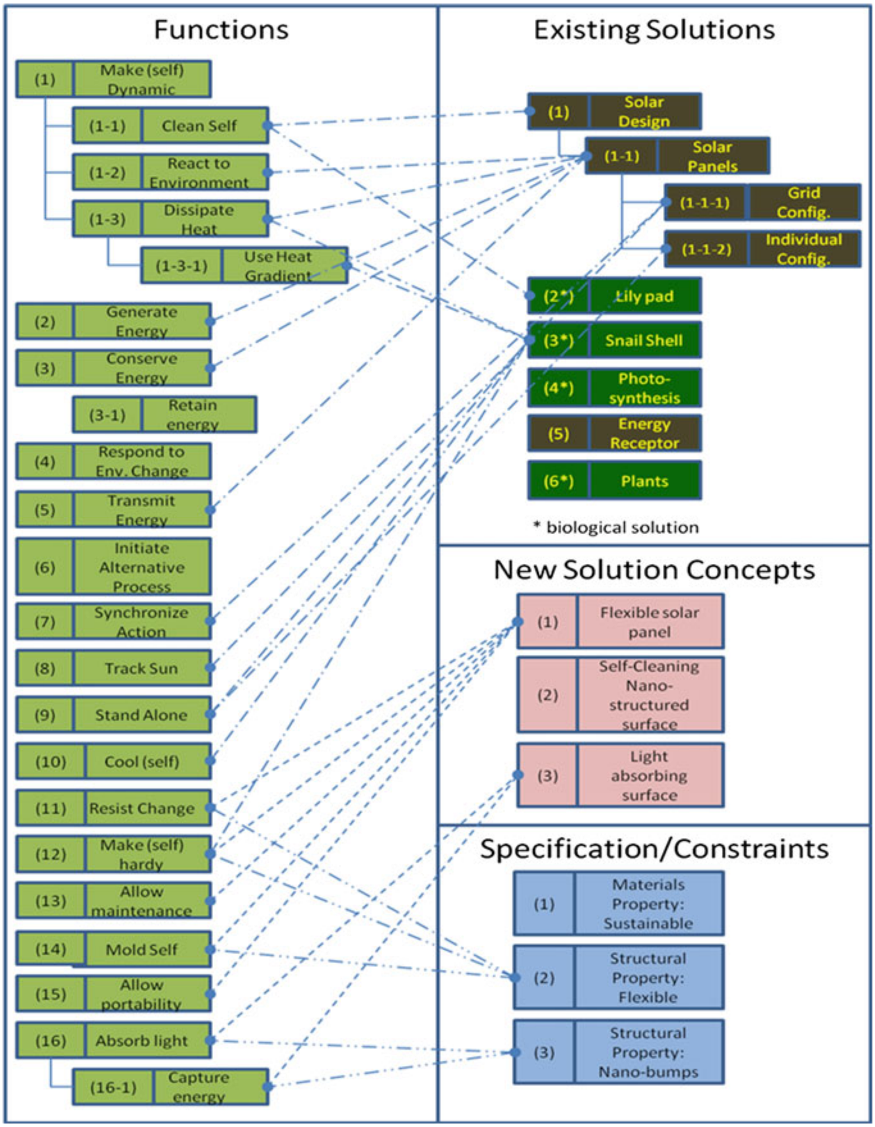


**Fig. 2** Visual representation of complete coding of problem description 2

## Data

We first present a descriptive account of how the design unfolded, followed by a quantitative description of the design documents, using the language of the problem schema. Both the descriptive account and quantitative analysis focus on the analogical problem evolution.

## Descriptive Summary

The team began with the open-ended problem of sustainably generating energy for a house. After an initial meeting, the problem description document identified a range of types of sustainable energy—wind, solar, water, geothermal—discussing solutions such as wind turbines, photovoltaic cells, towers of liquid sodium heated through reflected light, chemical batteries, and storage of energy for later use using compressed air. The document also mentioned fat as a means of storing energy in biology. Cost was highlighted as a salient constraint on their design. The document discussed different places in which the current technologies were used: from coastal areas, to farms and cities; they also discussed relevant weather conditions, such as the amount of wind or sun, and extreme weather conditions. Performance characteristics were universally vague, of the character "more efficient" or "costs less." Cost was the only constraint discussed but only in vague terms, e.g., noting that cost is a consideration.

The midterm documents discussed existing technological solutions of photovoltaic cells and coal plants. A wide range of biological sources were considered, including the desert snail, diatoms, photosynthesis, enzyme reactions, and the lotus leaf. Descriptions of the relevant functions of each biological source were provided; for example, the function of the desert snail is heat dissipation, which is performed by the structure of its shell. The midterm documents proposed solution-modifications to the photovoltaic cell, derived from each of these biological solutions. Thus, in the case of the self-cleaning lotus leaf, the documents proposed a self-cleaning photovoltaic cell. Solution proposals were little deeper than a function-structure pairing, none of which were (directly) developed further.

In the midterm documents, we noticed the addition of new functions, cleaning-self and dissipating heat, directly associated with biological solutions, and the dropping of other heat related functions, such as storing and directing heat, as the mirror/heat tower was dropped from the discussion. The environment, desert, from the mirror/heat tower solution remains in place, and is also related to the desert snail. Furthermore, we note the addition of the criteria "passively" connected to both functions attached to the biological solution. Manufacturing also is a rising concern, as the ability to reproduce materials and effects is highlighted.

At the third stage, the problem description continues its focus on solar panels and photovoltaic cells, and with all of the biological sources mentioned previously, except diatoms, which appear to have been dropped. Heat dissipation is discussed, but the design team now focuses on flexible, moldable and self-cleaning surfaces, derived again from the lotus leaf, and on a newfound perceived deficiency in current solar panels—rigidity. The environment under consideration shifted from a desert focus to an environment with greater temperature range, as well as the need to physically connect their solution to a home. Manufacturing nano-scale materials is again a manufacturing constraint, as well as the need for materials to be sustainable. There is also a shift in the problem focus from passive response to increased efficiency.

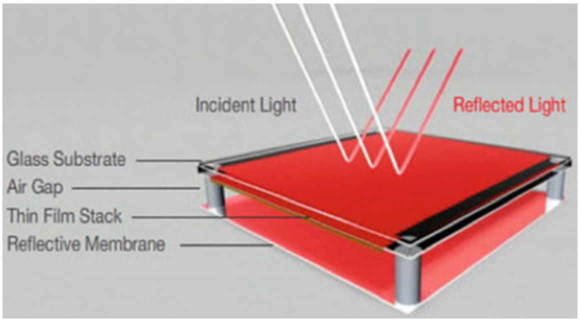


**Fig. 3** Final design concept rendering

In the final design, the design team arrives at its first instantiated solution (in terms of conceptual rendering, shown in Fig. 3), which focuses on the functions of regulation and cooling, rather than self-cleaning and flexibility discussed heavily in the previous problem description.

The design team appears to have radically changed the problem, moving from photovoltaics to solar thermal collectors for water heating, which run the risk of overheating and damaging their internal structure. The new solution proposed is a dynamic feedback regulation mechanism from the enzymes discussed in the midterm, which is combined with a mechanism from a newly introduced biological organism, the tortoise beetle, which uses its color changing shell for camouflage. The designers intend to use a mechanism similar to that used by the shell of the tortoise beetle to alter the color of the thermal collectors to change the amount of heat captured, depending on the internal heat of the unit.

In this case study, the designers shift from the problem concepts of a self-cleaning, flexible solar panel to the concepts of self-regulating, cooling thermal solar collectors. While this appears to be a significant shift in the problem, we can see the incremental nature of the process. Using and managing heat has been embedded in the teams thinking all along, from the mirror/heat tower, to the desert snail, to the environment of the desert, to the concept of dynamically responding to the environment. These concepts were accreted into the designers' problem schema from references to a number of solutions that were investigated along the way. When a new problem concept arose—overheating—the team was able to quickly pivot to the new problem focus and come up with a dramatic, creative solution.

## *Quantitative Summary*

After coding the design documents, we analyzed the concepts represented in the problem schema, as well as references made to existing manmade solutions, biological solutions and new solutions. Table 2 shows summary statistics for function and specification/constraint concepts for each of the four stages, as well

**Table 2** Summary statistics, number of unique concept instances

| Table | PD1 | Midterm | PD2 | Final |
|---|---|---|---|---|
| Functions | 25 | 20 | 20 | 9 |
| Specifications/constraints | 1 | 6 | 3 | 4 |
| Man-made solutions | 7 | 3 | 5 | 3 |
| Biological solutions | 2 | 5 | 4 | 2 |
| New (conjectured) solutions | 1 | 7 | 3 | 1 |

as the number of solutions at each stage. At this first level of description, some things already stand out. First, the number of functions considered at each stage remains between 20 and 25 until the final design, where it drops to 9. This seems to suggest that the designers were open to many possible combinations of functions for accomplishing their design objective, until the final design was instantiated. Second, the number of specifications/constraints is relatively very low, never more than 6. Of the 14 total, four were cost-related and three were sustainable materials related. Again, this seems to suggest that the designers wanted to maintain an open problem description as long as possible.

We observe that after a strong emphasis on man-made solutions in the initial stage, existing solutions references are rather evenly split in the next three stages, trending down slightly in the final stage. The number of new solutions discussed moves from one (the level of specificity for which was literally, "the new solution"), to seven—an explosion of independent solution ideas—back to a single final new solution in the end. While the trends in solution generation are not particularly surprising, we find the fact that designers consistently reference about the same number of existing solutions throughout the design cycle curious.

With respect to the number of functions, Table 3 considers the follow-through of each function from one stage to another. That is to say, did functions mentioned in earlier problem statements carry through to future problem statements? We see that from the initial generation of 25 functions, three of those functions carry forward into the Midterm, seven are considered in problem description 2 (PD2), and two ("generate energy" and "capture energy") follow through to the final. Likewise 17 new functions appear in the Midterm description, one of which ("adjust flow") appears in the final design. Fewer new functions appear in the third stage, just 10; of which 1 ("keep cool") makes it into the final model. In the final stage, there are more new functions than old. Five new functions appear in the final problem description document, while four have been carried through from previous descriptions. This alone tells a very interesting story. In this ill-defined design problem context, we see a great deal of exploration. Fifty-seven unique functions are considered, only nine of which eventually make it into the final solution. Over 80 % of the functions considered are discarded along the way.

**Table 3** Function concept carry over

| PD1 | Midterm | PD2 | Final |
|---|---|---|---|
| 25 new functions added | +17 new functions added | +10 new functions added | +5 new functions added |
| | →3 carried over from PD1 | →3 carried over from Midterm | →1 carried over from PD2 |
| | | →7 carried over from PD1 | →1 carried over from Midterm |
| | | | →2 carried over from PD1 |
| 25 total | 20 total | 20 total | 9 total |

## *Solution Relationship Data*

In this paper we will consider only one more level of detail in the data; the relationship of solutions to the concepts in the problem model, Tables 3, 4 and 5.

For any concept in the problem model (function or specification/constraint), that concept may be associated with: (1) an existing solution, either man-made or biological, (2) a new solution, or (3) not associated with another solution. Tables 3, 4 and 5 show for each stage the numbers of function and artifact specifications respectively and whether they are associated with (1) an existing solution, (2) a biological solution, or (3) no solution. We note that the numbers in these tables may sum to be greater than the total number of concepts reported in Tables 2 and 3, as each concept may be associated with one or more solution category. For example if both a biological and a non-biological solution were mentioned with respect to a particular function, such as reflect light, we would get two tallies for that concept. Likewise multiple solutions in the same category could reference the same concept, for example two separate manmade solutions may have mentioned light reflection.

Table 4 shows the number of function concepts in the problem statement that referenced an existing manmade solution, a biological solution or had no reference to any existing solution. This table shows an interesting trend that provides insight into the process of biologically inspired design. Table 4 shows that the design team initially conceptualizes functions in their problem description largely (18 out of 29 references) in terms of existing manmade solutions. In the midterm stage, functions in the problem description are largely (12 out of 22) referenced in relation to biological solutions. This suggests that designers are re-conceptualizing their problems at least in part by identifying and transferring potentially useful functional concepts from biological solutions to their problem model.

In the third stage, manmade and biological solution transfers are roughly equivalent, while in the final stage, the point of design instantiation, about half of the functions discussed in the final problem description originated from existing biological solutions. In total, 65 % (60 of 92) of function concepts can be attributed to existing (man-made or biological) solutions.

**Table 4** Function concepts by solution reference

| | PD1 | Midterm | PD2 | Final |
|---|---|---|---|---|
| Man made | 18 | 4 | 10 | 4 |
| Biological | 0 | 12 | 6 | 6 |
| No reference | 11 | 6 | 12 | 3 |
| Total | 29 | 22 | 28 | 13 |

**Table 5** Specification/constraint concepts by solution reference

| | PD1 | Midterm | PD2 | Final |
|---|---|---|---|---|
| Man made | 0 | 0 | 0 | 0 |
| Biological | 0 | 0 | 0 | 0 |
| No reference | 1 | 6 | 3 | 4 |
| Total | 1 | 6 | 3 | 4 |

The trend for specification/constraint in Table 5 with respect to existing and biological solutions is clear. Problem specifications and constraints, for example "must use sustainable materials," are not associated with, at least not explicitly in this case study, other solutions. Many of the specifications were with regard to cost and sustainable materials, which were likely inferred from the design context of "sustainable housing."

## Summary of Analysis

Our analysis of the above data suggests that the design team broadly explored different aspects of the problem description; committing to few concepts rigidly, holding open possibilities until the right confluence of problem description and the descriptions of existing solutions emerge to form a cohesive pair. This finding is similar to Dorst and Cross [2] with one major difference: as opposed to generating early solutions to problems as they are formulated, our designers employed other, in particular *analogical*, strategies to generate problem concepts and enrich their problem descriptions. Our study is quite clear on this point; using analogies to existing design cases is a powerful way to formulate the design problem.

Designers appear to tentatively adopt problem aspects from existing solutions, in particular functional aspects, temporarily appending them to an overall problem description. What makes some concepts stay while others are abandoned is not yet clear from our data. We speculate that the early function-solution pairs seen in the design trajectory were evaluated and abandoned for lack of knowledge or manufacturing know-how. This suggests an evaluation-pruning function early in the design process. Another point is that in addition to solution analogy and solution evaluation, other methods are clearly at work enhancing the problem description.

Analogical transfer accounts for at most 65 % of the new functions seen in this example, and for none of the specification/constraints. Analogical transfer seems to be limited in this case study to only certain classes of concepts.

## *Current and Future Research*

This paper uses a simple problem schema to show the relationship between existing solutions and design problem conceptualization. To increase the value and reliability of these studies, we are currently validating a more robust problem schema and coding methodology using 37 problem description instances and standard inter-rater reliability. In addition, this new schema is being used to more thoroughly analyze a number of additional, semester-long case studies of biologically inspired design. In Fall 2011, the problem schema and theory developed here were deployed as a new pedagogical tool in the ME/ISyE/MSE/PTFe/BIOL 4740 class. Further, much as previous empirical findings [8, 12] informed the development of an interactive design environment called DANE for supporting aspects of biologically inspired design [11] this new schema is being used as the basis for an interactive tool that assists with problem evolution, analogy identification and evaluation, and solution generation. Helms [23] provides an initial outline of the new interactive tool.

## Conclusions

The development of both design pedagogy and design technology depends on our understanding of design problems, products and processes. In this paper, we analyzed the process of the evolution of a problem in biologically inspired design from its inception through conceptual design. We draw two main conclusions from this work, one from the perspective of biologically inspired design and the other from the perspective of problem evolution in design in general. With respect to *why* and *when* analogies are used in biologically inspired design, we found that analogies are used for identifying, formulating, and transforming design problems very earlier in the design process. In addition, we have a partial answer to *what* is transferred; in this case study, we found that functions were transferred from biological designs to engineering problems, but specifications and constraints were not. Secondly, in the more general context of design as a whole, evolution of design problems typically is viewed as a co-evolution of design problems and solutions. Our analysis of the case study of design in this work suggests that significant problem evolution may occur independent of the generation of a new design solution for that problem, and that existing solutions to related problems serve as analogies that influence the way in which the problem is formulated.

**Acknowledgments** We thank the instructors of the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course, especially its coordinator Professor Jeannette Yen. We also thank the student designers in the class for their input in this study. This paper has benefited from many discussions with Bryan Wiltgen, Swaroop Vattam, and Jeannette Yen. This research was supported by a US National Science Foundation CreativeIT grant (#0855916) entitled "Computational Tools for Enhancing Creativity in Biologically Inspired Engineering Design."

## References

1. Maher ML, Tang H (2003) Co-evolution as a computational and cognitive model of design. Res Eng Design 14(1):47–64
2. Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem-solution. Des Stud 22(5):425–437
3. Bar-Cohen Y (ed) (2011) Biomimetics: nature-based innovation. CRC Press, Boca Raton
4. Benyus J (1997) Biomimicry: innovation inspired by nature. William Morrow, USA
5. Chakrabarti A, Shu L (2010) Biologically inspired design. AIEDAM 23:453–454
6. Vincent J, Mann D (2002) Systematic technology transfer from biology to engineering. Phil Trans Royal Soc London 360:156–173
7. Yen J, Weissburg M (2007) Perspectives on biologically inspired design: introduction to the collected contributions. J Bioinspiration Biomimetics 60(2):673
8. Helms M, Vattam S, Goel A (2009) Biologically inspired design: product and processes. Des Stud 30(5):606–622
9. Helms M, Vattam S, Goel A (2010) The effects of functional modeling on understanding complex biological systems. In: Proceedings of ASME IDETC/CIE conference on design theory and methods, Montreal, Canada, pp 107–115
10. Goel A, Rugaber S, Vattam S (2009) Structure, behavior and function of complex systems: the SBF modeling language. AIEDAM 23:23–35
11. Goel A, Vattam S, Wiltgen B, Helms M (2011) Cognitive, collaborative, conceptual and creative—four characteristics of the next generation of knowledge-based CAD systems: a study in biologically inspired design. Comput Aided Des 44(10):879–900
12. Vattam S, Helms M, Goel A (2010) A content account of creative analogies in biologically inspired design. AIEDAM 24:467–481
13. Yen J, Helms M, Goel A, Vattam S (2010) Evaluating biological systems for their potential in engineering design. Adv Nat Sci 3(2):27–40
14. Yen J, Weissburg M, Helms M, Goel A (2011) Biologically inspired design: a tool for interdisciplinary education. In: Bar-Cohen Y (ed) Biomimetics: nature-based innovation. Taylor and Francis, London
15. Cheong H, Chiu I, Shu L, Stone R, McAdams D (2011) Biologically meaningful keywords for functional terms of the functional basis. J Mech Des 133:021007
16. Mak T, Shu L (2008) Using descriptions of biological phenomena for idea generation. Res Eng Design 19(1):21–28
17. Shu L (2010) A natural-language approach to biomimetic design. AIEDAM 24(4):507–519
18. Goel A (1997) Design, analogy and creativity. IEEE Intell Syst 12(3):62–70
19. Dym C (1994) Engineering design: a synthesis of views. Cambridge University Press, Cambridge
20. French M (1999) Conceptual design for engineers, 3rd edn. Springer, Berlin
21. Pahl G, Beitz W (1996) Engineering design: a systematic approach. English edition. In: Wallace K (ed) 2nd edn. Springer, Berlin
22. Dinar M, Shah J, Hunt G, Campana F, Langley P (2011) Towards a formal representation model of problem formulation in design. In: Proceedings of ASME IDETC/CIE conference, Washington DC, pp 263–272

23. Helms M (2011) Solution based problem evolution and problem inception in biologically inspired design. Technical report, GVU, Georgia Institute of Technology, GIT-GVU-11-10
24. Erden M, Komoto H, van Beek T, D'Amelio V, Echavarria E, Tomiyama T (2008) A review of function modeling: approaches and applications. AIEDAM 22(2):147–169
25. Borgo S, Carrara M, Garbacz P, Vermaas P (2009) A formal ontological perspective on the behaviors and functions of technical artifacts. AIEDAM 23:3–21
26. Carrara M, Garbacz P, Vermaas P (2011) If engineering function is a family resemblance concept: assessing three formalization strategies. Appl Ontol 6(2):141–163
27. Simon H (1996) Sciences of the artificial, 3rd edn. MIT Press, Cambridge
28. Chandrasekaran B (1990) Design problem solving: a task analysis. AI Magazine 11(4):59–71
29. Stone R, Wood K (2000) Development of a functional basis for design. J Mech Des 122(4):359–370
30. Srinivasan V, Chakrabarti A (2011) An integrated model of designing. J Comput Inf Sci Eng 10(3):31013
31. Chandrasekaran B, Goel A, Iwasaki Y (1993) Functional representation as design rationale. IEEE Comput 48–56
32. Vattam S, Wiltgen B, Helms M, Goel A, Yen J (2010) DANE: fostering creativity in and through biologically inspired design. In Proceedings of 1st international conference on design creativity, Nov 2010, Kobe, Japan, pp 115–122
33. Dorst K, Dijkhuis J (1995) Comparing paradigms for describing design activity. Des Stud 16(2):261–274
34. Gero J, McNeil T (1998) An approach to the analysis of design protocols. Des Stud 19:21–61
35. Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Des Stud 19:455–483

# Understanding Analogical Reasoning in Biomimetic Design: An Inductive Approach

**Hyunmin Cheong, Gregory Hallihan and L. H. Shu**

**Abstract** This paper reports insights gained from observing groups of novice designers apply biological analogies to solve design problems. We recorded the discourse of fourth-year mechanical engineering students during biomimetic design sessions. We observed that the availability of associations from superficial or functional characteristics of biological knowledge led to fixation, which affected the designers' ability to identify the relevant analogy. In addition, even after identifying the analogy, the designers fixated on mapping irrelevant characteristics of biological knowledge, instead of developing additional solutions based on the previously detected analogy. The paper also presents initial work towards quantifying analogical reasoning in a design study.

## Introduction

Analogical reasoning involves the comparison of similarities between two concepts. Abstracting and transferring knowledge from one concept to another allows designers to develop novel design concepts. Design researchers, e.g. Goel [1] agree that analogical reasoning plays a key role in creative design.

In biomimetic or biologically inspired design, designers use analogical reasoning to compare similarities between biological phenomena and design problems, and then transfer analogous strategies to develop design solutions. Shu et al. [2] observed that although several innovative solutions to engineering problems have been inspired by biological phenomena, challenges still exist in developing generalized methodologies for biomimetic design. In particular, a number of

H. Cheong · G. Hallihan · L. H. Shu (✉)
University of Toronto, Toronto, Canada
e-mail: shu@mie.utoronto.ca

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_2,

obstacles prevent novice designers from correctly applying biological analogies, and effective methodologies that overcome these obstacles are still being developed.

We believe that analogical reasoning in the context of biomimetic design is still not fully understood. Therefore, our research goal is to gain a better understanding of the analogical reasoning process during biomimetic design. We used an inductive approach and observed groups of novice designers, working in a natural setting, apply biological analogies to solve design problems. The designers' dialogues were recorded and analyzed qualitatively.

The following sections provide background in biomimetic design and analogical reasoning. In addition, previous observational studies and protocol analyses in design research are reviewed to preface our methodology.

## *Relevant Work in Biomimetic Design*

Mak and Shu [3] studied cognitive factors that influence the application of biological analogies to engineering problems. The authors observed that text descriptions of biological phenomena that included principles and behaviors in addition to forms, tended to be more easily used by students as design stimuli. In later work, Mak and Shu [4] found that novice designers tend to fixate on irrelevant features of biological phenomena and incorrectly apply biological strategies to design problems.

Cheong and Shu [5] observed that text descriptions of biological phenomena containing causal relations are more likely to serve as useful analogies for design problems. Causal relations often explain how functions are achieved by behaviors. For example, "break down" enables "absorb" in the description "Humans absorb amino acids by breaking down proteins from food". Cheong et al. [6] developed a template to help designers extract strategies from causal relations contained in descriptions of biological phenomena. However, when novice designers used the template in a controlled experiment, the correctness of analogical transfer only improved marginally.

Vattam et al. [7] and Helms et al. [8] studied the cognitive account of biomimetic design in the context of students working on projects in a biologically inspired design course. Helms et al. reported a number of common errors made by designers, including solution fixation, misapplied analogy, and improper analogical transfer. Vattam et al. developed a conceptual framework of compound analogical design that extends existing models of analogy-based design to better represent biologically inspired design, and studied the distribution of analogies across different design phases. Both Helms et al. and Vattam et al. focused on understanding the process of biologically inspired design through an in situ study on the practices of novice designers. While their research had a broad context to observe novice designers' work over the course of a term project, we aim to identify insights through detailed analyses of designers' dialogues during 20-min design sessions.

Other previous research in biomimetic design has focused on developing models to support the access and use of biological information. The SBF model from Goel et al. [9] represents causal processes between states using the structure-behavior-function framework. Helms et al. [10] observed that the SBF model of biological systems helped designers understand complex relations in systems, such as causality. Vattam et al. [11] reported that DANE, a library of SBF models of biological systems, could potentially be used as a conceptualization tool.

Sartori et al. [12] used SAPPHiRE constructs to represent mechanisms of transfer in 20 biomimetic examples in the literature. SAPPHiRE, developed by Chakrabarti et al. [13], defines multiple levels of abstraction in order to explain how a biological system works to fulfill its goals. The authors found that successful biomimetic examples usually involve systems that share similarities at higher levels of abstraction.

Nagel and Stone [14] developed a framework that is primarily based on functional-modeling of biological systems with a set of terms from the "engineering-to-biology thesaurus". Although the authors provide a detailed description for using their technique, they do not empirically study its direct benefits to designers, or how designers use it in practice.

The above biomimetic design models, primarily developed to represent and index biological information, are effective at formally representing complex biological systems. However, their utility in the concept generation process requires further validation. For instance, Vattam et al. [11] reported the challenges of using SBF modeling in concept generation; novice designers were not willing to build models without seeing the direct benefits and were not convinced of DANE's usefulness and value.

We propose that better understanding of the cognitive processes in biomimetic design can help improve these models and corresponding heuristics, and ultimately lead to more effective biomimetic concept generation. In the following section, we discuss background research in analogical reasoning, which is fundamental to biomimetic concept generation.

## *Background in Analogical Reasoning*

Analogical reasoning is considered to be central to creative thought. For instance, Boden [15] claims that the creation of novel ideas often involves the transformation of existing knowledge into something new. In design, analogical reasoning allows individuals to find similarity between an existing knowledge base and a target design space, and transform that existing knowledge into new design solutions.

Gentner et al. [16] identify two levels at which similarities can be found in analogical reasoning: superficial and relational. The superficial level refers to object attributes. The relational level can be further decomposed into two levels: relation between objects and relation between relations, i.e., "higher-order relation".

In the context of biomimetic design, the superficial level corresponds to the attributes of biological entities (objects). The relation between objects then corresponds to the functions of biological entities, and the relation between relations can correspond to the causal or temporal relations between the functions of biological entities. The following example describes the different characteristic levels of enzymes.

| | |
|---|---|
| Superficial | Enzymes are *ribbon-shaped* |
| Functional | Enzymes *bind* to substrates |
| Causal | Enzymes *bind* to substrates to *form* enzyme-substrate complexes. |

Many researchers agree that successful analogical transfer occurs at the relational levels. Gentner et al. [16] note that finding similarities between higher-order relations is crucial to successful analogical reasoning. In the context of design, Goel [1] states "analogical transfer requires the use of generic abstractions, where the abstractions typically express the structure of relationships between generic types of objects and processes". In biomimetic design, designers must abstract biological knowledge to identify its relational similarities to design solutions.

## *Observational and Protocol Studies*

While experimental studies test the validity of hypotheses or interventions, observational studies are well suited to formulate hypotheses and develop interventions for future experiments. Dunbar [17] notes an important benefit of an observational study is that researchers can observe more natural and real-world behaviors of people, whereas those behaviors may be restricted in experimental studies. The use of observational studies in biomimetic design [3, 4, 7, 8] include our past work, and Vattam et al. [7] and Helms et al. [8], who observed students working in natural settings on a biologically inspired design project over an extended period.

However, one limitation of observational studies is the difficulty of collecting data. For this reason, many researchers use "think-aloud" techniques to elicit verbal dialogues from participants. Encouraging designers to verbalize their thoughts is presumed to reflect the designers' underlying cognition. The verbal dialogues are then transcribed to generate protocols, which offer useful data for both qualitative and quantitative analyses. Cross [18] discusses a number of observational protocol studies in design research, including their advantages and limitations.

Chiu and Shu [19] reported some limitations of verbal protocol studies. Verbalizing thought processes can be perceived as unnatural and adds a cognitive workload on designers, which can lead to results that may not reflect real-world performance. To address this, designers can be encouraged to participate in design processes naturally, speaking aloud to one another as they normally would. While this approach may not capture cognitive mechanisms in as much detail, the process is more natural and may better reflect actual design practices.

## *Protocol Analysis*

Once verbal protocols have been generated they can be analyzed qualitatively and quantitatively. Merriam [20] recommends that the analysis of qualitative data, such as design protocols, should ultimately be tailored towards the needs of the researcher. One of the most common methods of analyzing protocols in psychological and design research, and the most relevant to our approach, is qualitative coding. Qualitative coding segments a protocol based on categories of interest to the researcher.

Miles and Huberman [21] suggest that researchers should develop meaningful and clearly defined categories for coding. Goldschmidt [22] used design "moves", which identify ideas that transform the design situation and reflect the development of ideas. Kvan and Gao [23] adopted Schön's definition of design processes ("framing", "moving", and "reflecting"), in order to study the problem-framing process in design. Kan et al. [24] used a coding scheme based on the FBS (function-behavior-structure) ontology. Gero [25] suggests that the FBS coding scheme provides a common framework to represent design knowledge and allows consistency in protocol analysis.

Linkography, developed by Goldschmidt [22], is performed by linking related design "moves" and graphically represents protocol data. Linkography has been used to examine a wide variety of phenomena in design, including problem framing effects [23], visuo-spatial working memory load [26], and design fixation [27]. The analysis of linkographs has also progressed to include applying statistical models, cluster analysis [26], and entropy models [28]. These techniques help researchers quantify the relationships found in linkographs.

Computational linguistic methods are also used to analyze design protocols. Dong [29, 30] used latent semantic analysis to quantify coherent thinking and lexical chain analysis to evaluate concept formation in design teams. These computational linguistic models provide more objective and standardized ways to analyze protocols. However, Wang and Dong [31] point out that computational models still require the resource-intensive preparation of training data. They took a more efficient, yet sufficient approach, of using statistical patterns of relevant semantic features, e.g., keywords, to compute appraisals in design text.

Both the linkographic and computational linguistic approaches are used to quantify design protocols. These approaches help mitigate researcher bias and allow the application of various numerical/statistical analysis techniques. Our current research chooses instead to manually review protocols, as we learned that this process could lead to valuable insights. We agree with Brown's [32] observation that studying transformational creativity such as analogical reasoning can be challenging, and that we should "work upwards towards creativity," i.e., take an inductive, bottom-up approach. Therefore, we initially focused on using qualitative observation to better understand analogical reasoning in biomimetic design.

While the research approach was primarily qualitative, we also worked towards the quantitative/graphical representation of our design protocols. The results of the protocol analysis will be discussed after presenting the research method and qualitative observations.

## Methods

### *Participants*

The data for this experiment were collected from 30 engineering students (28 males and 2 females), during a design-by-analogy laboratory exercise in a fourth-year mechanical design course at the University of Toronto. All data collected came from students who consented to have their design session audio-recorded and to have the data used for research purposes.

### *Procedure*

The laboratory exercise required students to generate solutions for an engineering design problem by using a biological analogy as a source of inspiration. Three design problems were used, and each problem was paired with a description of a biological phenomenon as the source of analogy.

For practical reasons, three to four students were assigned to a group and each group worked on a single design problem. There were three laboratory stations with three groups at each station (see Table 1).

Each group was given 20 min to generate solutions for the design problem. One group (Group 9) used only 12 min and stated they could not generate any more solutions. At the beginning of each 20-min session, each member of the design group was provided with a written copy of the design problem and relevant biological phenomenon.

The order of problems was counterbalanced in a 3 × 3 *Latin square* matrix to control for problem effects. However, it is reasonable to expect the presence of a learning effect for the second and third design groups at each station, since they had the benefit of observing the preceding groups.

### *Design Problems and Biological Phenomena*

The following design problems and corresponding descriptions of biological phenomena were created by the researchers and provided to the design groups.

**Table 1** Details on experimental groups and design problems assigned

| Lab station | Design group # | # of students | Design problem |
|---|---|---|---|
| A | 1 | 4 | Promotional mailing |
| | 2 | 3 | Authorized disassembly |
| | 3 | 3 | Wet scrubber |
| B | 4 | 3 | Wet scrubber |
| | 5 | 4 | Promotional mailing |
| | 6 | 3 | Authorized disassembly |
| C | 7 | 3 | Authorized disassembly |
| | 8 | 4 | Wet scrubber |
| | 9 | 3 | Promotional mailing |

1. *Promotional Mailing Problem*

You are a marketing director for a credit card company. You are looking for an effective strategy to distribute sign-up promotional mailings within a city. You would like to distribute promotional mail to selected neighborhoods in the city so that a large proportion of the promotional mail actually results in people signing up. In other words, you don't want to waste resources on sending promotional mail to neighborhoods where people are not likely to sign up. Assuming that you don't have any demographic information of the city, how would you optimize the use of promotional mailings?

Biological Phenomenon (Ant): An ant colony can identify the shortest path between its nest and food source with the following strategy. Ants depart the colony to search randomly for food, laying down pheromones on the trail as they go. When an ant finds food, it follows its pheromone trail back to the nest, laying down another pheromone trail on the way. Pheromones have more time to dissipate on longer paths, and less time to dissipate on shorter paths. Shorter paths are also travelled more often relative to longer paths, so pheromones are laid down more frequently on shorter paths. Additional ants follow the strongest pheromone trails between the food source and the nest, further reinforcing the pheromone strength of the shortest path.

2. *Authorized Disassembly Problem—From Saitou* et al. [33]

Original equipment manufacturers (OEM's) want easy disassembly of their products to reduce disassembly cost and increase the net profit from reuse and recycling at product end of life. However, OEM's are also concerned with protecting high-value components from theft and access by competitors. How can you allow disassembly that is easy but only by those authorized? [33].

Biological Phenomenon (Enzymes): Enzymes are complex proteins that bind to specific substrates (molecules) and form enzyme-substrate complexes that perform biochemical activities. The specific binding is achieved when the active site of an enzyme geometrically matches its corresponding substrate. However, an enzyme changes its shape with environmental factors such as pH and temperature. This shape change alters the conformation of the enzyme's active site to the point where substrates can no longer fit, thereby disabling the function of the enzyme-substrate complex.

3. *Wet Scrubber Problem*

Wet scrubbers are air pollution control devices that remove pollutants from industrial exhaust systems. In conventional wet scrubbers, exhaust gas is brought into contact with a liquid solution that removes pollutants from the gas by dissolving or absorbing them into the liquid. The removal efficiency of pollutants is often improved by increasing the contact time or the contact area between the exhaust gas and the scrubber liquid solution. What other strategy could be used to increase the removal efficiency of wet scrubbers?

Biological Phenomenon (Penguins): Penguins are warm blooded yet keep their un-insulated feet at a temperature close to freezing to minimize heat transfer to the environment. The veins that carry cold blood from the feet back to the body are located closely to the arteries that carry warm blood from the body to the feet. The warm blood flows in the opposite direction as the cold blood, which allows the penguins to transfer the most heat to the cold blood. This reduces both the amount the returning blood can drop the core body temperature, and the amount of heat lost through the feet.

## Design Session Mediators

A research assistant was assigned to each laboratory station to facilitate and audio-record the design sessions. To control for any confounding effects introduced by the research assistants, they were provided with a script to handle potential questions from students, and were instructed not to contribute to the design process. The research assistants only interceded when design progress slowed or the students had settled on a design solution. After 20 min, the research assistants stopped the design session and provided the next group with the corresponding design problem.

## Design Protocols

Students in each design group were instructed to verbalize their ideas during the design process; these verbalizations were audio-recorded and transcribed for analysis. However, students were not asked to verbalize all of their thoughts. Because this was a group exercise and there was only one audio-recording device at each laboratory station, having a true talk-aloud experiment would have made transcribing the audio files very difficult.

**Table 2** Examples of each coding category for the wet scrubber problem

| Code | Example |
|---|---|
| Entity | "Veins have a lot of surface area so we can make sure that…I mean…the liquid we are using for the scrubbing, it can go through like really narrow pipes or whatever to increase the surface area" |
| Function | "We also did kind of blood circulation, 'cause uh, we are re-circulating [scrubber solution and exhaust gas]" |
| Strategy | "It says the opposite direction allows, like, most flow of gas exchange […] so make, I don't know, maybe we could make the […] liquid scrubber run in one direction, and […] gas run in the other direction. That increases the flow [exchange]" |

Two authors of this paper transcribed the audio files for each design group. After each transcript was generated, it was cross-reviewed by the other researcher to verify its accuracy. Some audio data was not interpretable, e.g., multiple designers speaking at once, designers murmuring very quietly, etc., and this data was excluded from further analysis.

## Protocol Coding

We coded participants' ideas that involved some type of comparison into three different categories:

| | |
|---|---|
| *Entity* | A comparison to superficial characteristics of entities of the biological phenomenon |
| *Function* | A comparison to functions of the biological phenomenon |
| *Strategy* | A comparison involving a higher-order relation (strategy) from the biological phenomenon. |

The method of coding design protocols into a set of defined categories is in line with other protocol analyses discussed previously in the introduction [23–25]. Creating mutually exclusive segments, however, was not possible for this coding scheme. Higher-level comparisons, such as the strategy level comparison, often invoke comparisons at the functional and superficial levels. In addition, segmenting the protocol based on participants' utterances or ideas was difficult due to multiple interruptions from other group members and many instances of incomplete ideas. To avoid bias in the segmentation and coding process, each protocol was segmented into 10-s units. This coding scheme allowed us to code occurrences of each type of similarity comparison and plot their occurrence over the time of the design protocol. Two of the authors individually coded the protocols, after which cases of disagreement were discussed until an agreement was reached. Table 2 shows examples of segments that contain each coding category.

**Table 3** Examples of analogous elements between the ant phenomenon and the promotional mailing problem at three levels of comparison

| Level of comparison | Ant phenomenon | Promotional mailing | Similarity |
|---|---|---|---|
| Strategy | Target food source based on feedback obtained from random travel | Target sign-ups based on feedback obtained from random mailing | ✔ |
| Functional | Traveling to food source | Sending out mail | × |
| Superficial | Food source | Sign-ups | × |

Only the strategy level of comparison features a high degree of similarity

## Qualitative Observations

We first drew qualitative observations from the protocols. While some of the observations agree with previous research in design and cognitive psychology, there were new insights that could contribute towards a better understanding of the analogical reasoning process in biomimetic design.

### *Detection of Analogies*

All three groups that worked on the promotional mailing problem were able to identify the relevant strategy from the ant phenomenon within the first 5 min of problem solving. However, two of the three groups that worked on the authorized disassembly problem could not identify the relevant strategy from the enzyme phenomenon in the 20-min period. This result was surprising. The promotional mailing problem required participants to detect an analogy that was mostly based on the similarity at the strategy level, with the analogous elements present at the functional level and the superficial level having little similarity (see Table 3). On the other hand, the authorized disassembly problem was paired with the enzyme phenomenon. The phenomenon featured analogous elements that may seem similar at all three levels, which could have helped participants identify the relevant analogy (see Table 4).

For the authorized disassembly problem, many participants fixated on making associations at the functional or superficial level and were not able to identify the analogous strategy. We suspect that the apparent similarity between analogous elements at the functional and superficial levels prevented the participants from detecting the relevant analogy. When the participants observed similarity at the low levels of comparison, which are more easily found than at the strategy level, they focused on implementing particular characteristics of functions and entities in their design solutions. On the other hand, the participants who solved the promotional mailing problem may have been able to easily identify the strategy because they could not find similarity between analogous elements at the

**Table 4** Examples of analogous elements between the enzyme phenomenon and the authorized disassembly problem at three levels of comparison

| Level of comparison | Enzyme phenomenon | Authorized disassembly | Similarity |
|---|---|---|---|
| Strategy | Bind based on specific substrate; temperature changes the shape of enzyme to release | Assemble based on specific part interface; temperature changes the shape of part interface to disassemble | ✔ |
| Functional | Binding of enzyme to substrate | Attaching of one part to another | ✔ |
| Superficial | Specific shape of substrate | Specific shape of part interface | ✔ |

All three levels of comparison feature some degree of similarity

functional and superficial levels. This finding is contrary to Gentner's [34] proposal that having similar analogous elements at the low levels of comparison helps people map higher-level relations.

The biological descriptions for these two problems did differ in length, and the effect of this difference can be complex. A longer description may provide additional context that can aid designers to identify higher-level relations. However, the same information also provides more stimuli that could distract designers from identifying the higher-level relations.

## *Influence of Readily Available Associations*

We suspect that readily available associations at the functional and superficial levels of comparison for the authorized disassembly problem caused participants to fixate on those particular levels. Participants were able to match analogous solutions from their knowledge with the concept of enzymes binding to specific shapes of substrates. The solutions developed by the students involved using or modifying various types of fasteners or interfaces such as mechanical screws, snap-fits, power supply interfaces, etc.

Most of these solutions were highly relevant to the student's domain knowledge in mechanical engineering. This tendency to develop solutions based on the familiar domain knowledge may be similar to Purcell and Gero's [35] finding that mechanical engineers tend to fixate on using familiar principles to solve design problems. We observed the tendency to depend on domain knowledge, especially if associations to the domain knowledge are readily available at low levels of comparison, prevented novice designers from identifying the analogy. This hypothesis might also explain why the participants were more successful in solving the promotional mailing problem. The problem goal involved logistic optimization and was different from conventional mechanical design problems; therefore, the participants may have been more open to applying the new knowledge gained from the ant phenomenon. In summary, domain knowledge was more likely to induce fixation, rather than help detect the analogy. This observation differs from Novick's [36] finding that domain expertise may help people access potentially useful analogies.

Some participants almost exclusively found associations at the superficial level. For the wet scrubber problem, one particular participant persistently tried to apply superficial characteristics of a penguin's feet, e.g., texture, color, in developing new types of mechanical scrubbers. Mak and Shu [4], Helms et al. [8], and Cheong et al. [6] also reported on novice designers' frequent fixation on superficial characteristics in biomimetic design. Interestingly, another participant within the same group pointed out twice that the analogy should be based on the countercurrent exchange of flows, not on superficial characteristics of penguins. This suggestion, however, did not stop the first participant from fixating on the superficial similarity. The following section discusses this failure to properly evaluate analogies in more detail.

## *Evaluation and Mapping of Analogy*

In some groups, participants fixated on their existing ideas and failed to realize the analogy even when another participant explicitly stated the analogy. What we found interesting was that "structural alignment and consistency", which Gentner [34] lists as important factors of analogy evaluation, had little effect on some participants' likelihood to move away from their fixated ideas. In other words, the fixation on initial ideas was so significant that the participants were no longer properly evaluating the analogies they used. A number of design researchers, including Rowe [37], Ball et al. [38], and Cardoso and Badke-Schaub [39], also report strong fixation effects on initial ideas.

In some cases, participants developed solutions based on the relevant strategy, but expressed that they were not sure if their analogies were correct and complete. This lack of confidence led to either abandoning the strategy or trying to force-fit non-analogous elements that seemed relevant to the strategy. Essentially, the detection of the analogy did not guarantee correct mapping of the analogy. In fact, two promotional mailing groups started to make irrelevant associations between the ant phenomenon and their solutions, e.g., identifying the optimal path to deliver mail or comparing a CEO to a queen ant, *after* they had detected the relevant strategy. One group that solved the wet scrubber problem also showed a similar tendency. After agreeing on using the countercurrent flow exchange, the group tried to elaborate their solution with irrelevant inferences from the penguin phenomenon, e.g., using vein-like channels and considering the distance between the penguin's heart and feet.

Once designers find the relevant analogy, they may be likely to look for new one-to-one mappings from the analog source, instead of performing one-to-many inferences. In other words, designers focus on using multiple features of the source analog, some of which may not be relevant, and fail to develop multiple solutions based on the analogous strategy. This observation suggests that designers may have fixated too much on *mapping* the analogy instead of *projecting* multiple inferences. Gentner [40] and Holyoak and Thagard [41] have reported one-to-one mapping as a constraint in analogical reasoning and we have indeed observed that it has a significant effect in design-by-analogy.

Some of these effects may be partially due to the structure of the experimental design task. The designers were given 20 min to generate concepts, but were not specifically asked to generate multiple solutions.

## *Facilitating Analogical Reasoning*

We observed one particular group overcome fixation, which could provide insights for facilitating analogical reasoning. The group was assigned the authorized disassembly problem, and one participant repeatedly asked questions to himself and other group members about whether they were fixating on specific aspects of the biological phenomenon, as well as how they could apply the biological phenomenon in new ways. These types of questions evidently shifted the group's focus from one particular level of comparison to identifying the relevant analogy. Based on this observation, we believe that the awareness of fixation and its effect on identifying the analogy is a key requirement for effective analogical reasoning in biomimetic design. Winkelmann and Hacker [42] also noted that design performance is increased through the use of interrogative questions, which stimulate reconsideration of the problem. Participants who ask these types of questions without external prompting might be demonstrating enhanced awareness, with the additional benefit that the questions promote increased problem solving among group members.

Lack of awareness of fixation during design problem solving is apparent not only amongst novice designers. Linsey et al. [43] reported that even experienced designers, mostly the engineering faculty members in the authors' study, were not able to accurately perceive the degree of fixation that they were experiencing. Most design-by-analogy methodologies generally do not seem to provide a means for participants to identify fixation effects. Chrysikou and Weisberg [44] provided defixation instructions that helped participants avoid fixating on pictorial examples; however, their instructions were problem specific and may not be transferrable to general design-by-analogy problems. Also, we observed a variety of fixation effects on familiar domain knowledge, superficial attributes, and initially inspired solutions. In complex design tasks including biomimetic design, these multiple types of fixation mean that any one specific mediation approach is unlikely to improve the design process in general. Methods that support biomimetic design [9, 13, 14], most of which are based on modeling biological knowledge, may help designers understand the complex biological information of interest. However, the methods do not fully support mitigating fixation during concept generation.

An effective solution to address this challenge may be to educate or train novice designers to better identify and apply analogies with enhanced awareness of fixation effects. In a meta-analysis of 70 studies on the effects of training programs on creativity, Scott et al. [45] concluded that the most effective programs were the ones that fostered the development of cognitive skills and the necessary strategies to apply them. Specific to biomimetic design, Nelson et al. [46] found that students who took a biologically inspired design course were able to develop more novel and diverse concepts than those who did not take the course and solved the same design problem. Nelson et al. concluded that increased novelty and variety might be due to the students' improved analogical reasoning capabilities from the biologically inspired design course.

Training could also work in congruence with existing methodologies of biomimetic design; therefore, we suggest those researching these methodologies study how designers use the tools. Observational studies on using the tools, such as the one conducted by Vattam et al. [11] would be an effective approach for this purpose. For our research, we are interested in conducting more observational studies to identify characteristics that allow designers to effectively perform analogical reasoning, and develop training materials or strategies to help designers take better advantage of biological analogies.

## Graphical Representation of Similarity Comparisons

After the initial review of our experimental transcripts, we performed a protocol analysis to examine trends in participants' similarity comparisons. The goal was to graphically represent different levels of similarity comparison, i.e., entity, function, strategy, occurring over time and see if those representations support our qualitative observations.

Figure 1 depicts the results of the protocol analysis. The y-axis represents a similarity comparison index; the index value is calculated using a rolling average of instances of similarity comparisons over five time segments (50 s). The graphs visually represent the distribution of similarity comparisons made over time.

In general, more functional-level comparisons coincided with the detection of relevant strategies. In most cases, the strategy-level comparison occurred right after or during an increase in functional-level comparisons.

For Groups 5 and 9 of the promotional mailing problem and Groups 4 and 8 of the wet scrubber problem, entity-level comparisons increased following strategy-level comparisons. This trend supports our observation of participants trying to map entity features of the analog, instead of exploring different solutions based on the detected strategy.

Groups 6 and 7 of the authorized disassembly problem made most comparisons at the functional-levels. These two groups fixated on the domain knowledge that was associated with functional aspects of the biological phenomenon. Group 4 of the wet scrubber problem had a large number of entity-level comparisons. The

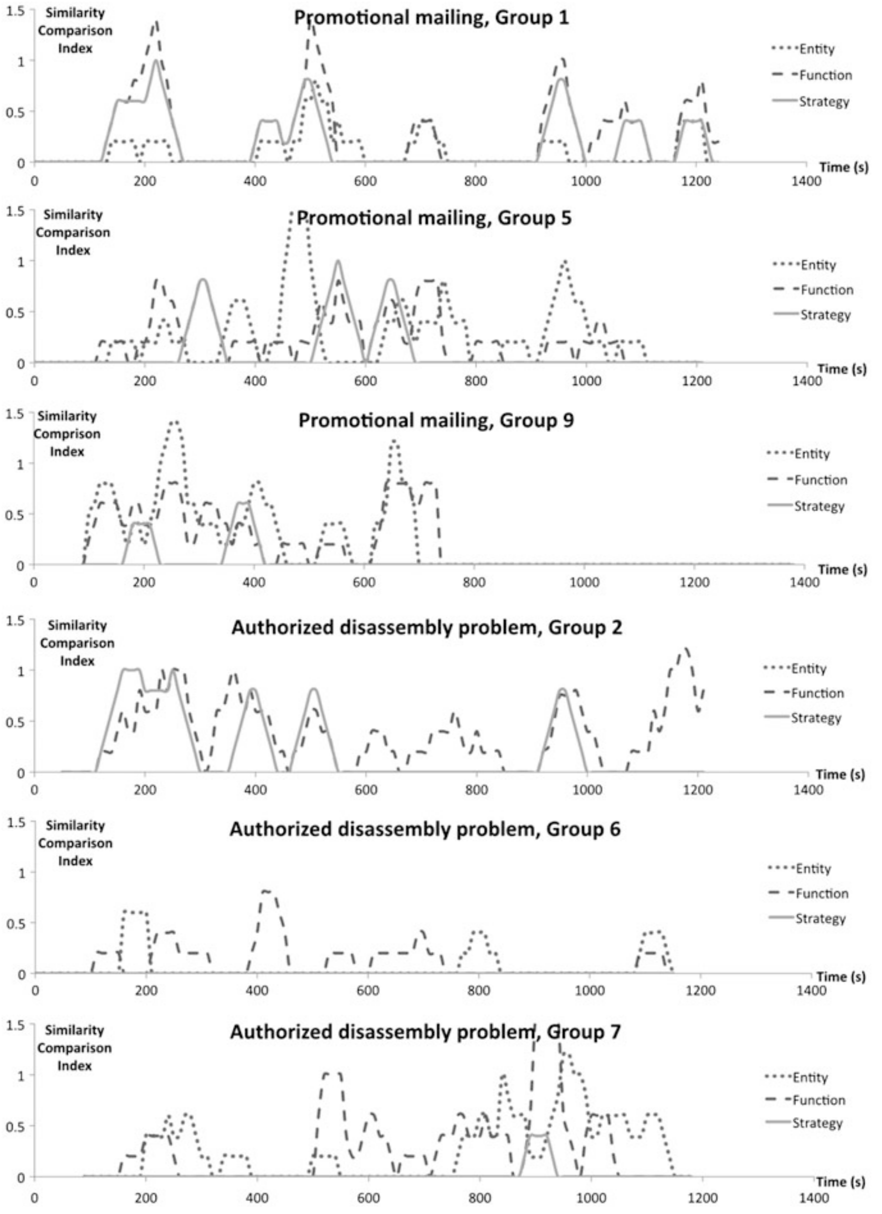


**Fig. 1** Distribution of similarity comparisons (entity, function, or strategy) over time. The similarity comparison index on the y-axis is the rolling average of instances of similarity comparisons over five time segments

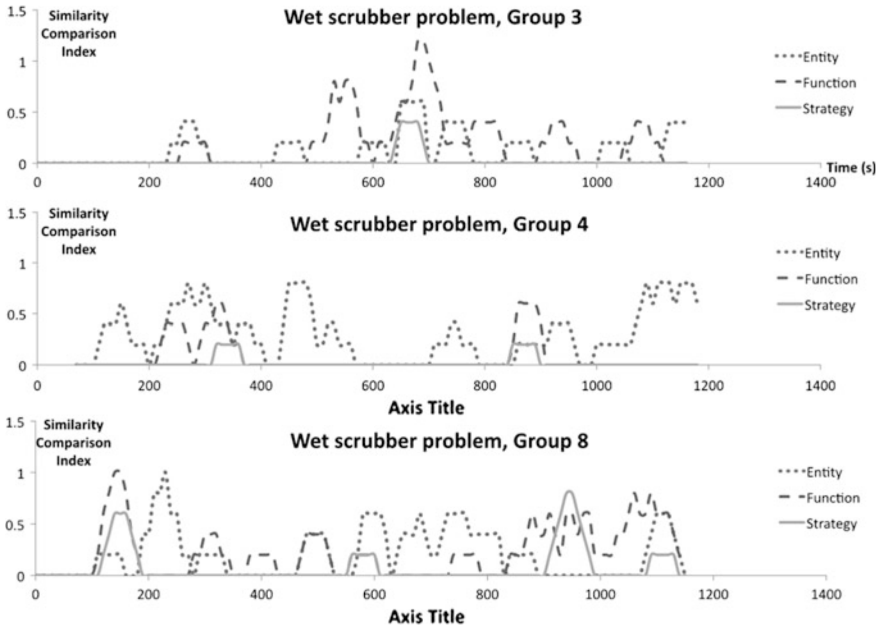


**Fig. 1** (continued)

participant who fixated on superficial characteristics of a penguin's feet was part of Group 4.

While the graphs were able to support some of our qualitative observations, the protocol analysis requires refinement. A particular aspect to address is the subjective and inferential nature of the qualitative coding. Adopting formal coding schemes such as Gero's [25] FBS could enable more consistent identification of coded segments. Computational linguistic and statistical analyses could also be used to perform more in-depth quantitative analysis.

## Conclusion

The current study took a qualitative, inductive approach to understand analogical reasoning in biomimetic design. The interesting observations include:

- Similarity between analogous elements at low levels of comparison, e.g., superficial and functional, prevented novice designers from detecting the overall analogy.
- Domain knowledge can provide readily available associations at low levels of comparison and induce fixation.

- Novice designers focus on mapping multiple features of the source analog, instead of projecting multiple inferences from the identified analogy, perhaps due to lack of confidence in the analogy.

We believe that analogical reasoning, used in practice for complex design tasks such as biomimetic design, can be influenced by many cognitive biases. For instance, we observed that fixation significantly influences the design process, perhaps more so than the ability to reason with analogy.

Another factor that influenced the results could be that our study involved novice designers. To generalize these findings to a larger population and wider context, more natural design situations should be considered for future research, e.g., include expert designers, perform longer design sessions, allow external reference sources and personal selection of analogies.

While the current research focused on qualitative observations, we also value the benefits of quantitative analysis. A number of researchers have well demonstrated the advantages of numerical and statistical analyses on design protocols. In particular, Dong [29] suggests quantitative analysis supported with computational tools opens the possibility of assessing design processes in near real time. Our future research will also explore different methods of quantitative protocol analysis.

**Acknowledgements** We thank the financial support of the Natural Sciences and Engineering Research Council of Canada. We also thank the students of MIE440 at the University of Toronto for participating in this study.

## References

1. Goel A (1997) Design, analogy, and creativity. IEEE Expert 123:62–70
2. Shu LH, Ueda K, Chiu I, Cheong H (2011) Biologically inspired design. CIRP Ann 765:1–21
3. Mak TW, Shu LH (2004) Abstraction of biological analogies for design. CIRP Ann 53(1):117–120
4. Mak TW, Shu LH (2008) Using descriptions of biological phenomena for idea generation. Res Eng Design 19(1):21–28
5. Cheong H, Shu LH (2009) Effective analogical transfer using biological descriptions retrieved with functional and biologically meaningful keywords. In: Proceedings of ASME iDETC2009-86680 (DTM)
6. Cheong H, Chiu I, Shu LH (2010) Extraction and transfer of biological analogies for creative concept generation. In: Proceedings of ASME iDETC2010-29006 (DTM)
7. Vattam S, Helms M, Goel A (2008) Compound analogical design: interaction between problem decomposition and analogical transfer in biologically inspired design. In: Design computing and cognition ’08, pp 377–396
8. Helms M, Vattam S, Goel A (2009) Biologically inspired design: process and products. Des Stud 30(5):606–622
9. Goel A, Rugaber S, Vattam S (2009) Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. AIEDAM 23(1):23–35
10. Helms M, Vattam S, Goel A (2010) The effects of functional modeling on understanding complex biological systems. In: Proceedings of ASME iDETC2010-28939 (DTM)

11. Vattam S, Wiltgen B, Helms M, Goel A, Yen J (2010) DANE: fostering creativity in and through biologically inspired design. In: Proceedings of 1st international conference on design creativity, Kobe, Japan, 127–132
12. Sartori J, Pal U, Chakrabarti A (2010) A methodology for supporting “transfer” in biomimetic design. AIEDAM 24(4):483–505
13. Chakrabarti A, Sarkar P, Leelavathamma B, Nataraju B (2005) A functional representation for aiding biomimetic and artificial inspiration of new ideas. AIEDAM 19(2):113–132
14. Nagel J, Stone RB (2010) A computational concept generation technique for biologically-inspired, engineering design. In: Design computing and cognition ’10, pp 721–740
15. Boden M (2004) The creative mind: myths and mechanisms. Routledge, New York
16. Gentner D, Holyoak KJ, Kokinov BK (2001) The analogical mind. MIT Press, Cambridge
17. Dunbar K (1995) How scientists really reason: scientific reasoning in real-world laboratories. In: Sternberg RJ, David J (eds) Mechanisms of insight. MIT Press, Cambridge, pp 365–395
18. Cross N (2001) Design cognition: results from protocol and other empirical studies of design activity. In: Eastman C, Newstatter W, McCracken M (eds) Design knowing and learning: cognition in design education. Elsevier, Oxford, pp 79–103
19. Chiu I, Shu LH (2010) Potential limitations of verbal protocols in design experiments. In: Proceedings of ASME iDETC2010-28675 (DTM)
20. Merriam S (2009) Qualitative research: a guide to design and implementation, 3rd edn. Jossey-Bass, San Francisco
21. Miles MB, Huberman AM (1994) Qualitative data analysis: an expanded sourcebook, 2nd edn. SAGE Publications Inc, Thousand Oaks
22. Goldschmidt G (1990) Linkography: assessing design productivity. In: Cyberetics and system ’90, R Trappl, World Scientific, Singapore pp 291–298
23. Kvan T, Gao S (2006) A comparative study of problem framing in multiple settings. In: Design computing and cognition ’06, pp 245–263
24. Kan J, Gero JS, Tang H (2010) Measuring cognitive design activity changes during an industry team brainstorming session. In: Design computing and cognition ’10, pp 621–640
25. Gero JS (2010a) Generalizing design cognition. In: Design thinking research symposium, Sydney
26. Bilda Z, Gero JS (2008) Idea development can occur using imagery only. In: Design computing and cognition ’08, pp 303–320
27. Gero JS (2010) Fixation and commitment while designing and its measurement. J Creative Behav 45(2):108–115
28. Kan J, Bilda Z, Gero J (2006) Comparing entropy measures of idea links in design protocols. In: Proceedings of design computing and cognition ’06, pp 265–284
29. Dong A (2004) Quantifying coherent thinking in design: a computational linguistic approach. In: Design computing and cognition ’04, pp 521–540
30. Dong A (2005) Concept formation as knowledge accumulation: a computational linguistics study. AIEDAM 20:35–53
31. Wang X, Dong A (2008) A case study of computing appraisals in design text. In: Design computing and cognition ’08, pp 573–592
32. Brown D (2010) The curse of creativity. In: Design computing and cognition ’10, pp 157–170
33. Saitou K, Shalaby M, Shu LH (2007) Bioanalogous mechanical joints for authorized disassembly. CIRP Ann 56(1):33–36
34. Gentner D (2006) Analogical reasoning, psychology of Encyclopedia of cognitive science. Wiley, New York
35. Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17(4):363–383
36. Novick LR (1988) Analogical transfer, problem similarity, and expertise. J Exp Psychol Learn Mem Cogn 14:510–520
37. Rowe PG (1991) Design thinking. MIT Press, Cambridge
38. Ball L, Ormerod TC, Morley NJ (2004) Spontaneous analogising in engineering design: a comparative analysis of experts and novices. Des Stud 25:495–508

39. Cardoso C, Badke-Schaub P (2011) Fixation of inspiration: creative problem solving in design. J Creative Behav 45:77–82
40. Gentner D (1983) Structure mapping: a theoretical framework for analogy. Cogn Sci 7:155–170
41. Holyoak K, Thagard P (1989) Analogical mapping by constraint satisfaction. Cogn Sci 13:295–355
42. Winkelmann C, Hacker W (2006) The improvement of design solutions by means of a question-answering technique. In: Design computing and cognition ’06, pp 603–618
43. Linsey JS, Tseng I, Fu K, Cagan J, Wood KL, Schunn C (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Mech Des 132:041003
44. Chrysikou EG, Weisberg RW (2005) Following the wrong footsteps: fixation effects of pictorial examples in a design problem-solving task. J Exp Psychol Learn Mem Cogn 31(5):1134–1148
45. Scott G, Leritz LE, Mumford MD (2004) The effectiveness of creativity training: a quantitative review. Creativity Res J 16(4):361–388
46. Nelson BA, Wilson JO, Yen J (2009) A study of biologically-inspired design as a context for enhancing student innovation. In: Frontiers in education conference 2009, San Antonio

# Evaluating Methods for Bioinspired Concept Generation

**Michael W. Glier, Joanna Tsenn, Daniel A. McAdams and Julie S. Linsey**

**Abstract** Bioinspired design, the practice of using biological organisms and systems to inspire the design of engineering systems, has traditionally been performed without the use of systematic tools or methods to aid the designer. In recent years however, several tools have been developed to help designers effectively use bioinspiration for engineering design. These methods include BioTRIZ, functional modeling, biological keyword searches, and online repositories such as Asknature.org. This paper briefly reviews some of these methods and presents the summary of three studies that offer empirical examinations of those methods. In two studies the methods are taught and used by groups of graduate-level engineering students. The successes and difficulties that the students encountered using the bioinspired design methods are discussed and evaluated. Additionally, a third controlled study examines a group of undergraduate mechanical engineering students with no formal training in ideation methods. The students were given one of two design problems and instructed to either generate ideas or to generate ideas while considering how nature might solve the problem. This controlled study allows a quantitative analysis of ad hoc approaches to bioinspired design.

## Introduction

When designers seek to produce innovative solutions to engineering problems, they depend heavily on analogies [1]. Designs can be inspired by analogous systems with functional, strategic, morphological, or other characteristics similar to the desired solution. Typically in engineering design, designers find analogies within the engineering domain. In contrast, bioinspired design uses nature to

---

M. W. Glier (✉) · J. Tsenn · D. A. McAdams · J. S. Linsey
Texas A&M University, College Station, TX, USA
e-mail: gliierm@tamu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_3,

inspire design solutions to engineering problems. Nature offers designers an abundance of time-tested solutions to difficult design problems; solutions that often use significantly different approaches than their engineering counterparts.

While bioinspired design can inspire innovative design solutions, designers often have difficulty in identifying pertinent analogies between the engineering and biological domains. Traditionally, the designer must understand both the engineering and biological systems to draw an analogy between the two [2]. Bioinspired design has often been performed informally on a case-by-case basis. Only in the past several years have formal, systematic methods been developed to aid in bioinspired design [3].

The paper begins by introducing several methods for bioinspired design, namely BioTRIZ, functional modeling, and biological keyword searches. With these tools in mind, the paper summarizes three recent studies on bioinspired design methods. The first study offers a quantitative comparison of the number of ideas generated by senior-level mechanical engineering students when solving one of two design problems, using either no formal idea generation method or a "directed" bioinspired approach where the students are instructed to consider how nature might approach the problem [4]. This will help gauge the efficacy of the directed method for engineering design and provide a valuable benchmark for future studies that quantitatively measure ideas generated using more formal bioinspired design methods. The following two studies evaluate the performance of two groups of graduate engineering students using some or all of the introduced formal bioinspired design methods to solve specific design problems [5]. While research has been done on the difficulties students encounter as they learn to use bioinspiration [6, 7], no published work examines the methods discussed here. The results of these methods are discussed, along with feedback on the methods from the students.

## Background

One of the great challenges for designers attempting to use biological systems to inspire engineering design solutions is the difficulty in drawing useful analogies between the two fields. A large amount of research has been invested into overcoming these challenges. Work in design cognition [7–9], design analogies [10–15], the creation of an array of tools and methods for bioinspired design [10, 16–26], and efforts in teaching bioinspired design as part of an engineering curriculum [6, 7, 27] have provided great insight into how to effectively use biology to inspire engineering design. While the majority of these contributions do not have a direct bearing on this work, [3] offers a thorough review of recent research in tools and methods for bioinspired design. The following sections describe the development and methodology of BioTRIZ, functional modeling for bioinspired design, and biology search tools. Because the first study examines the number of non-redundant ideas produced for a design problem, a metric for systematically measuring the number of ideas is also presented.

## *BioTRIZ*

BioTRIZ is based on Altshuller's Theory of Inventive Problem Solving (TRIZ) [28]. By framing a design problem in terms of pairs of conflicting parameters, TRIZ allows designers to find techniques that have been used to solve similar conflict in past designs [29, 30]. TRIZ identifies 39 system parameters that designers may choose to optimize as well as 40 inventive principles (IPs) that can be used to resolve design challenges. The set of conflicts and solutions is presented as a 39 by 39 "contradiction matrix". In each cell, the IPs that other designs have used to solve the conflicting parameters corresponding to the cell's row and column are listed.

Like TRIZ, BioTRIZ condenses design information into a contradiction matrix that lists IPs used to solve conflicts between system parameters [25]. However, while TRIZ shows designers how design problems have been solved in technical and engineering designs, BioTRIZ shows how the problems are solved by natural systems and is based on the analysis of approximately 500 biological phenomena [25]. One other important difference between TRIZ and BioTRIZ is that BioTRIZ groups the 39 system parameters of TRIZ into six fields of operation: substance, structure, space, time, energy, and information. Consequently, the conflict matrix for BioTRIZ is a 6 by 6 matrix. However, BioTRIZ does retain the 40 IPs used in TRIZ. The procedures used to apply BioTRIZ to a design problem are identical to those used for TRIZ.

## *Functional Modeling*

Functional modeling allows designers to examine a system at a purely functional level, without concentrating on the system's physical mechanisms. While this method is often useful in engineering design, it is particularly useful for bioinspired design as it helps designers develop engineering systems that mimic the functional behavior of biological systems without fixating on specific mechanisms used by the inspiring biological system. Many of the research efforts in applying functional modeling techniques to bioinspired design have used the Functional Basis, as described by Hirtz et al. [34]. Tinsley et al. determined that the Functional Basis is an effective medium for transferring biological design solutions to the engineering domain [22].

As biological systems tend to be very complex, they can often be modeled at multiple levels (e.g. cellular vs. organ level). Nagel et al. demonstrated that defining the category and scale of the model in advance allows designers to consistently isolate and model those facets of the biological system most applicable to their design [2]. To systematically apply functional modeling to bioinspired design, Nagel proposed a seven-step methodology wherein designers would begin by finding a reference for the biological system of interest, develop an understanding of the system's operation, define a question for the model to answer, define a category and scale for the model, develop the model within the defined bounds, and double check the model against a black box model and the defined question.

Functional modeling's systematic, detailed approach can help designers find properties of biological phenomena that may not be apparent with other design techniques. This technique allows designers to examine biological systems in a familiar, engineering oriented language; designers that have used functional modeling for traditional engineering design can simply extend its use to bioinspired design rather than having to learn a completely new technique. However, even with these advantages, designers still face certain challenges.

Simply finding a biological system that is relevant to a design problem poses significant difficulty to designers without a strong background in biology. Finding an appropriate reference for the relevant system is non-trivial given the vast amounts of biological literature available. To accurately model a biological system, the designer must have a firm understanding of the biological system. Even with an accurate and useful functional model, the designer's creativity is still necessary to conceptualize a functionally similar engineering system.

## *Biological Keyword Search*

To aid in drawing analogies from nature, information-gathering tools in the form of bio-keyword searches are used. Relevant biological systems can be identified using engineering oriented terms and the Functional Basis. The "engineering-to-biology thesaurus" relates terms from the Functional Basis to biologically meaningful keywords [18]. After creating a functional model using the functional basis terms, the terms can be used with the thesaurus to find biological correspondents. A search is then done with a conventional search engine or database on the biological terms to identify biological systems that can serve as sources of inspiration for design solutions. By linking general functional terms to more specific biological terms, the thesaurus helps designers find relevant sources of biological inspiration for their design problem.

## *Idea Generation Metric*

Comparing the effectiveness of idea generation processes poses many challenges and requires systematic metrics. One basic measure of ideation process effectiveness is the quantity of ideas produced using the process. Building from the work of Shah et al. [31, 32] and implemented by Linsey et al. [33], the basic definition of a single "idea" is something that fulfills at least one function of the

Functional Basis as described in Hirtz et al. [34]. The complete listing of counting rules is omitted here for brevity, but can be found in Linsey et al. [33].

To count the number of ideas produced by each participant, the rater must identify each component in every participant's design that fulfills one of the terms in the Functional Basis. A more detailed sketch or description of a solution will generally identify more features, so a more detailed solution will often have a larger number of ideas associated with it.

## Study Methods

The methods introduced for bioinspired design have the potential to help designers effectively use nature as a source of inspiration. However, before the methods can be adopted or successfully taught to students, they need to be evaluated to discover the difficulties encountered by designers and the effectiveness of the methods. In this section we will discuss the methods of each study, explaining the studies' purposes, the demographics of the participants, the problems that were posed to participants, and the methods used to analyze the results.

### *Directed Bioinspired Design Method Study*

Before claiming that the discussed formal methods can offer great advantages to the designer, we must first establish a baseline measurement for the natural, *ad hoc*, approach that a designer would adopt in the absence of any formal methods. This study compares the performance of novice designers using no formal idea generation methods to those using a "directed" bioinspired method. In the directed method, the participants were instructed to consider how nature would solve the problem.

The 121 senior-level undergraduate participants in this study were all enrolled in the first semester of Texas A&M's Mechanical Engineering capstone design course. At the time of the experiment, the students had not yet been taught idea generation methods in the course. Each participant was given one of four design problem statements, which are shown in Table 1 with the number of participants that received each statement. There were two design problems and two conditions per problem. One condition gave the basic design problem and the other instructed the participant to "imagine how nature would solve the problem". The statement for the corn shucker problem included a photograph of an ear of corn with husks and silks displayed [4].

The participants were instructed to write or sketch their ideas and attempt to generate as many ideas as possible with the greatest quality and variety possible, even if those ideas may not be technically feasible. The students were given 50 min to generate ideas and record them with pens and paper.

**Table 1** Problem statements and numbers of participants per condition for the directed method study

| Problem Statement Bracketed text added for directed condition | Participants (Normal/ Directed) |
|---|---|
| Corn is currently the most widely grown crop in the Americas with the United States producing 40 % of the world's harvest. However, only the loose corn kernels are used when bought canned or frozen in grocery stores. An ear of corn has a protective outer covering of leaves, known as the husk, and strands of corn silk threads run between the husk and the kernels. The removal of husk and silk to clean the corn is known as shucking corn. Design a device that quickly and cheaply shucks corn for mass production. **{Imagine how nature would solve this problem}**<br>**Customer Needs:**<br>• Must remove husk and silk from corn cob with minimal damage to kernels<br>• A large quantity of corn must be shucked quickly<br>• Low cost | 24/31 |
| Alarm clocks are essential for college students, however often times they will wake up a roommate and those around them as well. Design an alarm clock for individual use that will not disturb others. The clock should be portable for use in a variety of situations such as on the bus, in the library, or in a classroom. **{Imagine how nature would solve this problem}**<br>**Customer Needs:**<br>• Must wake up individual with no disturbance to others<br>• Must be portable and lightweight<br>• Electrical outlets are not available as a constant power source<br>• Low cost | 30/36 |

Additions to the statement for the directed conditions are noted within brackets

The collected ideas were rated for quantity using the previously described metric to determine the total number of non-redundant ideas generated by each participant. The number of ideas per condition was calculated as the average of the number of ideas per participant. To ensure that these results were reproducible, a set of results from ten participants from each condition were independently rated by a second rater [4].

## *TRIZ/BioTRIZ Study*

In this study, a group of 12 graduate-level mechanical engineering students was introduced to the BioTRIZ, functional modeling, and bio-keyword search tools over the course of three 50 min lectures in a graduate design course at Texas A&M University. Of these students, 7 were actively involved in design-related research. The course was structured to introduce many different design methods to the students, who then applied the techniques to a design project as part of the course.

After learning to use TRIZ, the students were issued a simple design problem and asked to generate solutions using the method. Later, the students were taught

BioTRIZ and asked to repeat the problem using the new method. The problem statement asked students to design a wire-tying tool to form rebar reinforcements for concrete structures. A copy of the problem statement is inset below with the bracketed text noting the parts that differed between TRIZ/BioTRIZ. Both problems were given as homework assignments due 1 week after they were assigned.

Concrete is a ubiquitous building material that plays a huge part in building infrastructure in developing countries. However concrete is, by itself, a very weak material, particularly under tensile loads. Consequently, concrete structures are almost always reinforced by pouring the concrete around a framework of steel reinforcing bars (rebar). The rebar framework is constructed by tying rebars together with steel wire ties. While these tied joints can be made by sophisticated automatic tying machines, such machines are expensive and largely unavailable in the developing world. Instead, the ties are formed by hand with a simple pair of pliers. This approach is difficult, time consuming, and produces weaker joints that result in weaker structures.

We wish to design a simple wire-tying tool appropriate for the developing world. A good design for this tool will be easy to use, easily portable, durable, and easy to repair with simple tools. Identify the design conflicts in this design in terms of {the Generalized Engineering Parameters /BioTRIZ's Fields of Operation}. {Use the conflicts you identify /Use BioTRIZ} to find Inventive Principles that can be used to resolve the design conflicts. Finally, use the Inventive Principles to generate at least one design for a wire-tying tool. Sketch your idea and note where you have applied the Inventive principles you found.

The collected assignments were later analyzed to find difficulties that the students encountered applying TRIZ and BioTRIZ. Each assignment was checked to ensure that each step of the methods was completed and that the methods were correctly applied. Further, the resulting solutions from the TRIZ and BioTRIZ assignments were compared for each participant to uncover similarities and differences. Any comments from the students or notes describing their thought processes were noted and evaluated along with feedback concerning the methods.

## *Bioinspired Redesign Project Study*

In this last study, the introduced bioinspired design techniques were taught to 15 interdisciplinary engineering graduate students at Texas Tech University. The students averaged 45 years of age and 20 years of industry experience. The majority of the students were Transdisciplinary Engineering majors, but others studied Systems Engineering, Mechanical Engineering, and Information Security. The students were working professionals in engineering fields who were enrolled in a program that consisted of classes 8 h a day, Friday through Sunday [5].

Bioinspired design methods were taught to the students during one of these weekend sessions. The students were given a long-term design project to be completed several weeks after the course. Ten of the students in the class submitted the design project. These projects gave us valuable insight into how students use the bioinspired design techniques and the challenges they face.

The project asked each student to select an engineering system, product, or technology and to redesign the system using the bioinspired design techniques taught in the course. Any system was acceptable for the project, but a system with innovative characteristics and a clear bioinspired component was preferable. After selecting the system, the students were instructed to pose the design problem in a solution-neutral manner with clear needs, constraints, and performance expectations. Students then attempted to find solutions to the design problem using BioTRIZ and a combination of functional modeling and bio-keyword searches. The students were asked to carefully document their work as they generated concepts with these methods. Finally, the students compared the ideas generated by each method and critiqued of the methods.

The submitted design reports were analyzed to see how well the students were able to apply the bioinspired design methods. Because the exact format of the design reports was not specified, the report structures varied greatly. Points where the methods were misapplied were noted and the students' accompanying explanations were studied to identify the likely causes of any mistakes. Student feedback on bioinspired design was compiled [5].

## Summary of Key Results and Discussions

With an understanding of some methods for bioinspired design and the setup of each study, we can proceed to discuss the relevant results from the studies. Because the Project study examines both BioTRIZ and functional modeling with bio-keyword searches, it is broken into two subsections for clarity.

### *Directed Bioinspired Design Method Study*

Figure 1 shows the average number of non-redundant ideas generated for each problem statement. The error bars indicate one standard error. The ideas were counted by a single rater using the quantity metric introduced earlier. A second rater counted results from ten participants from each condition to ensure that different individuals using the same metric would arrive at similar counts. Using Pearson's correlation coefficient, the inter-rater agreement is 0.87. Figure 1 suggests that there is no statistically significant difference between the number of non-redundant ideas generated using the directed method and using no idea generation method. A 2-tailed t-test with unequal variances confirms this, giving $p$-values of 0.58 and 0.34 for the corn shucker and alarm clock problems, respectively [4].

Simply considering nature when generating ideas does not increase the number of ideas that novice designers produce. This examination does not consider the

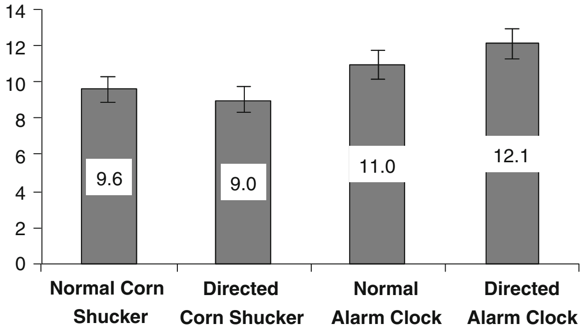


**Fig. 1** Average number of non-redundant ideas produced for each experimental condition (±1 SE)

effectiveness of the directed method when the designers have access to specific examples of analogous biological systems or specialized knowledge of some inspiring system. Further, we must consider that the directed method may help designers find solutions of higher quality, variety, or novelty. Nevertheless, this study suggests that using biological inspiration for engineering design without any formal approach is not advantageous to the designer. With this baseline measurement for ideation, more formal bioinspired design methods like BioTRIZ and functional modeling can be quantitatively compared in future studies. The qualitative results of the following studies, however, cannot be effectively compared to this experiment.

## *TRIZ/BioTRIZ Study*

The students in this study were very successful in applying both TRIZ and BioTRIZ. All students were able to identify the relevant needs from the problem statement, convert those needs into conflicts between engineering parameters or fields of operation, and use those conflicts to determine IPs through the conflict matrix. For both TRIZ and BioTRIZ, only one student failed to generate a solution although it is unclear whether this was due to difficulty in applying the methods or a misunderstanding of the problem's instructions.

The students used the same set of design conflicts to find IPs with both TRIZ and BioTRIZ, but the two methods typically led them to the generation of different solutions to the problem. However, the students demonstrated that the set of IPs generated using the two methods could often have common entries, despite the low similarity between their conflict matrices, as noted by Vincent et al. [25]. Generally, the common IPs come from different design conflicts. For example, consider Table 2, which is adapted from one student's submissions. For a single conflict, no IPs are shared, but when the pair of conflicts are considered, TRIZ and BioTRIZ both suggest using IPs 1, 4, 7, and 25. On average, 31 % of the inventive principles identified by students using TRIZ were also identified when using BioTRIZ.

Despite many overlapping IPs, the students produced different solutions using the two methods. However, five students produced solutions that were noticeably

**Table 2** Two sets of conflicting parameters, translated into engineering parameters and fields of operation, and the IPs that can be used to resolve the conflicts. The IPs are identified by number, as corresponding to Otto and Wood [35]

| Conflict pair | Conflicting parameter | Engineering parameter | Field of operation | TRIZ inventive principles | BioTRIZ inventive principles |
|---|---|---|---|---|---|
| 1 | Fast operation | Duration of action by moving object | Time | 1, 2, 3, 4, 7, 38 | 14, 25, 26, 28 |
| | Small/light weight | Shape | Space | | |
| 2 | Simple Design | Reparability | Information | 3, 10, 16, 23, 25 | 1, 4, 7, 16 |
| | Small and compact | Flexibility | Information | | |

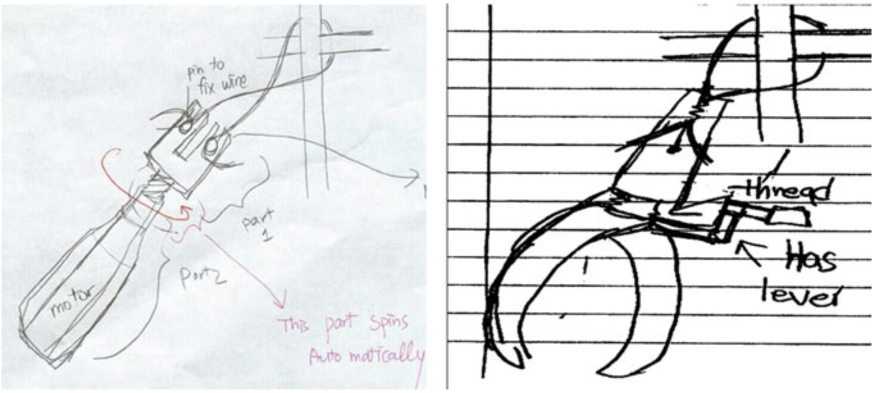


**Fig. 2** Designs of a wire-tying tool using TRIZ (*left*) and BioTRIZ (*right*). The TRIZ solution uses the principle of segmentation while the BioTRIZ uses the principle of moving in a new dimension

similar for the two methods. Although different IPs were used to change particular details, the overall form and operation remained similar. This is likely an example of design fixation where the designer had a specific design in mind and used the design method to modify that design. Figure 2 shows an example of this fixation where the student designed plier-like tools with a rotating head. Although the solutions use different IPs, the two tools are very similar in their overall operation.

## *Bioinspired Redesign Project Study*

To best organize the discussion of this study, the results are presented in two subsections, one addressing BioTRIZ and the other addressing the functional modeling and keyword searches. However, this organization is not necessarily reflective of how these methods were applied. Some students in this study did use BioTRIZ and functional modeling with keyword search tools to develop separate, distinct solutions. Others, however, used the methods together to develop a single solution. Further, it should be noted that even when students used the methods independently, the use of one design method likely biased the results produced using the other. Given the nature of this study, it is impossible to isolate these effects. Nonetheless, the study does demonstrate that the methods can lead designers to different solutions.

## BioTRIZ

The students were generally successful in using BioTRIZ to generate ideas. Following the method's clear procedure, all of the students were able to find Inventive Principles applicable to their problem. They only seemed to encounter difficulties after finding the IPs. Three of the students misunderstood the project statement and stopped after finding the IPs rather than proceeding to generate designs from them. Of the remaining students, two seemed to be fixated on either the original system they had selected or related existing solutions, generating variations on the initial system rather than original ideas. This fixation may be partly due to the fact that the project was a redesign rather than a blank-slate design [5].

For five of the reports, the solutions generated in BioTRIZ led to concepts that were a part of the final design rather than creating a complete design. For example, one student chose to look at generating energy from tidal forces. To resolve the issue of capturing energy in a small physical space, the student used the IP of "transition to another dimension" and created a design combining the horizontal movement of a shark tail and the vertical movement of a dolphin tail, as seen in Fig. 3. This combined with another of his designs, which was to position the capture device below water and the generator above to decrease the device's exposure to corrosive saltwater.

According to a student critique of the bioinspired design method in the reports, the students found BioTRIZ to be a new way of evaluating the problem. Three of the students commented that the method promoted creative and inventive ideas when trying to resolve the design conflicts. While BioTRIZ can be applied to challenging problems to find novel solutions, students also said that the method was easy to implement and straightforward. The students were guided through the process of finding conflicts and resolving them using the IPs. Three students remarked that the method was fast and time-efficient, however they also acknowledged that the quantity and originality in the designs generated using BioTRIZ was tied to the amount of time and effort applied by the student. Two of the students suggested that the method would work well when creating an initial design because the conflicting issues could be found at the beginning and resolved.

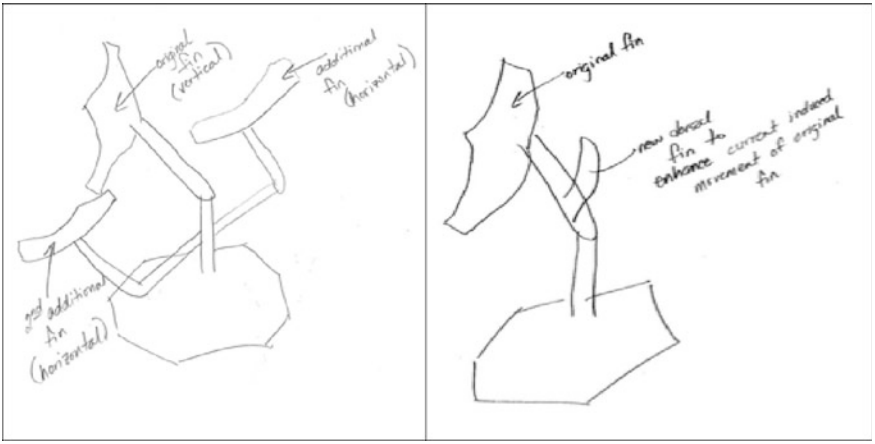


**Fig. 3** Bioinspired concepts to capture more energy from a tidal wave, generated with BioTRIZ (*left*) and functional modeling (*right*) [5]

## Functional Modeling and Keyword Search

Functional modeling was combined with bio-keyword searches since they both make use of the Functional Basis, which can be applied to the engineering-to-biology thesaurus. Unlike BioTRIZ, the successful creation of a functional model was only completed by seven students and was correctly used by even fewer. The greatest challenge the students faced was that the majority of the students were not familiar with functional modeling for engineering design and very few of them were able to create a useful functional model of their system [5]. While engineering students with some background using functional modeling might find its use in bioinspired design to be a simple extension of their knowledge, the students in the study had to learn an entirely new technique and did not have enough practice to create effective functional models of their systems. The students that did not fully understand functional modeling tended to have difficulty generating bio-keyword search terms that yielded useful sources of biological inspiration; some models were constructed such that they clearly led to the original engineering solution rather than being solution-neutral.

For example, one student chose to re-engineer a flapping-wing design for Micro Air Vehicles but was not able to produce a useful functional model; the resulting model contained such functions as “weight” and “speed”, which are not Functional Basis terms. The search terms derived from this model included each of the design functions in his functional model and also included items such as “flapping-wing.”

Another issue found during bio-keyword searches was a fixation on the original design. Once again, this was most likely due to the redesign nature of this project. However, this was more noticeable than for BioTRIZ because of the record of their choice in search terms.

**Fig. 4** Concept based directly on a biological system, instead of drawing an analogy [5]

Several students did not draw analogies from the biological sources of inspiration they found, but instead directly copied the biological systems. For instance, one student proposed creating electro-mechanical copies of fish as high-acceleration underwater vehicles. The resultant eel-based design, shown in Fig. 4, demonstrates the student's replication of an eel found in nature rather than using the animal as a basis for an analogy. Helms, Vattam, and Goel observed similar behavior in a study of design teams in an introductory bioinspired design course at Georgia Tech [7].

Despite the many problems faced by the students in using functional modeling for bioinspired design, a few students did form useful functional models of their system and drew effective design analogies from the biological sources. These students commented that the designs generated using functional modeling were different from those using BioTRIZ. Returning to the example of the student who chose to design devices to generate power from tidal energy, a very detailed functional model was created and four functional requirements were chosen for the bio-keyword search. From the search, a new concept was created after finding that combining the action of the dorsal fins and caudal fins on a fish can produce more thrust. The new design has the same purpose as the one shown in Fig. 3 from BioTRIZ, but accomplishes it in a new manner. This demonstrates that functional modeling and BioTRIZ can be used to complement one another rather than needing to choose one method over the other.

Not surprisingly, the students who had difficulty using functional modeling found the technique to be much less useful than BioTRIZ. Many observed that biological solutions from their keyword searches gave them valuable inspiration for generating new solutions. Using functional modeling required the students to perform problem segmentation and partition the capabilities of the constituents. The students pointed out that an in-depth understanding of the problem was needed in order to create the functional model. Similar to BioTRIZ, students reported that investing more time and effort in the functional modeling method would lead to more design concepts.

## Conclusions

Functional modeling, BioTRIZ, and bio-keyword searches all aid designers in finding design inspiration for engineering systems from natural systems. These tools can help designers access the vast amounts of design information contained in nature by giving a systematic approach to finding relevant biological systems and abstracting their designs to forms more accessible to engineers. We have examined three studies using bioinspired methods. The first compared an *ad hoc* approach to bioinspired design, directing novice designers to consider how nature might solve a problem, to idea generation without any systematic method. The second study examined a group of graduate design students taught both TRIZ and BioTRIZ to find the difficulties they encountered using the method. The final study showed how a group of working professionals was able to apply bioinspired design methods in simple design projects.

The first study demonstrated that attempting to use biology to inspire engineering design without the aid of a systematic approach offers the designer no advantage in producing more ideas. There was no significant difference in the number of ideas generated by novice designers that considered nature and those that did not.

The last two studies offered insight into how students apply BioTRIZ. Using BioTRIZ to identify design conflicts and IPs that can be used to resolve design conflicts is straightforward and posed no apparent difficulty to the students. The difficulty lies instead in conceptualizing specific solutions from the very general IPs. Further, some students saw no practical difference between TRIZ and BioTRIZ and consequently, saw little reason to use both methods for future designs. If designers are to adopt a new design method, the method must be useful and also perceived as useful.

The final study showed that more emphasis needs to be placed on teaching proper functional modeling. If individuals are already accustomed to functional modeling, then learning functional modeling for bioinspired design is easier. A useful functional model of a system would not only aid engineers in general engineering design, but it would also make the transition to using the technique for bioinspired design much more simple. It aids students in deconstructing a design problem into its functional components so that an analogous function in nature can be more easily found. The use of the functional models and the Functional Basis in conjunction with the engineering-biology thesaurus proved helpful and example systems found using the bio-keyword searches were useful sources of inspiration. The students who understood functional modeling and created a good model were able to retrieve better results from their search when using the terms from the engineering-to-biology thesaurus. While creating a functional model and performing the bio-keyword search takes more time than BioTRIZ, there are a greater potential number of design concepts due to the considerable amount of information on biological systems. One item to note is that the importance of drawing analogies from nature needs to be emphasized rather than trying to replicate nature directly. Without making use of analogies, a designer loses the opportunity to find innovative solutions that are inspired by nature at the functional level. The bioinspired methods help to reinforce the utilization of analogies, but analysis of the study shows that the problem of drawing analogies persists.

## Future Work

In terms of evaluating idea generation methods, while the quantity metric of the first study gives a starting point, solutions produced using the directed method should be analyzed using other metrics such as quality, novelty, and variety [4]. Even the quantity of ideas could be revisited, using a metric based on TRIZ conflicts rather than using the Functional Basis as a framework.

Learning more about these bioinspired design methods will help reveal how to most effectively use the methods. The results from these studies were a start, but the effectiveness of each technique must be measured in a controlled experiment. Further, students should be tested using standard ideation techniques or bioinspired methods as they progress through an engineering curriculum in order to observe the effects of the increasing engineering knowledge and experience. Additionally, students with domain specific knowledge, such as biology or biomedical majors, may be better able to draw inspiration from nature than designers without such background knowledge.

**Acknowledgements** Support for this project was provided by the National Science Foundation NSF EEC 1025155. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Boden MA (1994) Dimensions of creativity. MIT Press, Cambridge
2. Nagel JKS, Nagel RL, Stone RB, McAdams DA (2010) Function-based, biologically inspired concept generation. Artif Intell Eng Des, Anal Manuf 24:521–535
3. Glier MW, McAdams DA, Linsey JS (2011) Concepts in biomimetic design: methods and tools to incorporate into a biomimetic design course. Paper presented at the international design engineering technical conferences 2011, Washigton DC, 28–31 Aug 2011
4. Glier MW, Tsenn J, Linsey JS, McAdams DA (2012) Evaluating the directed method for bioinspired design. Paper presented at the international design engineering technical conferences 2012, Chicago, 12–15 Aug 2012

5. Glier MW, Tsenn J, Linsey JS, McAdams DA (2011) Methods for supporting bioinspired design. Paper presented at the international mechanical engineering congress & exposition 2011, Denver, 11–17 Nov 2011
6. Bruck HA et al (2007) Training mechanical engineering students to utilize biological inspiration during product development. Bioinspiration Biomimetics 2:S198–S209
7. Vattam S, Helms M, Goel AK (2007) Biologically-inspired innovation in engineering design: a cognitive study. Georgia Institute of Technolgy, GIT-GVU-07-0
8. Helms M, Vattam SS, Goel AK (2009) Biologically inspired design: process and products. Des Stud 30:606–622
9. Vattam SS, Helms ME, Goel AK (2010) A content account of creative analogies in biologically inspired design. Artif Intell Eng Des, Anal Manuf 24(04):467–481
10. Chiu I, Shu LH (2007) Using language as related stimuli for concept generation. Artif Intell Eng Des, Anal Manuf 21:103–121
11. Hacco E, Shu LH (2002) Biomimetic concept generation applied to design for remanufacture. Paper presented at the ASME design engineering technical conferences 2002, Montreal
12. Shu L (2006) Using biological analogies for engineering problem solving and design. In: Proceedings of 3rd CDEN/RCCI international design conferences, Toronto
13. Shu LH et al (2011) Biologically inspired design. CIRP Annals Manuf Tech 60:673–693
14. Mak TW, Shu LH (2004) Abstraction of biological analogies for design. CIRP Annals Manuf Tech 53:117–120
15. Mak TW, Shu LH (2008) Using descriptions of biological phenomena for idea generation. Res Eng Des 19:21–28
16. Chakrabarti A, Shu LH (2010) Biologically inspired design. Artif Intell Eng Des, Anal Manuf 24:453–454
17. Cheong H et al (2011) Biologically meaningful keywords for functional terms of the functional basis. J Mech Des 133:021007
18. Nagel J, Stone R, McAdams DA (2010) An engineering-to-biology thesaurus for engineering design. Paper presented at international design engineering technical conferences 2010, Montreal
19. Nagel RL et al (2008) Exploring the use of functional models in biomimetic conceptual design. J Mech Des 130:121102
20. Shu L et al (2007) Integrating function-based and biomimetic design for automatic concept generation. Paper presented at the international conference on engineering design '07, Paris
21. Shu LH (2010) A natural-language approach to biomimetic design. Artif Intell Eng Des, Anal Manuf 24:507–519
22. Tinsley A et al (2008) Exploring the use of functional models as a foundation for biomimetic conceptual design. Paper presented at the international design engineering technical conferences 2007, Las Vegas
23. Vakili V et al (2007) Including functional models of biological phenomena as design stimuli. Paper presented at the international design engineering technical conferences 2007, Las Vegas
24. Vattam S et al (2010) DANE: fostering creativity in and through biologically inspired design. In Taura T, Nagai Y (eds) Design Creativity 2010. Springer, Kobe
25. Vincent JFV et al (2006) Biomimetics: its practice and theory. J The Royal Soc Interface 3:471–482
26. Wadia A, McAdams DA (2010) Developing biomimetic guidelines for the highly optimized and robust design of complex products or their components. Paper presented at international design engineering technical conferences 2010, Montreal
27. Jenkins CH (2011) Doing BiE: lessons learned from teaching bio-inspired engineering. Paper presented at the international mechanical engineering congress & exposition 2011, Denver
28. Altshuller G, Rodman S (1999) The innovation algorithm: TRIZ, systematic innovation and technical creativity. Technical Innovation Center Inc., Worcester
29. Savransky SD (2000) Engineering of creativity: introduction to TRIZ methodology of inventive problem solving. CRC Press, Boca Raton

30. Terninko J (1998) Systematic innovation: an introduction to TRIZ; (theory of inventive problem solving). CRC Press, Boca Raton
31. Stone RB, Wood KL (2000) Development of a functional basis for design. J Mech Des 122:359–370
32. Shah J (2003) Metrics for measuring ideation effectiveness. Des Stud 24:111–134
33. Linsey J et al (2005) Collaborating to success: an experimental study of group idea generation techniques. Paper presented at the design engineering technical conferences 2005, Long Beach, CA
34. Hirtz J et al (2002) A functional basis for engineering design: Reconciling and evolving previous efforts. Res Eng Des 13(2):65–82
35. Otto K, Wood KL (2001) Product design: techniques in reverse engineering, systematic design, and new product development. Prentice Hall, Upper Saddle River

# Part II
# Design Cognition-1

# Role of Personas and Scenarios in Creating Shared Understanding of Functional Requirements: An Empirical Study

**Eric Blanco, Franck Pourroy and Serap Arikoglu**

**Abstract** Elicitation of requirements is a key step of the design activity. The building of a shared understanding of design requirements is essential to the performance of the design. Personas and scenarios are used in order to define end users and their needs. Their usage is becoming more and more popular, especially in Software and System Engineering and Human Computer Interaction (HCI). Our hypothesis is that scenarios and personas improve shared understanding of functional requirements between co-designers. In order to test this hypothesis, an empirical study has been undertaken in a laboratory context. This paper presents the protocol of the study and discusses the indicators used for measurement of shared understanding.

## Introduction

Design is sometime described as "a problem of resolving tension between what is needed and what can be done" [1]. During the engineering design process, the design team translates the end user needs into a set of product specifications, a measurable detail of what the product has to do in order to satisfy them. In this paper, the word "need" is used in the same sense as Ulrich and Eppinger [2]. That is to label any attribute of a potential product that is desired by the end user, for whom the product is designed. Ulrich and Eppinger [2] state that "product specifications do not tell the team how to address the customer needs, but they do represent an unambiguous agreement on what the team will attempt to achieve in order to satisfy the customer needs". In order to define the product specifications, functional analysis might be used [3]. Functional analysis builds a standard

---

E. Blanco (✉) · F. Pourroy · S. Arikoglu
Grenoble-INP, Grenoble, France
e-mail: Eric.Blanco@grenoble-inp.fr

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_4,

language to enable designers to share their viewpoints about needs and constraints. In functional analysis, the functional requirements refer to the needs and constraints. Product specifications are derived from the list of the defined functional requirements.

However, in the early stages of the new product design process, the end users are not always defined sufficiently well enough to clearly identify their needs, or for them to be involved in the design process. Moreover, at the early stages of the design process, the final product does not yet exist. What do exist are the intermediary objects which help designers to represent, manipulate and translate the product idea on which they work; such as sketches, diagrams, written specifications etc. [4, 5]. Even for contract projects it may be difficult to gain access to a client who is busy or located geographically at distance. Additionally, the designers generally work under time pressure, which makes it difficult to access end users to get data or integrate them to the design process. Furthermore, researchers, who undertake market and user research, are not typically the design actors, and the results very often comprise ambiguities, uncertainties and gaps that the designers have to manage.

Consequently, in the case that the information about the users is not available at the right time or difficult to understand or to remember, each design actor may interpret the end user needs differently and become sensitive to different product constraints. This lack of shared understanding of end users and of their needs, between design actors, may cause difficulties in defining product specifications and cause non-convergent design processes [6]. To overcome this, support methods might be used in order to define the end users and their needs in order to improve shared understanding of functional requirements between design actors. However, the appropriateness and effectiveness of the various methods is unknown.

In the literature, personas [7] and scenarios [8] are used in order to define end users and their needs. Their usage is becoming more and more popular, especially in Software and System Engineering and Human Computer Interaction (HCI) [9]. Our hypothesis is that scenarios and personas can be used to develop and improve shared understanding of functional requirements between co-designers. In order to test this hypothesis, an empirical study has been undertaken in a laboratory context. This paper focuses on the protocol of the study and discusses our ability to find indicators of share understanding.

The paper is organized as follows. In the next section the theoretical background of the research is explained in detail. Section three discusses how the empirical study was designed and conducted. Indicators of shared understanding are discussed in the last section.

## *Shared Understanding of Requirement*

Cooper and Kleinschmidt [10] underline that, for the product success, during the product definition phase there must be an agreement on: (1) the target market, (2) the customers' needs, wants, and preferences (3) the product concept and (4) product's attributes, features, specifications, and requirements. However, it is an inevitable natural occurrence that there are different points of view between design actors in the way they interpret the users and their needs. Then, the design actors have to clarify their views and build a shared understanding.

In a design process, design actors bring with them their own beliefs, responsibilities, language, interests, jargon, and knowledge to the design team. As a result of this, each actor might see the design object differently within different perspectives. This is described as object world by Bucciarelli [11], personnel mental model by Badke-Schaub et al. [12] or as perspective, heuristics and interpretation by Page in his model of diversity [13]. All of them referring to distinct theoretical background but addressing the differences in which the members of a team can see the world. Bucciarelli's work comes from a social background, and the term 'object world' describes the assemblage of social, technical and symbolic components that make design possible within specific engineering domains. The Mental models "are simplification of the world" and "internal working models of the world" [12]. Thus, a mental model includes the three components of the diversity model of Page: perspective, heuristics and interpretation. In Page model, perspective has then a restrictive definition as "a map from reality to an internal language, such that each distinct object situation, problem or event gets mapped to a unique word" [13, p. 31]. Even if diversity and existence of diverse perspectives can be expected for creative problem solving, it is admitted that teams should "share at least some aspect about the task and the team" [12]. Badke-Schaub for example claims that "there are strong indices that shared mental models have an impact on creative problem-solving" (p. 10). Moreover, the importance of shared understanding in the performance of a design team had been highlighted in design research (see [14] for an updated literature review). The necessity of constructing common ground within the team to facilitate collaboration is acknowledged in the literature. Detienne [15] highlights the importance of creating negotiation mechanisms and grounding activity in order to manage the multiple perspectives in design groups.

The notion of common ground represents the knowledge that actors have in common and their awareness of this uniformity. Clark and Brennan [16] state that effective communication requires grounding activity. The grounding activity helps design actors to co-create the shared representation of the current situation of the problem, solutions. In a psychological perspective, authors refer to construction of team mental models, built on the communication based on individual mental model [12].

This process of building a common ground is also referred to as a framing cycle [6] or as managing multiple perspectives by Detienne [15]. The frames are defined as structures of belief, perception and appreciation that guide one's way of viewing and attempting to solve it. The framing cycle consists of making individuals' perspectives explicit, making conflicts salient, and building a common ground. In the following work, this construction of a shared understanding is detailed into two sub-sections: (1) perspective clarification, where the perspectives are explicated and (2) convergence: which is building a common frame.

### *Perspective Clarification*

As mentioned above, it is important that design actors externalize and communicate their frames. In the literature, the creation of common ground is reasoned to improve the effectiveness of communication. Stumpf and McDonnell [17] claim that: "the team's interaction to share frames provides a legitimate indication of the quality of team processes". Visser [18] also underlines the importance of creating a common ground during the co-designing activity, with her words: "It is then essential that designers, who each also have their personal perspective, establish a 'common ground'". Then, the design actors can create shared representations, which "concern agreements, especially on the definition of tasks, states of the design, references of central notions, and weights of criteria and constraints".

Different mediums of communication may be used for accomplishing this purpose, such as, conversation or sketching. For example, in conversation, the aim is to ensure that what has been said has been also understood. Creation of a common vocabulary can greatly improve perspective clarification within the team [19]. Conklin et al. [20] argue that shared displays are also beneficial to clarify the disagreements in a group: "When ideas and concerns are mediated via a shared display, challenges to positions assume a more neutral, less personal tone. It helps participants clarify the nature of their disagreement". The roles of mediations in this perspective clarification has been shown in many studies [4, 11, 21–24].

### *Convergence*

It is probably inevitable that there are disagreements within a design team. Positively, the divergence of opinions can stimulate creative ideas and solutions to the problems [25]. The diversity can enriched the results of the team [13]. The task conflicts can enforce team members to realize deeper analysis, which can increase learning and development of new and creative insights, and lead team to be more creative [26]. However, when the conflicts are not managed effectively, they can slow decision-making and keep members away from concentrating on the real task. They can also increase tension between team members, and cause interpersonal conflicts that can be detrimental to the creativity process [25]. Thus, a design team, which desires to reach an acceptable conclusion to their design task, has to find ways of resolving, or perhaps avoiding their conflicts [27].

As a traditional point of view, generally accepted opinions are chosen through negotiation. The notion of negotiation describes the way that the design actors reach agreement, which is based on argumentation [15]. With argumentation, the designers try to “convince themselves and their peers of the sense and validity of a particular solution, or of the necessity to respect a particular constraint related to the problem” [28]. As Détienne mentions, negotiation does not force a person to accept an argument but the conversation, which covers the arguments for and against a frame, makes it possible to get an agreement. The measure of agreement is quite difficult. In [12], authors claim that “the measurement of team mental models should reveal the degree of convergence among team members”. They explore propositions of the literature to access and measure mental models. External representations, interviews, team observation and graphics are used to access to mental models. In the next section, we will detail how we propose to use external representation to find indicators of convergence and perspective clarifications without tracking complete mental model, following the works on intermediary objects a mediation of design activity and the roles of objects as external representations of future product or design problems [21].

## *Scenarios for Shared Understanding of Functional Requirements*

Despite their popularity, there is no common definition of what the term “scenario” means, their use also varies widely in different design contexts. In this paper, the term scenario is used in the same sense with Carroll [8], stories about people and their activities. Each scenario includes the setting, agents/actors who have specific goals/objectives and sequences of action and events [8]. In the early stages of the design process, talking about the end users and their actual activities allow designers to elaborate the requirements, analyze and prioritize them. They also guide the projected scenarios, which explain the future activities, after the creation of the new product. In that way the designers evoke new views on defined needs and define new ones. In other words, scenarios are used to help designers to focus on end users and their activities and how these activities may be changed because of a new design. They serve as a communication tool between designers.

In the literature, even if the focus is on end users and their needs, some researchers prefer to use vague definitions of end users while building scenarios. For example, in Carroll’s scenarios [8], we do not see the detailed description of the users; generally just a name or the job description. However, Cooper [7] argues that by focusing on the behaviors and the goals of specific end users, the designers can satisfy a particular class of users with similar goals. Cooper proposes the utilization of personas-representative user archetypes-, to provoke common sense of categories of end users. Personas are fictional people who have names, details, and goals. They may be presented in their working and/or living environments and tied to particular activities that they are practicing. Cooper’s “goal-directed design” focuses the design effort for achieving persona goals, which covers the goals of the target market. Cooper points out that, once personas have been created then scenarios can be constructed around them. They are used to improve the power of scenarios. Pruitt and Grudin [29] argue that scenarios are less engaging and difficult to memorize when not built on personas. They also mention that personas help to prioritize functions for a product development cycle and facilitate decision-making process [29]. On this basis, we formulate the hypothesis that scenario and persona usage might encourage a shared understanding of functional requirements within a design team. In order to test this hypothesis two questions are posed:

1. How to test if the design actors converge through a shared understanding of the requirements during a design meeting?
2. How to evaluate if the scenarios and personas are effective in creating shared understanding between design actors?

The following section will present the empirical study we had carried out to answer this questions.

## Design of the Empirical Study

### *Design Situation Observations*

Video recording is often used in order to observe and understand the design activity [30]. Audio and video captures make it possible to ensure reliable analysis. Hicks et al. [31] proposes a process model in order to realize a structured observation. This is an iterative approach that involves five main phases: (1) Monitor, (2) Capture, (3) Analyze, (4) Prepare, and (5) Intervene. In the monitoring phase the researchers define what will be monitored during the design activity: the actors, their interaction, the objects, etc. The technology and the tools that will be used for monitoring are also prepared in this phase. The inputs, outputs, content and relationships between activities and interactions are then captured in the second phase. In the third phase the data is analyzed and interpreted. The last two phases are respectively the preparation of new tools or methods that will have the impact on the activity and ensuring that those interventions are beneficial.

It is admitted that each situation of design is unique and context embedded. Comparing and analysing design situations can be difficult. Visser tends to highlight the generic characteristics of design and different forms of design that appears in different design situations [32]. Hicks et al. also recommend a fine definition of the design situation to perform observation. According to Prudhomme et al.'s model [28], a design situation contains four main elements: task, actor, object and environment. A design task expresses a goal and the conditions in which work should be realized, whereas the design object, or the product is the entity on which designers work. The design actors are the people who are involved into design process. Finally the environment element is described by the industry, the available technical means and the project organization. This model gives a macroscopic view of a design situation. By taking this model as a reference, the relevant considerations that have to be addressed in an observational research can be defined. In this research we realized an empirical study in a laboratory environment, which is based on the Hicks et al.'s process model. The detail of this study is presented in the next section.

## *Framework of the Empirical Study*

The empirical study is built in different steps, with two groups of designers. A control group, which won't use the scenarios and personas as a method during the design meeting, was used. Comparing the results of the control group (referred as group A) and experiment group (referred as group B) might help us to evaluate the effectiveness of scenarios and personas.

The main design task is completed by individual tasks that allow extracting individual external representation of requirements. Consequently, three main steps were defined for the empirical study:

Step 1: An individual step in which each participant builds his own representation of the product specifications.

Step 2: The design meeting step: during this stage the subjects elicit the functional requirements collectively. While the experiment group is asked to use scenarios and personas, the control group is free in the choice of a working method.

Step 3: A second individual step to make explicit the participants' representations of the product after the group meeting. The same representation media than in the first step has to be used to make the comparison easier.

In addition, because the participants have to be prepared for the study, a preliminary training phase is required. In these preliminary steps, the participants are trained to the tools and methods that they will use during the study.

A set of 4 experiments had been run in French and in English. Some of the experiments were conducted by other researchers to validate the replicability of the protocol. Not all of these experiments have been analysed today, and in this paper we focus on one of the experiment realized in U.K. For this reason we can consider this paper more than a case study as it is studying only one experiment even if this case is part of a larger experimental protocol.

## *The Design Situation*

### Actors

In this paper, the design actors are the subjects who participated in the study. Because our focus is to analyze the collective activity, we had to use more than one participant. Studies showed that in functional analysis teams with more than 5–6 people, tend to be divided into small informal groups with only a core of 3–4 people doing real work [33]. So, we decided to use 4 participants in each of the two groups (group A and B). The composition of the groups was configured to be as similar as possible using PhD students and postdoctoral research engineers with engineering degrees and similar levels of experience. All participants were volunteers and were not remunerated. They were not informed about the research question. They were told their collective activity would be observed and recorded (video and audio) as a part of the study. However, after the experiment a presentation was given to explain the research context and answer to the participant' questions.

### Task

The focused domain of the research is industrial design. We chose to construct a design meeting typical of the early stages of a new product design process, during which design actors elicit functional requirements of the new product. It was also decide to limit the duration of this meeting up to 90 min.

Before the design meeting, the participants are given some time to think individually about the product idea. They are asked to represent the product idea in the form of a 5W table (When, Why, Who, What and Where), which provides information regarding their individual perspectives about the product specifications. The question How was eliminated from the original 5W1H approach because it could possibly focus the subjects on the technical possibilities, hence limiting their perspectives. However, the aim of this step was to focus subjects on generating alternative solutions not creating a specific one (divergence). The duration of this was fixed to 20 min. Again, after the design meeting the participants are asked to fulfill a 5W table in order to see if the discussions changed their individual perspective about the product specifications.

During the design meeting, they are asked to elicit the functional requirements collectively in the form of a Function-Criteria-Level (FCL) table. As mentioned before, while the group A was free to use any method(s) they felt appropriate for defining the functional requirements, the group B was required to use the scenarios and personas.

## Object

The product to be worked on was a "digital calendar". The product idea was chosen from an open innovation web platform. This site allows its visitors to submit new product ideas, to commit arguments, or to make commentaries about product ideas. We had three reasons for choosing this design object: (1) We had the opportunity to analyze from the platform the discussions between the various contributors to this product idea, which gave us a possible list of requirements, making it possible to test the acceptability of the experiment; (2) As subjects have a very limited time for achieving the design task, materials have to be simplified. So, we have chosen a product idea for which the participants may feel familiar with and contribute to easily. Nevertheless, we checked that all our subjects were naïve regarding this product idea; (3) The idea was pointed out as the most popular one on the site, so that we think that it can be interesting for the subjects to work on it.

## Environment

As mentioned in The Design Situation, the environment element is described by the industry, the available technology and project organization. In this research, because the study was realized in a laboratory layout, the industry was not considered.

The available technology for the subjects during the design meeting was limited with the supplied facilities. During the before and after steps of the design meeting, in order to realize their individual tasks, each subject was provided with a computer. The previous research on sketches shows that they play an important role in design process. As Ferguson [34] states: *Many features and the qualities of the objects that a technologist thinks about cannot be reduced to unambiguous verbal descriptions: therefore, they are dealt with in the mind by a visual, non-verbal process*. Thus, the subjects were also supplied with some draft papers and pens in each step, in order to allow them to sketch or write freely.

During the design meeting, both of the groups were provided with a computer for completing FCL table. Group A was also provided with a whiteboard and board markers that they might use to apply their methods, while as the group B was supplied with another computer in order to create the personas and scenarios in PowerPoint format. Because they had a limited time, group B was asked to use media which is easy to create and manipulate such as text or storyboards. Thus, they were also supplied with a set of pictures selected randomly from Google's image library (which were rooms of a house, an office and a selection of faces) that might be used for scenario and persona creation.

In terms of the project organization, within each group, one of the subjects was proposed as the manager of the design meeting according to his/her previous experience of managing. His/her role was to manage the time, ensure that the tasks would be realized and organize the relationship between the subjects. The choice of a manager may have positive or negative effects, which is not within the scope

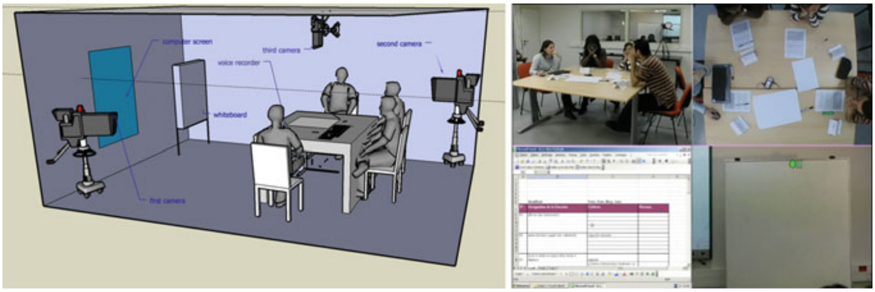


Fig. 1 Observatory room view and 4-PIP (group A)

of this paper. Otherwise, all the subjects had all equal rights during the meeting. The subjects were trained before the experiment with the aid of pre-prepared material. This included a document containing the explanation of the tools and methods (in additional to the information supplied to group A, the group B was informed about scenarios and personas), and examples of their usage. A formation document was also prepared for the manager in order to explain his/her responsibilities. Both of the documents were sent to subjects three days before the experiment via e-mail.

## *Observation Protocol*

### Monitor

The design meetings of the subjects were video and audio taped. Therefore, an observatory room was prepared, equipped with video and audio recording facilities. A voice recorder (placed on the table) and three movie cameras recorded the design activity (see Fig. 1). The movie cameras were installed to capture different views: a close view of the subjects when sitting at the table, the whiteboard and overhead view of the table. They were fixed and were not moved or repositioned during the session for not disturbing the participants. The experimenter and the recording equipment were situated in a neighboring room that the participants could not see. For group B, the same experiment layout was used with the difference that the movie camera recording the whiteboard was removed.

### Capture

The outputs of the each step were captured for the analysis phase. The same process was also followed for group B with the exception described below in step 2. The different steps of the empirical study are captured as follows:

Step 1: (20 min) The 5W tables, the sketches, and the rough drafts are captured in this step.

Step 2: (1 h 30 min) During the meeting the three camera views and the computer screen are recorded and mixed into one 4-PIP (four pictures in picture) combined view (see Fig. 1). A time stamp of the date, the time in hours, minutes and seconds is included in the video image. Group B's 4-PIP combined view contains two computer screens (one for the FCM table and one for the scenarios and personas' computer) and two camera views. For group B we also capture the created files for defining scenarios and personas. All potential documents produced during this step are also captured.

Step 3: This step consists in three sub-steps:

Step 3.1:

(5 min) In this step the subjects are asked to rank the 5 most important requirements from the FCM table that they collectively created before. The aim was to know if the subjects of a same group would assign the same importance to the defined requirements. The individual ranking tables are captured.

Step 3.2:

(10 min) New vision of the problem: new 5W tables completed individually by the participants are captured.

Step 3.3:

(10 min) In this last sub-step the structured open question interviews were realised with the participants. There are three reasons for conducting these interviews: (1) to check if the participant is in agreement with the group results—i.e. to gather subjects' individual perspectives on the FCM table completed collectively; (2) to understand the argumentation behind their ranking table; (3) to get some comments and critics on the design tools and methods used during the meeting. The interviews were audio recorded for a later analysis.

## Analyze

This step of the process model defines how the data captured during the three main steps of our empirical protocol are analysed. This analysis is performed in two steps: evaluating the validity of the data, and answering to the research question.

The validity of the data is pointed out by Bryman [35] as a key issue in social research. The main types of validity are:

- The *internal validity*: which is concerned with the causal relationship between the variables and the gathered results. In this research, in order ensure the internal validity, we have to be sure that the control group did not also use the scenarios and personas as a method.

- The *measurement validity*: the measures gathered from the analyzing method have to be verified. In our study, the measures are often the result of a coding process of the raw data (video, transcripts, design deliverables, etc. To control the validity of this coding process, a double-coding is systematically performed, and the coding results of each coder are compared using the Cohen's Kappa Calculations. Cohen's kappa coefficient is a statistical measure used to quantify agreement between two raters for categorical items [36].
- The *external and ecological validity*: the external validity is concerned with the question of whether the results of a study can be generalized beyond the specific research context. On the other hand, the ecological validity of the study is concerned with the questions if the findings are applicable to people's everyday settings.

Answering to the research question is here based on a series of indicators that we defined in order to analyze the captured data. Some of these indicators are quantitative and some others are more qualitative. Some of them are based on an internal analysis, requiring an in depth analysis of the whole corpus of the meeting. Some other indicators stay on an external analysis and are only based on the before and after-meeting deliverables.

These different indicators are presented in the following section.

## Problem of Understanding Between the Design Actors

### *Convergence*

As a first step, we investigate whether scenarios and personas have an impact on the convergence of the participants to a common perspective in terms of the requirements. The level of convergence in each group is evaluated through three indicators in order to make comparisons.

#### Indicator 1: Similarity of the Ranking Lists

This indicator is defined using the Spearman's rho $(\rho)$ coefficient [37]. This coefficient is used to observe whether the participants agree to each other's view, as far as the importance of the functional requirements are concerned. The calculated value of Spearman's rho varies from $-1$ to +1 and makes it possible to compare two by two the ranking lists of each participant. A $-1$ Spearman's rho means a perfect negative correlation, while a +1 Spearman's rho means a perfect positive one. A correlation of zero means that there is no relationship between the two variables.

**Table 1** Spearman's rho for each pair of participants and group average correlation

| | Group A | Group B |
|---|---|---|
| Participants 1 and 2 | 0.08 | 0.98 |
| Participants 1 and 3 | −0.48 | 0.32 |
| Participants 1 and 4 | −0.48 | 0.57 |
| Participants 2 and 3 | 0.30 | 0.29 |
| Participants 2 and 4 | 0.30 | 0.62 |
| Participants 3 and 4 | 0.77 | 0.37 |
| Average | 0.08 | 0.53 |
| Strength of the correlation | Slight (positive) | Moderate (positive) |

Table 1 shows the results of these calculations for both groups. The last row gives the interpreted strength of the correlation following Landis and Koch's standards [38].

Thus, in group B which used the scenarios and persona approach, the ranking lists of the four participants are clearly more coherent than in group A. However, this result has to be cautiously interpreted since the total number of functions listed by each group is small (15 for group A and 12 for group B). This means that the probability that the ranking lists are coherent by chance is high.

## Indicator 2: Convergence of the 5W Tables

This indicator is based on the analysis of the individual 5W tables made by the participants before and after the meeting. The idea is to see whether their individual representation of the product converged to a common one. In addition, comparing the 5W tables before the meeting makes it possible to see if the participants already had common perspectives.

Since a similar idea might be expressed in different words by two participants, a coding schema was defined in order to associate a unique identifier to each of the different ideas in the tables. The coding of each table was carried out by two different coders in order to check the reliability of the schema. This double coding allows the calculation of Cohen's Kappa index [38] which measures the level of agreement between the coders. Table 2 shows the results of this calculation. These values are generally interpreted as a moderate (0.41–0.60) or substantial (0.61–0.80) agreement, making it possible to analyse the results of the coding.

These results are presented Table 3. In both groups, we observe a decreasing total number of ideas that could be a sign of convergence of the two groups after the meeting. The rate number of ideas shared by all the participants in the group B is higher than in group A; Even if this rate was already higher before the meeting for this group B, the increasing is significant.

This second indicator tends to show the potential impact of scenarios and persona on the convergence of the 5W tables. This result has to be confirmed by other experiments.

**Table 2** Cohen's Kappa index for the codings of the 5W tables

| | Group A | Group B |
|---|---|---|
| Before meeting | 0.62 | 0.60 |
| After meeting | 0.60 | 0.68 |

**Table 3** Number of ideas and sharing of these ideas before and after the meeting

| | Group A | | Group B | |
|---|---|---|---|---|
| | Total number of ideas | Number of ideas shared by all the participants | Total number of ideas | Number of ideas shared by all the participants |
| Before meeting | 56 | 4 (7 %) | 48 | 8 (16 %) |
| After meeting | 37 | 3 (8 %) | 33 | 9 (27 %) |

### Indicator 3: Convergence During Verbal Communication

This qualitative indicator is based on analysing the group discussions to see whether the scenarios and personas had an influence on the way that the requirements where discussed.

In group B, personas create a common reference, each participant being aware of whom they are talking about. For example, in the 55th min, participant 3 says "[…] because when her daughter's using it, she will be there to support her child as well." At this point, everybody knows exactly who the daughter is since she was previously defined as a persona.

Moreover, the participants used personas as a medium to communicate their viewpoints. In other words, they made reference to personas, while presenting their arguments. For example, to support his point of view that the product has to be portable, in the 40th min of the meeting, participant 1 refers to the persona Emily and creates a fragment scenario around her: "I think one of the criteria is portable, isn't it? So, she can take it into garden and play with her teddy."

They also evaluated requirements and make decisions by referring to the personas. For example, in the 12th min of the meeting, participant 1 eliminates the requirement "reminding the bills" by referring to a specific persona: "I think one of the usefulness of the calendar is to see when the bills are due and stuff like that. […] But he is not going to be using those sorts of things."

Moreover, usage of personas helped the participants to identify conflicting requirements. In other words, they realised that a requirement, which can be essential for a persona might be disturbing for another one. For example, while discussing about "sharing the personal planning with the other users of the calendar", participant 1 says referring to the son Clayton: "[…] He wouldn't use it then. If he knows that mother can see everything." A total of nine proposed requirements were eliminated in similar ways, because they were not appropriated to one or more personas. The participants clarified and also strengthened their arguments based on the personas' characteristics and on the scenarios created around them. Due to the fact that all the participants built personas collectively, this created common references and negotiation process was easier to take decisions.

In group A, we did not notice such constructions. The participants did not discuss about, who are the real users of the product. Moreover, there is no elimination of the requirements. The group just focused on listing the functional requirements. These primary observations, exhibited here by selected quotes, require deeper investigations for extracting quantitative elements from the observation. Metrics should also be found for this third indicator.

To summarize on the convergence issue, the first two indicators do not lead to a cut-and-dried conclusion, giving contradictory results, of at best showing slight differences which cannot be reliable due to poor-sized samples. On the other hand, the qualitative analysis of the transcript clearly shows a positive impact of the scenario and persona on the convergence of the participant to a common perspective.

## *Perspective Clarification*

### Indicator 4: Mutual Awareness

Because the participants can have a good mutual understanding without necessary sharing a single vision of the design, we also considered the potential awareness of their agreements/disagreements as an indicator. The latter is based on analysing the post-meeting interviews of the participants, and more particularly their answers to the questions of whether in their point of view, their ranking of the functions would be shared by the group. In the case that their answer was negative, they were asked to comment on the differences that could exist.

Table 4 presents their answers to these questions. In group A, two participants think that the ranking lists will be different, which is quite true (see Table 2). Participant 3 is undecided. Only participant 4 estimates that the group's ranking list are similar, except the one of participant 3. In contrast with his assumption, according to Table 2, participant 4 has a substantial positive correlation with participant 3. In group B, they commonly imagine that their lists are similar, at least the first three functions. Analysing their lists shows that two participants have a common list, but not the same ranking. It comes also that two participants have the same first three functions in their list. Moreover, two participants claim that functions F4, F6 and F1 are the fundamental ones and should be listed by everyone, which is quite true. On the other hand, participant 1 believes that his ranking order is different from other group members' one. He is partially wrong in the sense that he has an almost perfect agreement with participant 2 ($\rho = 0.98$). The differences between the two groups are too narrow to make a clear distinction.

**Table 4** Comments of the participants about their ranking of the functions

| | Group A | Group B |
|---|---|---|
| Participant 1 | No, I do not think it will be the same | No, I don't think so. Because they will add functions that will make the product more complex. I try to keep it simple |
| Participant 2 | Probably not | Yes, I think. Because there are fundamental functions: F6, F4 and F1 that will be common. In general we will agree |
| Participant 3 | I'm not sure | The first three will be the same. There are fundamental functions like F6, F1 and F4 |
| Participant 4 | Yes, I think so but may be participant 3 will not have the same one. The most important ones will be functions F1.1 and F2 | Yes, probably |

We consider that in both groups the participants were similarly aware and unaware of their disagreements and agreements.

## Conclusion and Next Steps

The results presented here are from the analysis of one experiment while we conducted four of them. This paper can be considered as a preliminary case study, even if we developed a replicable and reliable protocol to estimate the impact of scenario and persona on the shared understanding of functional requirements. A description of the design situation was proposed to facilitate the replicability and double coding was used to ensure the validation of the analysis. We proposed indicators to measure the share understanding of design actors within the team involved. Two Indicators are based on external representations produced by the designers, two indicators are based on conversation analysis and post meeting interviews.

The study shows that all the groups have difficulties to converge to a common ranking of the functions identified in the meeting. The correlations calculated, even when positive, are still moderate according to correlation coefficient. Our analysis shows that the Scenario and Persona usage influence positively the correlation. But the 5W analysis doesn't show a strong, unique and common description of the product to be designed. Even if the group agreed to a list of requirements in the FCM tables jointly produced, the individual rankings and the individual representations of the problems mediated by the 5W sheet, remain different. Similarly, the study shows that the groups have difficulties to clarify their perspectives. The groups are not especially aware of their agreements and disagreements.

As a limitation of the analysis, we can mention that the study doesn't allow to definitely concluding on the positive impact of scenario and persona on the shared understanding. The study should be expended, enlarging the number of situations observed to confirm the observations. The external indicators proposed to measure shared understanding have to be completed by the analysis of the discussions during the design meeting. The in-depth study of the verbal interaction is not presented here, but some preliminary results of the conversation analysis show qualitative differences between group A and group B. The Scenario and persona are used as argumentation and for negotiation. The persona serves in the elimination of some requirements. Thus the study shows also a significant increasing of the number of functions discussed during the meeting. We also observed that, in that phase of requirements elicitation, fragments of scenarios are always used by designers. But changes are observed from designer centered scenario based on designers' experience to scenario involving personas. Thus, other impacts of scenario and persona are highlighted by the study, but their complete analysis requires new indicators.

**Acknowledgements** Authors gratefully thank reviewers for their deep and relevant comments. This research had been funded by Region Rhône-Alpes.

## References

1. Conklin J (2006) Wicked problems and social complexity. In: Conklin J (ed) Dialogue mapping: building shared understanding of wicked problems. Wiley, New York
2. Ulrich KT, Eppinger SD (2004) Product design and development, 3rd edn. McGraw-Hill, New York
3. AFNOR (1996) NF-EN-1325-1 I-Value management, value analysis, functional analysis vocabulary ± Part 1: value analysis and functional analysis: Paris
4. Vinck D (2011) Taking intermediary objects and equipping work into account in the study of engineering practices. Eng Stud 3(1):25–44
5. Vinck D, Jeantet A (1995) Mediating and commissioning objects in the sociotechnical process of product design: a conceptual approach. In: COSTA3 workshop designs, networks and strategies. European Community
6. Hey JHG, Joyce CK, Beckman SL (2007) Framing innovation: negotiating shared frames during early design phases. J Des Res 6(1-2): 79–99
7. Cooper A (1999) The inmates are running the asylum, Indianapolis
8. Carroll JM (2000) Making use: scenario-based design of human-computer interactions. MIT press, Cambridge
9. Miaskiewicz T, Kozar KA (2011) Personas and user-centered design: how can personas benefit product design processes? Des Stud 32(5):417–430
10. Cooper RG, Kleinschmidt EJ (1993) Screening new products for potential winners. Long Range Plan 26(6):74–81
11. Bucciarelli LL (1994) Designing engineers. MIT Press, Cambridge
12. Badke-Schaub P et al (2007) Mental models in design teams: a valid approach to performance in design collaboration? CoDesign: Int J CoCreation Des Arts 3(1):5–20
13. Page SE (2007) The difference, how the power of diversity creates better groups, firms, schools, and societies. Princeton University Press, Princeton

14. Kleinsmann MJ, Buijs J, Valkenburg R (2010) Understanding the complexity of knowledge integration in collaborative new product development teams: a case study. J Eng Tech Manage 27(1–2):20–32
15. Détienne F (2006) Collaborative design: managing task interdependencies and multiple perspectives. Interact Comput 18:1–20
16. Clark HH, Brennan SE (1991) Grounding in communication. In: Resnick L, Levine JM, Teasley SD (eds) Perspectives on socially shared cognition. APA, Washington DC
17. Stumpf SC, McDonnell J (2002) Talking about team framing: using argumentation to analyse and support experiential learning in early design episodes. Des Stud 23(1):5–23
18. Visser W (2006) Designing as construction of representations: a dynamic viewpoint in cognitive design research. Hum–Comput Interact 21(1):103–152
19. Dong A (2005) The latent semantic approach to studying design team communication. Des Stud 26:445–461
20. Conklin J et al (2003) Facilitated hypertext for collective sensemaking: 15 Years, pp 123–124
21. Boujut JF, Laureillard P (2002) A co-operation framework for product–process integration in engineering design. Des Stud 23:497–513
22. Henderson K (1999) On line and on paper. MIT Press, Cambridge MA
23. Blanco E (2003) Rough drafts. Revealing and mediating design. In: Vinck D (ed) Everyday engineering: an ethnography of design and innovation. MIT press, Cambridge, pp 181–201
24. Boujut JF, Blanco E (2003) Intermediary objects as a means to foster co-operation in engineering design. Comput Support Collaborative Work 12(2):205–219
25. Chen MH (2006) Understanding the benefits and detriments of conflict on team creativity process. Creativity Innov Manage 15(1):105–116
26. De Dreu CK, Weingart RL (2003) Task versus relationship conflict, team performance, and team member satisfaction: a meta-analysis. J Appl Psychol 88(4):741–749
27. Cross N, Cross AC (1995) Observations of teamwork and social processes in design. Des Stud 16(2):143–170
28. Prudhomme G, Pourroy F, Lund K (2007) An empirical study of engineering knowledge dynamics in a design situation. J Des Res 6(3):333–358
29. Pruitt J, Grudin J (2002) Personas, participatory design and product development: an infrastructure for engagement. In: Proceedings of participatory design conference, pp 144–161
30. McDonnell J, LLoyd P (2009) About: designing–analysing design meetings.CRC Press, Balkema, p 422
31. Hicks BJ, et al (2009) An intelligent design environment: overcoming fundamental barriers to realizing a step change in design performance and innovation. In: ICED'09, Stanford University
32. Visser W (2009) Design: one, but in different forms. Des Stud 30(3):187
33. Fowler T (1990) Value analysis in design. Van Nostrand Reinhold, NewYork
34. Fergusson ES (1992) Engineering and the mind's eye. MIT press, Cambridge
35. Bryman A (2001) Social research methods. Oxford University Press, Oxford
36. Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Measur 20(1):159–174
37. Coolican H (1999) Research methods and statistics in psychology, 3rd edn. Hodder and Stoughton, London
38. Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33:159–174

# Exploring Designing Styles Using a Problem–Solution Division

**Hao Jiang, John S. Gero and Ching-Chiaun Yen**

**Abstract** This paper presents a measurement-based exploration of designing styles within the context of different design disciplines and tasks based on the design cognition of small design teams. Twelve final-year industrial design and twelve mechanical engineering design students were recruited to form teams of two. Each team undertook two conceptual product design tasks with different classes of requirements. Protocols of conversations and observations of design activities were then examined using an ontologically-based coding scheme. A problem–solution index was proposed to classify design sessions into problem-focused and solution-focused designing styles. Results suggest that industrial design student teams have a designing style that is more focused on the design problem than mechanical engineering student teams. The same design team may change its relative focusing on problem or solution in response to different classes of design requirements.

## Introduction

Design cognition is often modeled as a search process across two notional design "spaces" of problem and solution. However, there is not a consensus on how a designer's cognitive processes progress during the designing process. The problem–solving view of design claims that the designing process commences with an exploration within the problem space. Goel and Pirolli [1], for example, considered that designers would engage in an initial problem structuring phase before

---

H. Jiang (✉) · C.-C. Yen
National University of Singapore, Singapore, Singapore
e-mail: jiangh82@nus.edu.sg

J. S. Gero
Krasnow Institute for Advanced Study, Fairfax, VA, USA

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_5,

moving into solution development. An alternative school of thought argues that design thinking is primarily abductive or solution-focused [2]. Designers would jump into the solution space in the very beginning of the designing process, before the problem is formulated. In particular, designers need to employ conjectures of (partial) solutions to analyze the ill-structured or wicked design problems, i.e., "analysis through synthesis" [3, 4]. Differing from these two phase-based views, Schön [5, 6] modeled problem setting ("seeing") and problem solving ("moving") of design as a reflective conversation: designing progresses in cycles of "seeing-moving-seeing" rather than two distinct phases of problem analysis and solution development. This interactive view is echoed with a co-evolutionary model of designing [7, 8].

In addition to the debates about generic design paradigms, the tendency to focus on problem or solution is subject to the designers' experience level. Restrepo and Christiaans [9] and Kruger and Cross's [10] studies found that more experienced industrial designers usually follow a problem-focused strategy, which tends to produce better results. Lloyd and Scott [11], to the contrary, argued that more experienced engineering designers tended to focus more on solutions (higher percentage of generative mode actions) than those with less experience. These conflicting findings imply that designers' problem or solution focusing may be specific to design disciplines. No direct evidence can be found regarding this issue in the current literature.

In order to explore and compare designers' focus on either problem or solution spaces between different disciplines and/or circumstances, we need a new measurement independent of particular assumptions about how problem or solution focus is organized in the design cognitive process. The proposed measurement is developed within an ontological framework, directly capturing the meta-level structures of design cognition, i.e., the designing style of a design session, in terms of problem-focused and solution-focused design issues. This new measurement is then applied to a protocol study comparing industrial and engineering design students' design cognition behind two product conceptual design tasks.

## Ontologically-Based Protocol Analysis

The empirical basis for the measurement of problem- and solution-focused designing styles comes from protocol analyses of design sessions that are coded using design issues based on the function-behavior-structure (FBS) coding scheme. A general design ontology, namely, the FBS ontology [12, 13], models designing in terms of three basic classes of ontological variables: function, behavior and structure. It creates a useful ground for interpretation of design cognition across different design domains. A domain-independent coding scheme, based this ontology, has been developed with six design issues: function (F), expected behavior (Be), behavior from structure (Bs), structure (S), requirement (R) and design description (D), Fig. 1.

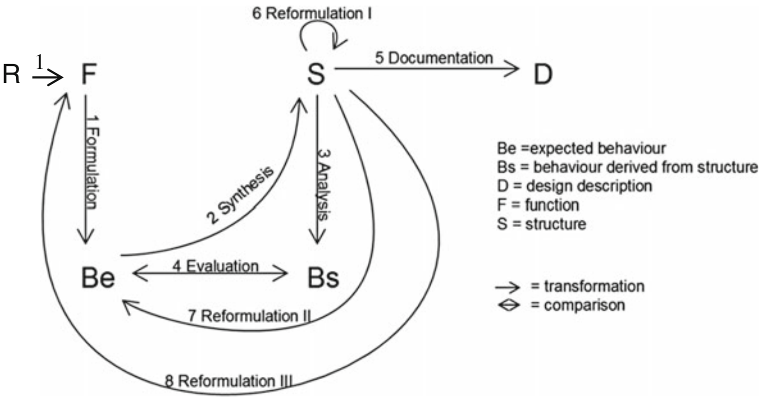


**Fig. 1** The FBS ontology [13]

**Table 1** Mapping FBS design issues onto problem and solution spaces

| Problem/solution space | Design issue |
|---|---|
| Problem space = Problem-focused design issues | Requirement (R)<br>Function (F)<br>Expected behavior (Be) |
| Solution space = Solution-focused design issues | Behavior from structure (Bs)<br>Structure (S) |

In the FBS ontology, problem formulation mainly involves reasoning about requirement, function and expected behavior, while reasoning about structure and behavior from structure are related to artifacts as a solution to the formulated problem [12, 14]. The ontologically-based design issues can be categorized into problem-focused and solution-focused design issues as shown in Table 1. The FBS-based coding scheme does not code the design description issues with this view of problem and solution spaces. Description issues are therefore excluded in the exploration of problem- and solution-focused designing styles.

## *Problem–Solution Index for the Whole Design Session*

The designing styles, or the meta-level structure over the cognitive processes behind design activities, can be explored at various levels of granularity. It may refer to an overall characteristic of the entire design session, or a "signature" of a dynamic process that describes the time-based changes of the designers' cognitive focus.

The simplest form is to use a single-value measurement to summarize the designing style of an entire design session. The problem–solution (P–S) index is proposed as a ratio measurement, computing the ratio of the total occurrences of the design issues concerned with the problem space to the sum of those related to the solution space, Eq. (1).

$$\text{P}-\text{S index} = \frac{\sum(\text{Problem}-\text{related issues})}{\sum(\text{Solution}-\text{related issues})} = \frac{\sum(R, F, Be)}{\sum(Bs, S)} \qquad (1)$$

The P–S index value quantifies the relative focusing on problem or solution. We define a design session with a P–S index larger than 1 as one with a problem-focused designing style, and a session with a P–S index value less than or equal to 1 as one with a solution-focused style.

### *Sequential Problem–Solution Index Across Design Session*

Designing is a dynamic process. A single value for the P–S index for the entire session will collapse any time-based changes. A fractioning technique is incorporated in this study to tap into the designing styles within a design session [15, 16]. A preliminary exploration of the dynamic nature of designing styles can be undertaken in two halves of the design protocols. The finer subdivision of the design session eventually leads to a sequential P–S index delineating the trajectory of cognitive progression during the design session.

Here the entire design session is divided into 10 consecutive non-overlapping sections each with an equal number of design issues. The P–S index for each section is calculated. A sequence of temporally ordered P–S indexes is read as a "signature" of dynamic designing style.

## A Comparative Study Applying the Problem- and Solution-Focused Measurements

The proposed measurements for problem- and solution-focused designing styles are then applied to a protocol study comparing final-year undergraduate industrial design (ID) and mechanical engineering design (ME) students' design cognition in various classes of conceptual design requirements [17]. The design experiment used 3 × 2 mixed-model factorial design. The design discipline is a between-subjects factor. The type of design tasks is a within-subjects factor. Each design team undertook two conceptual design tasks in this study. Four main hypotheses related to students' problem- and solution-focused designing styles are proposed to be tested in this research:

1. All design sessions demonstrate a common pattern of designing behavior, independent of the variables of discipline and type of task;

2. There are significant inter-disciplinary differences between the designing styles of ID and ME teams;
3. A design team's designing style will be different when designing for different classes of design requirements; and
4. There are two mutually exclusive hypotheses about the designing style of mixed teams consisting of both ID and ME participants:
    - 4a. Mixed teams will exhibit a designing style that is the average of ID and ME teams' designing styles; or
    - 4b. Mixed teams will exhibit a designing style that is different from either ID or ME teams' designing styles.

## *Participants*

Twenty-four final-year undergraduate design students (12 ID and 12 ME) from the National University of Singapore participated voluntarily in this research. All participants had finished the taught courses and were involving in their final-year projects at the time of the experiment. All of them had at least three years of design exercises/projects experience and some had intern experience in design firms. According to the pre-test questionnaire and follow-up interviews, they all claimed to have above-average design expertise among their classmates. Five ID students were award winners of international/regional design competitions.

The unit of experiment was a two-person design team. Literature has identified that, to deal with the increasing complexity of contemporary context, product design had shifted from predominantly individual activity towards predominantly team-oriented activity [18]. Two participants, either from the same discipline or different ones, were paired to work collaboratively in two conceptual design exercises. The factor of design discipline therefore has 3 states: ID, ME and Mixed teams.

## *Experiment Tasks*

Two conceptual design tasks were used based on Keinonen's taxonomy of product development concepts and visionary concepts [19]. The first task, Task CM, was to design a coffee maker for the existing market. It simulated a typical initial stage of a normal new product development (NPD) process. Designers were expected to consider practical factors related to a NPD project, e.g., market and user analysis, supporting technology and resources.

The second task was to design a next-generation personal entertainment system/device for the year 2025 (Task PES). It was a visionary task beyond the normal NPD time frame and with open-ended requirements. In terms of the three-pronged nature of a design problem that consists of determined, undetermined and underdetermined elements [20], Task PES faced a very limited amount of determined/unalterable factors. Designers were expected to use design concepts as a tangible means to explore future scenarios.

### *Ontologically-Based Protocol Segmentation and Coding*

The participants' conceptual design activities (including conversations and gestures) were audio and video recorded, and then segmented and coded using the FBS ontologically-based protocol coding scheme [15]. Each segment in a FBS-coded protocol was strictly assigned only one of the six design issues. There are no overlapped or multi-coded segments. If an utterance was identified to contain more than one design issue, it was further segmented. Those utterances that did not fit in any of six the FBS categories were marked as "other" (O). These non-design issues were discarded in the following analysis.

The FBS-based protocol segmentation and coding process involves subjective judgments. It used the Delphi method to minimize coder bias and improve the coding reliability [14, 21]. Each set of design protocols was separately segmented and coded twice. An arbitration process was then undertaken to resolve previous coding disagreements and improve the quality of final coding.

After arbitration, the re-categorization of the design issues into the problem-related issue and solution-related issue was automatically assigned based on the mapping shown in Table 1. Some coding examples are presented in Table 2.

## Results

As a consequence of the FBS-based segmentation and coding, the observation of design activities were converted into a sequence of design issues, and then problem- and solution-focused issues, which become the foundational data for the subsequent analyses of ID and ME students' designing styles.

### *Reliability of the FBS-Based Protocol Coding*

The use of the Delphi method results in three sets of FBS-coded protocols, i.e., two separate rounds of coding and an arbitrated coding. The agreement between the arbitrated and the second coding ($M = 89.1\ \%$, $SD = 1.7\ \%$) is significantly higher than with the first coding ($M = 80.6\ \%$, $SD = 3.7\ \%$) The improvement of reliability reflects the "process gain" enabled by the Delphi method. As the disagreements in the first two rounds of coding have been resolved in the arbitration

**Table 2** Coding examples

| # | Segmented protocol | Design issue[a] | P–S space[b] |
|---|---|---|---|
| 1 | We need to consider that this product expects to be launched in 3 years | R | $P_S$ |
| 2 | In terms of material, what do you think is the future trend | F | $P_S$ |
| 3 | This should be used like reading a book | Be | $P_S$ |
| 4 | So the smell can come out from here | Bs | $S_S$ |
| 5 | Here is a dispenser … | S | $S_S$ |
| 6 | (Draw a dispenser) | D | N/A |
| 7 | Should we plan out storyboard first or should we draw first? | O | N/A |

[a] The symbols for FBS design issues are the same with Fig. 1

[b] $P_S$ problem-related issue; $S_S$ solution-related issue; *N/A* not applicable

process, the following analyses are performed on the arbitrated protocols, which are considered to be the most reliable dataset.

## Descriptive Statistics

Each design session's occurrences of design issues were normalized by dividing them with the total number of design issues in that session. The normalized design issue distributions are shown in Table 3 and Fig. 2.

ID sessions had the highest percentages of function issues and then followed by the Mixed sessions and ME sessions. ID sessions also had higher percentages of expected behavior issues than these other two groups. ME sessions had the highest percentages of the solution-related issues and ID sessions had the lowest ones.

For inter-task comparisons, Task PES tended to have more function and expected behavior issues than Task CM, whereas the percentage of structure issues in Task CM was higher than that in Task PES.

## Problem–Solution Indexes Charactering the Whole Sessions

The values of P–S index for each design session are shown in Table 4. The problem-focused sessions are highlighted by bold fonts. The results are also plotted in Fig. 3, against a line at the value of 1.00 for the P–S index signifying the boundary between problem-focused and solution-focused designing styles.

ID PES sessions had significant higher P–S index values than other sessions, demonstrating a strong tendency of focusing on problem-related issues. The P–S index value of ID CM sessions are around the threshold of problem–solution division. The designing styles of the rest sessions were solution-focused, as indicated by the relatively low P–S index values.

**Table 3** Normalized distribution of design issues (%)

| Design issues Groups | | Requirement (R) | Function (F) | Expected behavior (Be) | Behavior from Structure (Bs) | Structure (S) | Description (D) |
|---|---|---|---|---|---|---|---|
| ID CM | Mean | 0.9 | 23.6 | 15.6 | 20.7 | 20.3 | 19.0 |
| | SD | 0.4 | 2.8 | 4.2 | 3.8 | 1.7 | 6.7 |
| ID PES | Mean | 1.5 | 28.0 | 23.1 | 15.7 | 13.2 | 18.7 |
| | SD | 0.6 | 3.9 | 2.9 | 2.4 | 3.3 | 4.4 |
| Mix CM | Mean | 0.9 | 17.9 | 12.7 | 27.4 | 21.5 | 19.6 |
| | SD | 0.4 | 7.9 | 1.0 | 6.6 | 3.0 | 2.3 |
| Mix PES | Mean | 1.5 | 17.3 | 14.4 | 27.2 | 18.0 | 21.6 |
| | SD | 0.5 | 2.9 | 3.6 | 8.4 | 4.5 | 5.2 |
| ME CM | Mean | 1.8 | 11.4 | 13.5 | 28.0 | 28.3 | 16.9 |
| | SD | 0.4 | 6.0 | 3.7 | 3.2 | 8.2 | 5.4 |
| ME PES | Mean | 1.1 | 12.1 | 15.6 | 31.2 | 19.8 | 20.1 |
| | SD | 0.4 | 2.9 | 6.2 | 7.2 | 2.9 | 7.0 |

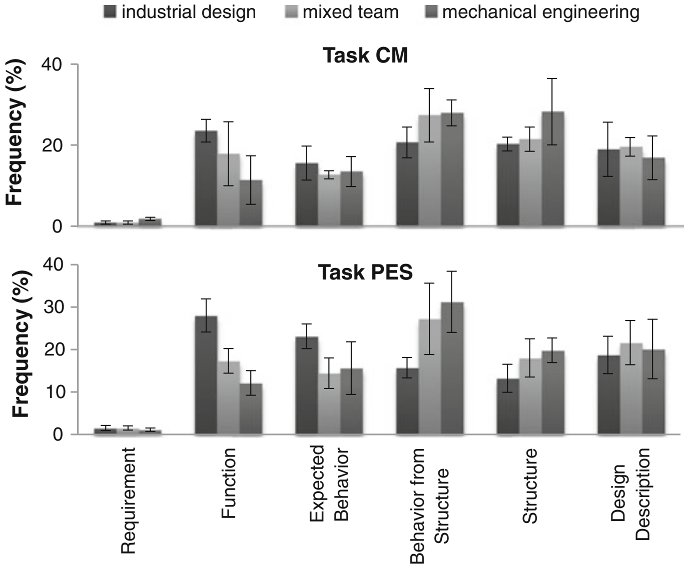


**Fig. 2** Frequency distributions of the design issues (%)

The aggregated P–S index values are presented in Fig. 4, against the boundary of the index value equal to 1. This indicates that ID teams generally focused more on the design problem than did the mixed and ME teams.

The designing styles of ID sessions were significantly different between the two tasks, from near equal focus on both problem and solution for the CM task to a

**Table 4** Values of P–S index

| Groups | Value of P–S index for each team | | | | Mean | Std. Dev |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | | |
| ID CM | 0.90 | **1.01** | **1.13** | 0.88 | 0.98 | 0.11 |
| ID PES | **2.04** | **2.32** | **1.74** | **1.40** | 1.88 | 0.39 |
| Mix CM | 0.95 | 0.36 | 0.89 | 0.53 | 0.68 | 0.28 |
| Mix PES | 0.48 | 0.76 | 0.77 | **1.01** | 0.76 | 0.22 |
| ME CM | 0.28 | 0.34 | 0.55 | 0.80 | 0.49 | 0.23 |
| ME PES | 0.56 | 0.75 | 0.46 | 0.54 | 0.58 | 0.12 |

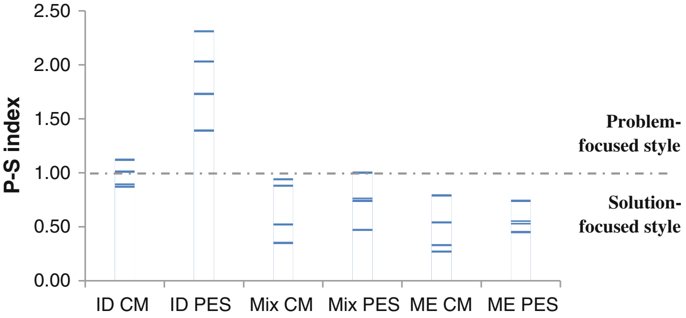


**Fig. 3** Values of P–S index and designing styles

highly problem-focused designing style for the PES task. The mixed and ME teams did not show this behavior.

## *Problem–Solution Indexes in Fractioned Design Protocols*

The division of the entire session's design protocol into two halves provides a preliminary understanding into the development of design cognition within a design session, Fig. 5. Mann-Whitney U tests and Wilcoxon Signed Ranks test were then applied to produce inter-session and within-session comparisons.

### Inter-session Comparisons

The inter-session comparisons of P–S indexes between different disciplinary teams and between different tasks are summarized in Table 5. In the first half of the design sessions' protocols, each disciplinary design team showed a similar

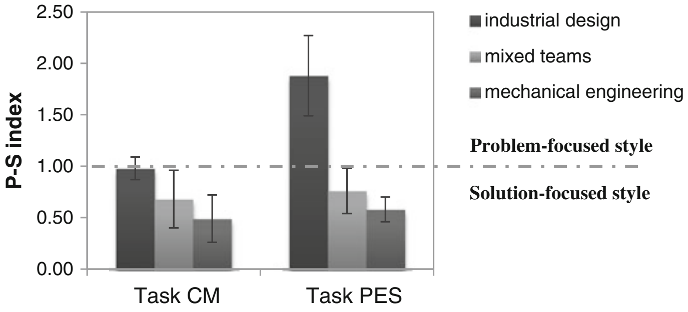


**Fig. 4** Aggregated P–S index values

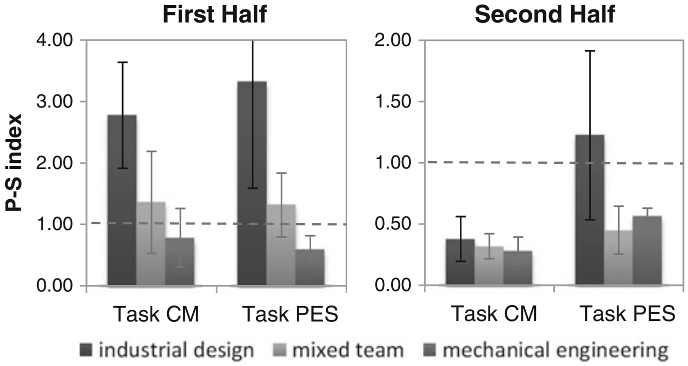


**Fig. 5** P–S indexes in two halves of a design session

**Table 5** Inter-session comparisons of P–S indexes in the fractioned protocols

| Group | First half of design session | Second half of design session |
|---|---|---|
| ID teams | ID CM $\approx$ ID PES | ID CM $<$ ID PES |
| Mixed teams | Mix CM $\approx$ Mix PES | Mix CM $\approx$ Mix PES |
| ME teams | ME CM $\approx$ ME PES | ME CM $<$ ME PES |
| Task CM | ID CM $>$ Mix CM $\approx$ ME CM | ID CM $\approx$ Mix CM $\approx$ ME PES |
| Task PES | ID PES $>$ Mix PES $>$ ME PES | ID PES $>$ Mix PES $\approx$ ME PES |

$\approx$ not significantly different
$>$ significantly larger than
$<$ significantly smaller than

designing style between the two tasks. When progressing to the second half of the design session, ID and ME teams significantly focused more on problem in the Task PES than they did in the Task CM. The P–S indexes of the mixed teams' sessions were not significantly different between the two tasks.

Figure 5 and Table 5 also indicate that, in the second half of Task CM sessions' protocols, there were no significant differences between ID, mixed and ME teams' P–S index values. For the remaining halves, the ID sessions had a much higher P–S index than the other two groups. The mixed teams' P–S index was also significantly higher than ME teams in the first half of Task PES sessions.

### Within-session Comparisons

Comparing the first and second halves of design sessions' P–S indexes indicates that all design sessions' cognitive focus in the first half was more on the problem than in the second, with the exception of the ME PES sessions, Table 6.

## *Sequential Problem–Solution Division and Indexes*

After subdividing the entire design session into 10 non-overlapping sections, most groups' sequential P–S division was found to have a significant negative correlation to the fractioned protocol sections, Table 7. As the percentage of problem-related issues and P–S index are two representations of the same set of measurements, their correlation coefficients with time and the *p*-values were the same. The decreasing focusing on problem, measured in sequential P–S indexes, was consistent with the P–S indexes in two halves of design session, Table 6. ME PES sessions' decreasing trend was not statistically significant.

The mean sequential percentage of problem-related issues can be plotted against their fractioned sections along with the boundary of the P–S division of 50 %, Fig. 6. The corresponding P–S indexes are presented in Fig. 7. These two figures can be read as "signatures" of design dynamics over a design session.

## Preliminary Test of Hypotheses

This section presents a qualitative test in response to the four main hypotheses about whether designers' tendency to focus on problem or solution is affected by design teams' disciplinary backgrounds and the nature of design requirements.

**Table 6** Comparing P–S indexes between the first and second halves of design sessions' protocols

| Task | Task CM | | Task PES | |
|---|---|---|---|---|
| Discipline | Z statistics | *p*-value (1-tailed) | Z Statistics | *p*-value (1-tailed) |
| ID teams | −1.826 | 0.034* | −1.826 | 0.034* |
| Mixed teams | −1.826 | 0.034* | −1.826 | 0.034* |
| ME teams | −1.826* | 0.034* | −0.365 | 0.358 |

* $p < 0.05$

**Table 7** Correlation between sequential problem issues and fractioned sections

| Group | Spearman's rho | Sig (2-tailed) | Slope (Problem-issue %) | Slope (P–S index) |
|---|---|---|---|---|
| ID CM | −0.914 | 0.000* | −0.090 | −1.586 |
| ID PES | −0.732 | 0.000* | −0.050 | −0.641 |
| Mix CM | −0.724 | 0.000* | −0.056 | −0.425 |
| Mix PES | −0.629 | 0.000* | −0.048 | −0.254 |
| ME CM | −0.518 | 0.001* | −0.043 | −0.124 |
| ME PES | −0.218 | 0.176 | −0.015 | −0.092 |

$p < 0.005$

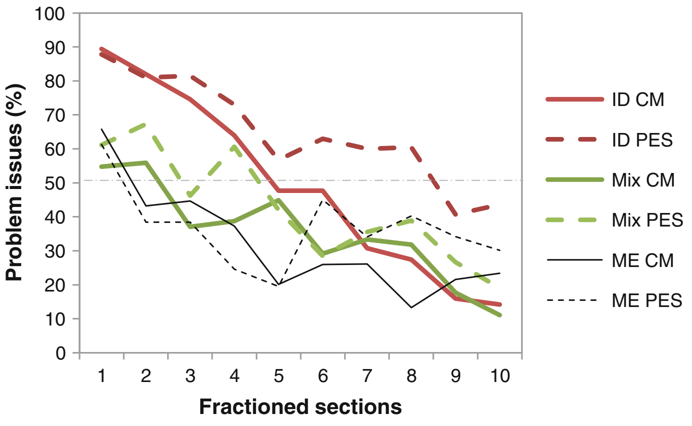


**Fig. 6** Sequential distributions of problem-related issues (%)

## *General Trend From More Problem-Focused to More Solution-Focused*

Hypothesis 1 is that there are similarities between design sessions in terms of the designing style, which are independent of specific design disciplines and tasks. Figures 6 and 7 demonstrate that, despite fluctuations, designers' cognitive focus

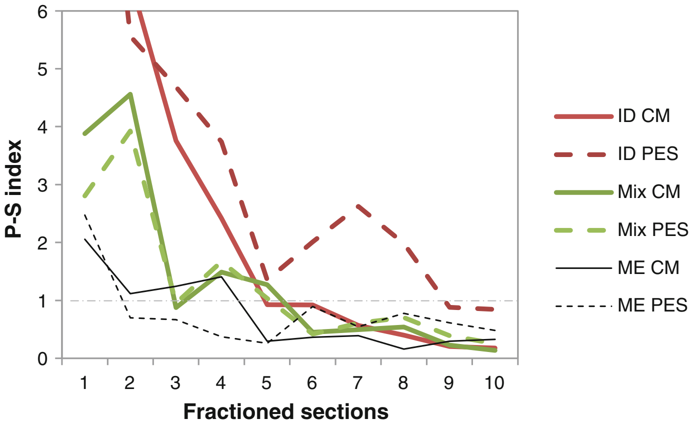


**Fig. 7** Sequential P–S indexes along the fractioned 10 sections of session

on design problems decreased along with the progression of designing. All session's linear estimation lines for their sequential P–S division and P–S indexes were negative, Table 7.

This supports the argument that there is a regularity in design cognition transcending specific parameters of designing [15]. However, the decreasing focus on problem is a relatively weak pattern. Some sessions only demonstrate a small tendency to move more towards a solution-focus, as their correlation with the fractioned sections was not significant, Tables 5 and 7.

## *Inter-disciplinary Differences*

Hypothesis 2 is that teams with different disciplinary backgrounds result in different designing styles. According to Restrepo and Christiaans [9], Kruger and Cross [10] and Lloyd and Scott's [11] findings, ID teams may focus more on problem-related issues and ME teams tend to be more solution-focused. The evidentiary support for this hypothesis is demonstrated by both the single-value P–S indexes and sequential P–S division.

In both tasks, as shown in Fig. 4, ID teams have the highest overall P–S indexes and ME teams present the lowest values. Figures 6 and 7 show that the curves of the ID sessions are generally above the curves of the ME sessions.

In general, ID teams' designing style focuses more on the design problem, and ME teams have a more solution-focused style. However, the identified inter-disciplinary differences do not have the same significance between the two tasks. Figures 5 and 6 show that ID teams demonstrate two distinct designing styles between the two tasks. The problem-focused designing style dominated in the ID PES session, and they gradually shifted from the problem-focused designing style to a solution-focused one during the Task CM sessions. ME teams, to the contrary, show a strong solution-focused designing style for both tasks. They give problem-related issues an equal emphasis to solution-related ones in the beginning of the design sessions, Figs. 6 and 7. This result implies that ID teams may lend themselves more to a designing style change in response to the change of design tasks, whereas ME teams adopt one designing style to cope with different classes of requirements.

### *Inter-task Differences*

The test of Hypothesis 2 has already involved Hypothesis 3, i.e., the designing style of one team may change in accordance to the requirement changes. Table 4 and Fig. 4 indicate that ID and ME teams both show a higher P–S index in Task PES than Task CM. Figure 5 and Table 5 further show that the inter-task differences occurred in the later stages of design sessions; for each disciplinary team, there was no significant difference of P–S indexes in the first half of the design session.

The best-fit line of Task PES's sequential P–S division is also found to have a smaller slope value than that of Task CM, Table 7. The decrease of problem focus progresses at a slower rate in Task PES. The graphs of sequential P–S division, Figs. 6 and 7, show that Task CM and Task PES of ID and ME teams each start with a similar focus on the problem, but their designing styles differ as the design session progresses. In the latter part of the design sessions, designing for Task PES, the focus on problem is relatively larger than that of Task CM.

A qualitative assessment of video recording and transcripts [17] indicates that the formulation of design problem is revisited periodically in the latter episodes of Task PES session. The ill-defined, open-ended design task may require more effort on problem reframing. The same behavior is rarely observed in the relatively well-defined Task CM.

### *Averaging ID and ME Styles in the Mixed Teams*

Hypotheses 4a and 4b are mutually exclusive, i.e., whether or not Mixed teams will exhibit a designing style that is the average of ID and ME teams' designing styles. Tables 3, 4 and Fig. 4 show that in most measurements of Mixed teams, the design issues, P–S index value and best-fit line slope of sequential P–S division, the behavior of Mixed teams is always between that of ID and ME teams.

Figures 6 and 7 show finer-grained results about the time-based change of cognitive focus in the P–S division. These two figures show that in the beginning of the design session (i.e., the first two fractioned sections), Mixed teams resemble the ID teams' designing style, deploying their primary focuses on the design problem. With the progress of designing, Mixed teams exhibit a designing style similar to ME teams, as evidenced in the interwoven curves of Mixed and ME teams in the latter 5 fractioned sections. This suggests that ID and ME students may make a stronger contribution in the problem formulation and solution development respectively.

Another finding about Mixed teams is the relative stability of their designing style. The paired-sample Wilcoxon Signed Rank test indicates that Mixed teams' P–S indexes are not significantly different between the two tasks. Visual presentations of sequential P–S division in Figs. 6 and 7 also demonstrate a similar trajectory between Task CM and Task PES.

## Conclusion

This paper presents a novel measurement of designing styles through reinterpreting the FBS design issues [15, 16] through the dichotomy of problem and solution spaces. Compared to a set of design issues, the single-value P–S division can facilitate an efficient comparison between groups, in particular in the cases involving more than two groups. The translation of the design issues into cognitive focuses on problem and solution, another commonly used terminology in design research, also provides a connection between the FBS-based protocol studies with non-FBS-coded design research.

This new P–S division measurement is then applied to a set of results from an experiment, which examines the effect of design discipline and the type of task on the style of designing. It is found that ID teams tend to have problem-focused designing styles and ME teams have a very solution-focused style. A small design team's designing style may shift while designing with different classes of requirements. The same group of designers tends to focus more on design problem when they deal with open-ended design requirements.

These results also imply that, simply grouping people from different disciplines may result in a designing style mixing the characteristics of those disciplines. An efficient multidisciplinary design team may require team building efforts to make it happen.

Due to the explorative nature of this study, these findings are tentative and limited by the small sample size in this experiment. Future studies with a larger sample size are needed to generalize these findings and provide more insights in the relevant areas.

In addition, ontological design processes are a consequence of the FBS-based protocol segmentation and coding, defined as transitional processes between pairs of design issues [15, 16]. Categorizing ontological design processes into problem-focused and solution-focused design processes may provide a new perspective to examine designing styles.

**Acknowledgements** The second author's contribution to this work is funded by the US National Science Foundation under grant numbers CMMI–0926908 and IIS-1002079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Goel V, Pirolli P (1992) The structure of design problem spaces. Cogn Sci 16(3):395–429
2. Dorst K (2010) The nature of design thinking. In: Dorst K, Stewart S, Staudinger I, Paton B, Dong A (eds) Design thinking research symposium (DTRS) 8: interpreting design thinking, pp 131–139. University of Technology Sydney, 19–20 Oct 2010
3. Lawson BR (1979) Cognitive strategies in architectural design. Ergonomics 22(1):59–68
4. Darke J (1979) The primary generator and the design process. Des Stud 1(1):36–44
5. Schön DA (1992) Designing as reflective conversation with the materials of a design situation. Knowl-Based Syst 5(1):3–14
6. Schön DA (1991) The reflective practitioner: how professionals think in action, Paperback edn. Ashgate, Aldershot, UK
7. Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem–solution. Des Stud 22(5):425–437
8. Maher ML, Tang H-H (2003) Co-evolution as a computational and cognitive model of design. Res Eng Design 14(1):47–64
9. Restrepo J, Christiaans H (2004) Problem structuring and information access in design. J Des Res 4(2). doi: 10.1504/JDR.2004.009842
10. Kruger C, Cross N (2006) Solution driven versus problem driven design: strategies and outcomes. Des Stud 27(5):527–548
11. Lloyd P, Scott P (1994) Discovering the design problem. Des Stud 15(2):125–140
12. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36
13. Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Des Stud 25(4):373–391
14. Gero JS, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19(1):21–61
15. Pourmohamadi M, Gero JS (2011) Linkographer: An analysis tool to study design protocols based on FBS coding scheme. In: 18th international conference on engineering design (ICED'11), Copenhagen, Denmark, 15–18 Aug
16. Gero JS (2010) Generalizing design cognition research. In: Dorst K, Stewart SC, Staudinger I, Paton B, Dong A (eds) DTRS 8: interpreting design thinking, pp 187–198
17. Jiang H, Yen CC (2010) Understanding senior design students' product conceptual design activities: a comparison between industrial and engineering design students. In: The 2010 design research society (DRS) international conference "Design & Complexity", Montreal, Canada, 7–9 July, CD un-numbered
18. McDonnell J, Lloyd P (eds) (2009) About: designing: analysing design meetings. CRC Press, Boca Raton
19. Keinonen T (2006) Introduction to concept design. In: Keinonen T, Takala R (eds) Product concept design: a review of the conceptual design of products in industry. Springer, Berlin, pp 1–31
20. Dorst K (2003) The problem of design problems. In: Cross N, Edmonds E (eds) Expertise in design, design thinking research symposium 6, pp 135–147
21. Purcell T, Gero JS, Edwards H, McNeill T (1996) The data in design protocols: the issue of data coding, data analysis in the development of models of the design process. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, Hoboken, NJ, pp 225–252

# Mitigating Design Fixation Effects in Engineering Design Through Product Dissection Activities

## Christine Toh, Scarlett Miller and Gül Kremer

**Abstract** Design fixation plays an important role in design idea generation, and has been found to be complex in its definition and implications. Identifying the factors that influence fixation is crucial in understanding how to improve design pedagogy and mitigate fixation effects. One way to potentially mitigate fixation is through product dissection activities as this activity has been shown to increase creativity and design exploration in engineering design. However, since product dissection has not been studied in terms of design fixation, it is unclear if, or how, this type of activity influences fixation. In addition, although prior work studied product dissection in a team environment, it did not study how individual factors such as personality attributes influence one's involvement, or exposure to the dissection. This is an important factor to study in order to understand how team-based dissection activities influence design fixation because the participation of each team member can be affected by factors such as personality traits. Therefore, this study explores the interaction between product dissection, personality traits, and design fixation in an engineering design class setting. It was found that design fixation was indeed impacted by extraversion and conscientiousness personality traits when adjusting for semester standing and exposure to the dissection activity. These findings implicate personality in the product dissection activity, as well as suggest product dissection as a way to mitigate design fixation. By understanding these interactions, the overall design process can be enhanced, as well as our understanding of design cognition.

---

C. Toh · S. Miller (✉) · G. Kremer
Pennsylvania State University, University Park, PA, USA
e-mail: shm13@psu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_6,

## Introduction

Hertzberger [1] once said that “Everything that is absorbed and registered in your mind adds to the collection of ideas stored in the memory: a sort of library that you can consult whenever a problem arises. So, essentially the more you have seen, experienced, and absorbed, the more points of reference you will have.” This saying finds truth in the field of engineering design, where the field has changed from a design from scratch environment to a design through synthesis environment, where designers transform, combine, or adapt elements of existing designs in order to generate new ideas [2, 3]. However, the use of examples can also negatively impact the design process in the form of design fixation [4], a potentially limiting adherence to existing examples. That is, the information that designers ‘absorb and register’ in their mind have the potential to fixate them during the design process. Furthermore, because design fixation occurs across different levels of expertise [5] and contexts [6], it is important to understand how different activities affect fixation during the design process. The development of methods that reduce fixation effects is important in enhancing the overall design process, as well as contributing to our understanding of design cognition.

One way to potentially mitigate fixation is through product dissection, as dissection has been shown to increase creativity and design exploration in engineering design [7]. However, since product dissection has not been studied in terms of design fixation, it is unclear if, or how, this type of activity influences fixation. In addition, although prior work [8] studied product dissection in a team environment, it did not study how individual factors such as personality attributes influence one’s involvement or exposure to the dissection activity. This is important because not every team member participates equally in design activities, [9] and thus, could have varying levels of fixation based on their exposure level. This involvement could vary due to individual factors such as personality attributes. Therefore, the purpose of this paper is to two-fold. First, we seek to understand how individual factors such as personality attributes affect exposure time in team-based dissection activities. Second, we aim to explore the impact of product dissection activities on design fixation in a team environment.

### *Design Fixation*

Familiarity with the design fixation literature is important in understanding the purpose of this study. Anecdotal and historical accounts have shown that even the most creative ideas are developed through minor extensions of familiar concepts [10]. Therefore, design examples, or known solutions to a design problem, can serve as a catalyst for design activities by stimulating idea generation and orienting the designer to the problem space [11]. Although this mapping of old to new can facilitate progress, it can also limit an individual’s ability to ‘think outside the box’ or move beyond familiar concepts to develop something truly unique.

Jansson and Smith [12] were the first to study fixation effects in design. They hypothesized that designers who were shown pictorial examples prior to idea generation would experience a mental block, reducing access to other ways of solving the problem. Their work validated this theory, when they found that designers in the example condition reused more features from the example set compared to those who were not. This was found to be true for both novice (students) and expert (practitioners) designers even when example features were deemed inappropriate. They defined this lack of flexibility in the design process as design fixation, or a "blind and sometimes counter-productive adherence to a limited set of ideas in the design process". Follow up studies also reported similar findings on the fixation effects of examples during the design process [13–16].

While these studies highlight the presence of fixation in design, other research has shown the complex nature of fixation. For example, Purcell and Gero [6] found that although designers can get stuck on existing examples, design fixation might be dependent on variables such as the designer's domain knowledge. Tseng et al. [17] also explored the complexity of design fixation and found that the timing and analogical similarity of the examples presented impacted fixation effects. Other studies indicate that design fixation is all-pervasive in that it even affects experts in the field that are aware of the limiting effects of design examples. For instance, Linsey et al. [5] showed that engineering design faculty who research fixation effects can become fixated during the design process, without even realizing that fixation is happening. These studies highlight the complexity of fixation and the variety of effects that can impact the type and strength of fixation that occur during the design process.

Although the evidence that design fixation occurs is quite compelling, researchers believe that it may be possible to overcome constraining effects by providing participants with de-biasing instructions [13] or by providing useful analogies [14]. The results from these studies highlight the possibility of mitigating fixation effects caused by examples, but require additional information (instructions and analogical operators) to be provided during the design activity to de-fixate the designers, which is hard to replicate in prototypical design situations. In other words, design tasks rarely come with de-biasing instructions or analogical operators that effectively mitigate the effects of fixation. Furthermore, because fixation happens in an unconscious manner [18], it is not always easy to perform an intervention at the design stage. Nevertheless, these works direct the field to focus on methods of mitigating design fixation effects, starting with understanding the factors that contribute to fixation in *existing* design activities. Therefore, the goal of this study is to understand how product dissection activities, a tool frequently used during the re-design process, affects fixation. Product dissection is particularly alluring for mitigating fixation effects as it can be implemented without specificity to the problem (no de-biasing instructions or analogical operators need to be generated), and the products for use in dissection activities are generally available to the designers.

## *Product Dissection*

Product dissection is often utilized during the design process as a way to systematically uncover opportunities for re-design [7]. Designers take apart or analyze all components and subcomponents of a product [19], adding to the understanding of its structure and properties, and uncovering opportunities for product improvement [20]. Ultimately, the goal of dissection is to improve the maintainability and reliability of a product, implement new technologies, and increase the functionality of the product [21] through the examination, study, capture, and modification of existing products. As such, the role of product dissection in design is important in enhancing the design process and improving the quality of the generated designs.

The benefits of product dissection activities are realized in both industry and academia. At the industry level, companies perform product dissection to provide competitive benchmarks and gain knowledge and insight of a particular product. At the classroom setting, product dissection provides students insight into industry practice [20] and 'hands-on' experience [22]. One study on dissection has shown that students that perform product dissection in a team environment are more creative, develop more ideas, and explore both the form and function of a design compared to those that do not [7]. This deeper exploration of the design space as a result of dissection activities suggests that product dissection could have a constructive effect on design fixation, and has implications for designers beyond the classroom setting. In addition, the literature shows the successful implementation of product dissection activities in engineering design classrooms and highlights the growing importance of hands-on experiences in engineering education [19, 20], and [23]. This is important because it contributes to the overall understanding of the design process as it is implemented in industry, and can help enhance the quality of the generated designs in various settings.

Although these studies highlight the utility of product dissection activities during the design process, they neither investigated how this type of activity affects fixation, nor how individual factors such as personality mediates involvement in dissection activities. In this paper, we respond to this research gap.

## *Team Performance and Personality*

Although product dissection may be a useful tool for mitigating fixation, it is often conducted in a team environment, and therefore, all team members may be affected differently by the dissection activity due to team involvement. This unequal involvement in design activities could be attributed to individual factors such as personality, which could result in varying levels of fixation based on their exposure to the dissection activity. However, the role of personality traits on design fixation or team product dissection activities has not been explored in the literature. Therefore, it is important that we study personality attributes as they relate to the exposure to the dissection activity and design fixation.

The Big Five Factors of Personality (Five Factor Model) framework developed by Costa and McCrea [24] is used extensively in the literature, and is recognized as a reliable measure of personality. This model of personality states that personality has five dimensions: Neuroticism, Extraversion, Openness to Experience, Agreeableness, and Conscientiousness. These attributes have been shown to play a significant role in small team performance [9], a setting that is common in engineering design. For instance, those that score high on agreeableness tend to engage in teamwork, are more cooperative, and have a higher quality of personal interaction, while those who score high in neuroticism often do not cooperate in a team environment [25]. The extraversion personality trait has also been positively linked to successful team performance [26], while conscientiousness has been shown to be negatively correlated with social loafing [27]. Therefore, we hypothesize that personality attributes will effect team dynamics and social loafing, and thus, individual exposure to the product dissection activity.

The purpose of this study is to assess how personality traits affect team performance and exposure time in a product dissection activity. This is important because personality is hypothesized to impact design fixation in team environments. By examining the role of personality in engineering design, the overall design process can be enhanced, adding to our understanding of design cognition.

### *Research Objectives*

The purpose of this study is then two-fold. The first is to examine the relationship between product dissection activities on design fixation. The second objective is to explore the implications of individual personality attributes on the exposure to product dissection activities in team design projects. It is hypothesized that the personality of an individual is correlated to the product dissection process in a team environment, and ultimately, affects the design fixation effects encountered by individual team members. To test these hypotheses, an exploratory study was conducted in a first-year engineering design classroom involving a product dissection activity and a re-design of an electric toothbrush. The results obtained from this study will be used to contribute to the understanding of how team-based dissection activities influence design fixation, and to identify new research paths that extend the knowledge of de-fixating methods, even in a team environment.

## Exploratory Study to Examine Design Fixation

### *Participants*

The participants in this experiment were undergraduate students in a first year engineering design course at a large northeastern university. There were 76 students (61 males, 15 females) that participated in this study from three different sections of the course. Each section consisted of 4-member design teams. Teams were assigned by the instructor based on prior expertise and knowledge of engineering design so as to balance the performance of the teams. This was accomplished through questionnaires that were given at the start of the semester that asked about student proficiencies in the following areas: 2D and 3D modeling, sketching and engineering design experience.

Personality measures for each participant were captured prior to the start of the study using the short Five Factor Model (FFM) online questionnaire (Short Form for the IPIP-NEO (International Personality Item Pool Representation of the NEO PI-R$^{\text{TM}}$) [28]).

## *Procedure*

The design teams were tasked with redesigning an electric toothbrush for increased portability. Two of the three sections (44 students) re-designed the Oral-B Advance Power 400 electric toothbrush while the other sections (32 students) redesigned the Oral-B Cross Action Power electric toothbrush, both seen in Fig. 1.

Each team was given 90 min during one class period to perform a product dissection of the electric toothbrush they were assigned to redesign. During this activity, participants were asked to develop a bill of materials for each subcomponent and identify the team member that led each individual part dissection. In total, 18 participants dissected the brush head, 15 dissected the body, 19 participants dissected both categories, and 3 participants did not participate in the dissection for these two categories. The dissected toothbrushes are shown in Fig. 2.

A week later, the participants attended a brainstorming session, where each team member was given 30 min to generate as many ideas as they could for the redesigned toothbrush without consulting the other participants. The participants were not informed of the brainstorming session prior to its start. During the brainstorming session, participants were asked to sketch as many concepts as possible, writing notes on each sketch such that an outsider would be able to understand the concepts upon isolated inspection. Participants were asked to focus their ideas on two of four categories: brush head, body design, energy mechanism and power supply/accessories (Example in Fig. 3). Each team had to select two team members to develop ideas in each of the four categories. As an example, team member 1 may have developed ideas for the brush head and power supply, team member 2 the brush head and energy mechanism, team member 3 the energy mechanism and body design and team member 4 the body design and power supply. For this paper we will be focusing on only the ideas developed for the brush head and the body design. In total, 18 participants generated ideas for the brush head, 15 for the body design, 19 participants generated ideas for both categories, and 3 participants did not generate ideas for these two categories. On average, participants generated 3 ideas for the toothbrush body and 4.5 ideas for the toothbrush head.

**Fig. 1** Electric toothbrushes used for the design project. *Left* Oral-B cross action power, *right* Oral-B advance power 400

**Fig. 2** Dissected electric toothbrushes

## *Metrics*

To quantify the degree of design fixation for the ideas developed, the metrics developed by Linsey et al. [5] were utilized including: (1) number of ideas, (2) number of same features (number of times features from the example solution appear in generated concepts), and (3) percent fixation (percentage of features from the example solution that appear at least once in participant solutions). In order to develop metrics 2 and 3, two independent raters were recruited to judge each idea based on the method developed by Linsey et al. [5].

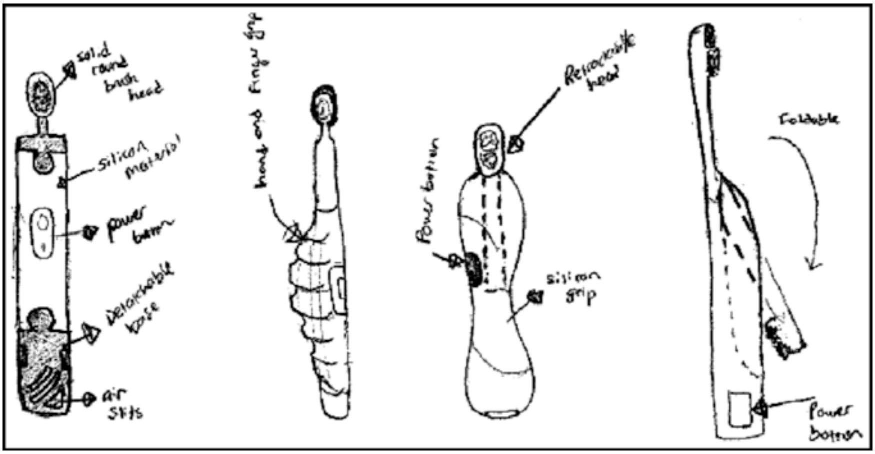


**Fig. 3** Sequential concepts generated for the body design by participant 23

Thirty-one questions were developed to access the similarity of the design ideas developed by the students to the original toothbrush's body design and brush head design, including characteristics such as similarity in shape or size. These questions were developed using the principles of exploratory qualitative analysis [29], and initially were tested against the design concepts to ensure that all the variations present in the designs were addressed. Subcategories within each category (brush head and body) were also generated in order to organize the rating process, as seen in the Appendix.

Two independent raters were asked to rate each generated idea using a scale ranging from 1 = Agree because it is explicitly shown visually AND in writing, 2 = Slightly agree because it is shown either only visually OR only in writing, 3 = Slightly disagree because it is shown either only visually OR only in writing, 4 = Disagree because it is explicitly shown visually AND in writing, and 5 = Not explicitly stated. Ratings with an affirmative response (1 or 2) were rated as similar, and thus, fixated, in the analysis of the data, while negative ratings (3 and 4) were not. The rating scale was developed in order to account for the variation in design presentation, with design ideas presented visually, in writing, or both.

This rating scheme was developed through discussions and training sessions with the raters in order to develop an intuitive and reliable scale. In addition, a design benchmarking handbook was developed to assist the raters in identifying key fixation points, as well as act as a reference during the rating process. The inter-rater reliability was 85.2 % when the responses were grouped using method described above (1 or 2 = fixated, 3 or 4 = not fixated). Disputes were settled in conference between the raters as was done previously by Chrysikou and Weisberg [30]), and a Cohen's Kappa of 0.759 was achieved for the rating method.

In order to examine the effects of the dissection activity on the amount of fixation present in the designs, several metrics were defined:

*# Parts Exposure*: *The number of parts each participant dissected within each category (brush head and body design). In order to examine the exposure of each participant compared to their team members, this metric was ranked for each team member (1–4). A participant with a score of 4 dissected the most parts in their design team.*

*# Ideas*: *The number of ideas each participant generated for each category (brush head and body design).*

*# Same Features*: *The number of features in the generated concept that were deemed similar to the original design by the raters. For this study, an answer of options 1 or 2 by the rater were considered as features similar to the original design, and rating statements that were answered using options 3 and 4 were considered as features that were different than the original design.*

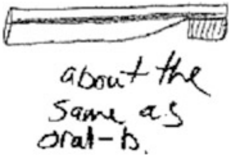
*% Fixation*: *The # of similar features divided by the number of questions rated by the coders for each design (excluding the questions deemed not explicitly stated, or option 5). Examples of designs that were rated and considered non-fixated (low % fixation) compared to the original design are shown in* Table 1.

## Statistical Analysis

In order to address our first hypothesis that exposure to the dissection activity affects design fixation, an MANOVA was performed with the independent variables of # parts exposure for both the brush head and the body design and the dependent variables of % fixation, # of ideas, and # of same features. The exposure variable was taken as combination of exposure to both the brush head dissection and the body design dissection, where there were 18 participants that dissected the brush head alone and 15 participants that dissected the body design alone. There were 19 participants that dissected both the brush head and body of the toothbrush, and were considered separately for the purposes of this analysis. Therefore, the total sample size for the analysis was 76.

In order to address our second hypothesis that personality attributes effect exposure to product dissection activities in team design projects, and thus design fixation, a second analysis was completed. The five personality traits were analyzed for their effects on the product dissection activity by performing a Pearson two-tailed significance test between the personality traits and the # of parts exposure.

**Table 1** Example designs rated as non-fixated with responses for the corresponding rating statements using the rating scale discussed above

| Rating statements | Similar/different from original design | |
|---|---|---|
| <br><br> | | <br> |
| The idea has the same location and number of brush heads | Agree | Disagree |
| The idea has the same shaped brush head | Agree | Disagree |
| The idea has the same bristle length, hardness, and/or direction on the brush head | Agree | Disagree |
| The idea generates the same number of movement types (only rotation, rotation AND vibration, etc.) | Not Explicitly Stated | Agree |
| The idea has the same type and/or range of brush head movement (rotational/translational/vibrational/angle of rotation) | Not Explicitly Stated | Disagree |
| The idea's brush head is similar to the original design | Agree | Disagree |
| The idea performs the same functions (no toothpaste, no tongue scraper, no floss) | Agree | Agree |
| The idea's general characteristics are similar to the original design | Agree | Disagree |
| # of similar features (1 or 2 response) | 6 | 2 |
| # questions not rated as a 5 | 6 | 8 |
| % Fixation | 100 | 25.0 |

Ratings of 1 or 2 were rated as 'agree' whereas 3 or 4 were rated as 'disagree'

Finally, an MANCOVA was performed with the dependent variable being % fixation, # of ideas, and # same features, and the independent variable being the 5 personality traits analyzed independently. The covariates for all 5 ANCOVAs were semester standing and # parts exposed for both the brush head and body designs.

Semester standing was chosen as a covariate in order to achieve statistical control of extraneous or 'nuisance' variables [31–33], and # parts exposed was chosen as a covariate in order to isolate the effects of different exposure time to the dissection activity. Profile plots were generated by categorizing the personality traits into 3 groups (lowest, average, highest), with cut-off scores being half a standard deviation from the mean, as was done previously by Garcia et al. [34].

SPSS v 20.0 was used to perform all of the statistical tests. The level of significance was 0.05.

## Results

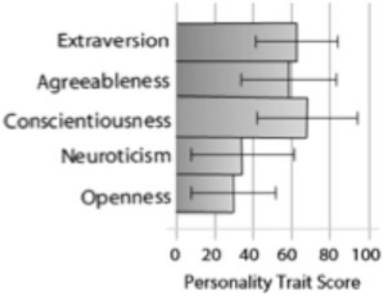
We hypothesized that exposure to dissection activity would affect the fixation effects encountered during the idea generation activity. The test for equality of covariance matrices between # parts exposure and % fixation and # ideas was performed and passed ($p > 0.5$). Therefore, a MANOVA was conducted on these variables. The results revealed that the relationship between # parts exposed for the **brush head designs** and both the % fixation and # ideas was significant ($F = 2.80$, $p < 0.03$; Wilk's $\lambda = 0.854$, partial $\varepsilon^2 = 0.076$), but the relationship between # parts exposed for the **body designs** and % fixation and # ideas was not significant ($F = 2.04$, $p < 0.09$; Wilk's $\lambda = 0.890$, partial $\varepsilon^2 = 0.057$).

To examine these relationships further, follow-up univariate tests were performed on the # parts exposed for the body designs. Prior to testing, the % fixation and # ideas variables were found to have homogeneity of variances ($p > 0.5$). Post hoc comparisons using the Tukey HSD test indicated that the mean # ideas for the group that ranked 2 in # parts exposed for the brush head designs ($M = 4.43$, $SD = 1.612$), was significantly different ($p < 0.03$) from the group that ranked 3 in # parts exposed for brush head designs ($M = 6.50$, $SD = 1.732$). Additionally, the mean # ideas for the group that ranked 1 in the # parts exposed for the brush head designs ($M = 4.55$, $SD = 0.783$) was also significantly different ($p < 0.4$) from the group that ranked 3 ($M = 6.50$, $SD = 1.732$). In other words, those that were exposed to more parts during the dissection activity produced more ideas during the idea generation activity. This relationship indicates that team members that perform the brunt of the dissection activity in their team appeared to have generated more ideas.

The second question we sought to address was if exposure to a product dissection activity in a team environment was impacted by individual personality attributes. The personality distribution of our participants can be seen in Fig. 4.

Our correlation test between the # parts exposed (both ranked and unranked) for each part (brush head and tooth brush body) and the personality traits revealed that for the brush head design, while extraversion ($r = 0.25$, $p < 0.05$) was significantly correlated with # parts exposure (see Table 2). This means that people who score high in extraversion dissected more brush head parts than those that scored low in extraversion. There were no significant correlations for the body design,

**Fig. 4** Personality trait distribution of the participants



indicating that personality did not play a factor in the number of parts the individual dissected for that category.

To further our analysis, a test for the homogeneity of covariance was performed. The results revealed that extraversion ($p > 0.6$), agreeableness ($p > 0.4$), conscientiousness ($p > 0.4$), neuroticism ($p > 0.3$), and openness ($p > 0.3$) did not differ on the covariates of # parts exposure and semester standing. This indicates that assumption of homogeneity of covariances was not violated. Therefore, a second analysis was performed with n MANCOVA and these attributes.

The MANCOVA results indicated a significant relationship between extraversion and both % fixation and # ideas ($F = 1.643$, $p < 0.02$; Wilk's $\lambda = 0.095$, partial $\varepsilon^2 = 0.692$), when we adjusted for semester standing and the number of parts the participant was exposed to during the dissection activity (see Table in Appendix). Further tests also revealed a significant relationship between conscientiousness and both % fixation and # ideas ($F = 1.590$, $p < 0.03$; Wilk's $\lambda = 0.107$, partial $\varepsilon^2 = 0.672$) and openness and both % fixation and # ideas ($F = 1.662$, $p < 0.02$; Wilk's $\lambda = 0.204$, partial $\varepsilon^2 = 0.549$). MANCOVAs using the agreeableness and neuroticism personality did not reveal any significant results. Therefore, post hoc tests were only performed to explore the univariate effect of % fixation and # ideas on the extraversion, conscientiousness, and openness personality traits. These tests revealed that openness significantly affected the # of ideas generated ($F = 2.05$, $p < 0.02$). Marginally relationships were found between extraversion on the # of ideas generated ($F = 1.76$, $p < 0.05$) and between conscientiousness and the % fixation ($F = 1.72$, $p < 0.06$), as seen in Table 3.

In order to explore these relationships in more detail, profile plots were generated for each of the relevant relationships, as seen in Table 4. Based on the MANCOVA and profile plots, the following results were found: Individuals who scored low on extraversion had the highest # ideas, followed by those that scored the highest, and then those that scored the average. Another trend was found for the conscientiousness personality trait. Individuals that scored average on conscientiousness had the lowest % fixation compared to those that scored the lowest or highest. When the openness personality trait was used, it was found that those that scored the lowest on openness had the highest # ideas, those that scored average had the second highest # ideas, and those that scored the highest on openness had the lowest # ideas.

**Table 2** Correlations of # parts exposure and all 5 personality traits

| | N = 76 | Extraversion | Agreeableness | Conscientiousness | Neuroticism | Openness |
|---|---|---|---|---|---|---|
| Brush head # parts exposure | Pearson correlation | **252** | 0.058 | 0.147 | −0.157 | 0.032 |
| | Sig. (2-tailed) | **0.03** | 0.63 | 0.21 | 0.18 | 0.79 |
| Body design # parts exposure | Pearson Correlation | 0.028 | −0.108 | −0.077 | −0.047 | 0.06 |

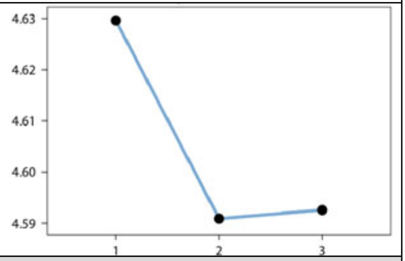
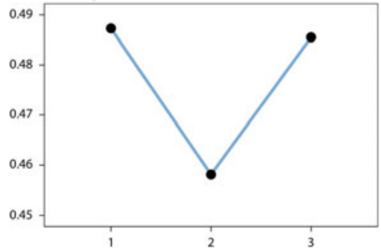
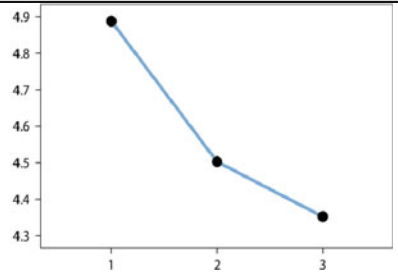
Bolded results indicate statistical significance

**Table 3** MANCOVA results between the dependent variables being % fixation and # ideas and the independent variables being all 5 personality traits

| | N = 76 | Extraversion | Agreeableness | Conscientiousness | Neuroticism | Openness |
|---|---|---|---|---|---|---|
| % fixation and # ideas | F-statistic | **1.643** | 1.29 | **1.60** | 1.22 | **1.66** |
| | Sig. | **0.02** | 0.14 | **0.03** | 0.21 | **0.02** |
| % Fixation | F-statistic | 1.67 | | **1.72** | | 1.11 |
| | Sig. | 0.71 | | **0.06** | | 0.38 |
| # Ideas | F-statistic | **1.76** | | 1.61 | | **2.05** |
| | Sig. | **0.05** | | 0.09 | | **0.02** |

Covariates were taken as the participant’s semester standing and the # parts exposed (brush head and body design)
Bolded results indicate statistical significance

**Table 4** Profile plots of Estimated marginal means of % fixation and # ideas versus Extraversion, Conscientiousness, and Openness

<table>
<tr><th></th><th>% Fixation</th><th># Ideas</th></tr>
<tr><td>Extraversion</td><td></td><td>
</td></tr>
<tr><td>Conscientiousness</td><td>
</td><td></td></tr>
<tr><td>Openness</td><td></td><td>
</td></tr>
</table>

All profile plots had covariates evaluated at semester standing = 1.93, # parts exposed for brush head = 1.45, # parts exposed for body design = 1.37

These results indicate that there is some type of relationship between the personality attributes of individuals within an engineering design team and the amount of fixation experienced and the number of ideas generated. In addition, differences in both semester standing and exposure to the dissection activity resulted in differences in the amount of fixation experienced by each participant.

## Conclusion

The purpose of this paper was to explore the interaction between product dissection, personality traits, and design fixation in engineering design. We hypothesized that fixation effects could potentially be mitigated through product dissection activities as this activity has been shown to increase creativity and design exploration in engineering design. However, since product dissection has not been studied in terms of design fixation, it was unclear if, or how, this type of activity influences fixation. It addition, since product dissection is often performed in a team environment, individuals may have different interactions with the dissected parts based on aspects such as personality. Therefore, a study was conducted in a first year engineering design class to understand how personality attributes affects exposure time and design fixation.

The results from our study indicate that individual personality traits can affect the amount of exposure to the dissection activity. In particular, we found that the more extraverted an individual was, the more involved they were in the dissection activity. This was unsurprising as prior research has shown that extraversion has been positively linked to successful team performance [26]. However, our results not only linked these personality attributes to exposure to the dissection activity, but also highlighted the potential role of certain personality traits in the amount of fixation experienced by the participant as well as the number of ideas generated. Specifically, we found that individuals that scored high on openness tended to generate significantly less ideas. The extraversion and conscientiousness personality trait were also found to play a marginally significant role on the amount of fixation and number of ideas generated. These results are important because they implicate personality in design fixation expression, and hence, a significant factor in the overall design process. In addition, our results showed that individuals who scored high on extraversion were likely to be more exposed to the dissection activity. Therefore or results show that the exposure to the dissection activity is related to personality attributes of team members and also affects design fixation.

While personality traits were found to play a role in the fixation experienced by the participants, other factors such as semester standing and exposure to the dissection activity were also found to affect the personality-fixation relationship. However, our original analysis showed no statistically significant relationship between exposure to the dissection activity and design fixation. On the other hand, it was found that exposure to the dissection activity tended to encourage participants to generate more ideas. These findings agree with previous studies that have found design fixation to be complex, and as a result, can be impacted by many factors (such as personality and exposure to dissection activities) in subtle and multi-faceted ways.

These findings generally support our hypothesis that personality traits and product dissection activities impact design fixation effects. They also highlight the positive effect of product dissection activities in a team environment, but also raise interesting research questions concerning the exact nature of this relationship. Although our study reveals a relationship between personality traits and exposure to the dissection activity, this was only true for the brush head dissection activity, and not for the body design dissection activity. One possible reason for this discrepancy is the difference in level of familiarity and prior exposure to the concepts associated with the part. In other words, it is possible that the participants were more familiar with the concept of improving the ergonomics of the toothbrush handle, but were less familiar with the concept of improving the brushing efficiency of the toothbrush head. Therefore, upon dissecting the brush head part, participants gained more familiarity with the part and were inspired to create a better design. In addition, the use of semester standing as a covariate in the analysis indicates a significant impact of experience within engineering on design fixation.

From this study, the complex nature of individual difference and personality traits is recognized as both a limitation and something to leverage in engineering design research. In other words, because our participants could not be randomly prescribed personality traits, the current work is an exploratory study and lacks the power of a fully experimental design. Similarly, because we were unable to control for exposure to the dissection activity directly, the results of the study could have been influenced by other confounding variables such as drawing participants from different sections of the course. We attempted to adjust for other confounding variables, such as semester standing in this study, but future studies should explore design fixation in a controlled environment, as well as include other confounding variables as covariates in the analysis. Therefore, the effect of this activity on design fixation has to be examined in-depth in future studies to gain a better understanding of its role in the design process.

Overall, our results show that design fixation effects are indeed related to the exposure to a dissection activity and individual personality traits of designers. This has important implications for engineering design research, because it builds on our understanding of cognitive processes as it applies to idea generation, and thus, the overall design process. Future studies should explore the relationship between idea generation techniques of both the form and function of a product on design fixation. The effects of different personality traits on different idea generation techniques should also be examined for its impact on design fixation in order to provide a deeper understanding of how design activities impact design fixation.

## Appendix

See Table 5.

**Table 5** Rating Statements developed for the brush head designs and body designs

| Brush head design | | |
|---|---|---|
| 1 | Brush head | The idea has the same location and number of brush heads |
| 2 | | The idea has the same shaped brush head |
| 3 | | The idea has the same bristle length, hardness, and/or direction on the brush head |
| 4 | | The idea generates the same number of movement types (only rotation, rotation AND vibration, etc.) |
| 5 | | The idea has the same type and/or range of brush head movement (rotational/translational/vibrational/angle of rotation) |
| 6 | | The idea has the same operating speed |
| 7 | | The idea's brush head is similar to the original design |
| 8 | Neck | The idea has a neck that is the same shape and size |
| 9 | | The idea has a neck that has the same rigidity and flexibility |
| 10 | | The idea has a neck that has the same appearance (solid, single piece) |
| 11 | | The idea's neck design is similar to the original design |
| 12 | General characteristics | The idea has the same overall size |
| 13 | | The idea uses the same materials |
| 14 | | The idea performs the same functions (no toothpaste, tongue scraper, flosser) |
| 15 | | The idea connects with the rest of the toothbrush in the same way |
| 16 | | The idea's general characteristics are similar to the original design |
| *Body design* | | |
| 1 | Battery access | The idea uses the same method to remove and access the battery(ies) |
| 2 | | The idea has the same battery access location |
| 3 | | The idea's battery access design is similar to the original design |
| 4 | Power activation | The idea uses the same type of power button |
| 5 | | The idea has the same power button location |
| 6 | | The idea's power activation design is similar to the original design |
| 7 | General characteristics | The idea has the same shape |
| 8 | | The idea uses the same method of providing grip |
| 9 | | The idea uses the same materials |
| 10 | | The idea has the same number of components |
| 11 | | The idea has the same functional features. (no power indicator, no tongue scrubber, no flashlight) |
| 12 | | The idea has the same size and weight |
| 13 | | The idea has the same color |
| 14 | | The idea has the same level of portability |
| 15 | | The idea's general characteristics are similar to the original design |

# References

1. Hertzberger H (1991) Lessons for student in architecture. Uitgeverij, Rotterdam
2. Eckert C, Stacey M, Earl C (2005) Reasoning by references to past design. In: International workshop on studying designers. Proceedings of international workshop on studying designers
3. Eckert C, Stacey M (2000) Sources of inspiration: a language of design. Des Stud 21:523–538
4. Jansson DG, Smith SM (1991) Design fixation. Des Stud 12:3–11
5. Linsey JS, Tseng I, Fu K, Cagan J, Wood KL, Schunn C (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Mech Des 132:41003
6. Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17:363–383
7. Grantham K, Okudan G, Simpson T, Ashour O (2010) A study on situated cognition: product dissection's effect on redesign activities. International Design Engineering Montreal, Quebec
8. Lamancusa J, Jorgensen JE, Fridley JL (1996) Product dissection—a tool for benchmarking in the process of teaching design. In: Frontiers in education conference, Salt Lake City, UT
9. Mann RD (1959) A review of the relationships between personality and performance in small groups. Psychol Bull 56:241–270
10. Ward TB, Smith SM, Finke RA (1999) Creative cognition. In: Sternberg RJ (ed) Handbook of creativity. Cambridge University Press, New York, pp 189–212
11. Dugosh KL, Paulus PB, Roland EJ, Yang H (2000) Cognitive stimulation in brainstorming. J Pers Soc Psychol 79:722–735
12. Jansson D, Smith S (1991) Design fixation. Des Stud 12:3–11
13. Chrysikou EG, Weisberg RW (2005) Following in the wrong footsteps: Fixation effects of pictorial examples in a design problem-solving task. J Exp Psychol 31:1134–11448
14. Linsey JS, Tseng I, Fu K, Cagan J, Wood KL, Schunn C (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Eng Des 132:1–12
15. Perttula M, Sipila P (2007) The idea exposure paradigm in design idea generation. J Eng Des 18:93–102
16. Marsh RL, Ward TB, Landau JD (1999) The inadvertent use of prior knowledge in a generative cognitive task. Mem Cogn 27:94–105
17. Tseng I, Moss J, Cagan J, Kotovsky K (2008) The role of timing and analogical similarity in the stimulation of idea generation in design. Des Stud 29:203–221
18. Linsey JS, Wood KL, Markman AB (2008) Modality and representation in analogy. Artif Intell Eng Des Anal Manuf 22:85–100
19. Wood KL, Jensen D, Bezdek J, Otto KN (2001) Reverse engineering and redesign: courses to incrementally and systematically teach design. J Eng Educ 90:363–374
20. Lamancusa J, Gardner JF (1999) Product dissection in academia: teaching engineering the way we learned it. In: International conference on engineering education. Ostrava, Czech Republic
21. Sage AP (2001) Systems reengineering. Wiley, Hoboken, NJ
22. Brereton MF (1998) The role of hardware in learning engineering fundamentals: an empirical study of engineering design and dissection activity. In: Mechanical engineering 1998. Stanford University, Palo Alto
23. Simpson TW, Thevenot HJ (2007) Using product dissection to integrate product family design research into the classroom and improve students' understanding of platform commonality. Int J Eng Educ 23:120–130
24. Costa PT, McCrea RR (1992) Revised NEO personality inventory (NEO PI-R) and NEO five-factor inventory (NEO-FFI). Psychological Assessment Resources, Odessa, Fla
25. LePine JA, Van Dyne L (2001) Voice and cooperative behavior as contrasting forms of contextual performance: evidence of differential relationships with big five personality characteristics and cognitive ability. J Appl Psychol 86:326–336

26. Driskell JE, Goodwin GF, Salas E, O'Shea PG (2006) What makes a good team player? Personality and team effectiveness. Group Dyn Theor Res Pract 10:249–271
27. Hoon H, Tan ML (2008) Organization citizenship behavior and social loading: the role of personality, motives, and contextual factors. J Psychol 142:89–108
28. Johnson J (n.d.) Short form for the IPIP-NEO (international personality item pool representation of the NEO PI-R$^{TM}$). Cited 2011; available from http://www.personal.psu.edu/j5j/IPIP/ipipneo120.htm
29. Ogot M, Okudan-Kremer G (2006) Engineering design: a practical guide. Trafford Publishing, Bloomington
30. Chrysikou EG, Weisberg WW (2005) Following the wrong footsteps: fixation effects of pictorial examples in a design problem-solving task. J Exp Psychol 31:1134–1148
31. Colvin CR (1993) Childhoold antecedents of young-adult judgability. J Pers 61:611–635
32. Kenny DA (1994) Interpersonal perception: a social relations analysis. Guildford, New York
33. Vogt SD, Randall C (2003) Interpersonal orientation and the accuracy of personality judgments. J Pers 71:265–295
34. Garcia LF, Aluja A, Garcia O, Cuevas L (2005) Is openness to experience and independent personality dimension. J Individ Differ 26:1–7

# Design Fixation: A Cloak of Many Colors

**Robert J. Youmans and Tomasz Arciszewski**

**Abstract** The term *design fixation* is often used interchangeably to refer to situations where designers limit their creative output because of an overreliance on features of preexisting designs, or more generally, an overreliance on a specific body of knowledge directly associated with a problem. In this paper, we argue that interdisciplinary interest in design fixation has led to increasingly broad definitions of the phenomenon which may be undermining empirical research efforts, educational efforts to minimize fixation, and the transdisciplinary distribution of knowledge about fixation effects. To address these issues, the authors recommend that researchers consider categorizing fixation phenomena into one of three classifications: *unconscious adherence* to the influence of prior designs, *conscious blocks* to change, and *intentional resistance* to new ideas. Next, we distinguish between *concept-based* design fixation, fixation to a specific class of known design concepts, and *knowledge-based* design fixation, fixation to a problem-specific knowledge base. With these distinctions in place, we propose a system of *orders* of design fixation, recommend methods for reducing fixation in inventive design, and recommend areas that are in need of further research within the field of design science.

## The Importance of Design Fixation

The concept of *design fixation*, originally defined as a blind adherence to a set of ideas or concepts limiting the output of conceptual design [1], has for 20 years provided researchers from a variety of backgrounds with a compelling, important, and uniquely cross-disciplinary design phenomenon to study. The research is

---

R. J. Youmans (✉) · T. Arciszewski
George Mason University, Fairfax, VA, USA
e-mail: ryouman2@gmu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_7,

compelling because design fixation limits a designer's creative thoughts and actions by anchoring them in the past at the stage of design when creative thinking may have its greatest effect. Design fixation research is also important because innovative products and systems catalyze advances in medicine, art, and science [2], often leading to large financial rewards [3]. Design fixation is thought to affect the mental processes of a designer at the earliest stages of the design process, a period when the architectures of final designs are established, technologies are chosen, and the bulk of the costs (often upwards of 70 %) for a product are committed [4]. In engineering terms, fixation occurs during the conceptual design process, a time during which any given final design outcome is extremely sensitive to the assumptions and chosen strategies of the designer. Fixation during conceptual design can prevent a designer from developing feasible design concepts with consequences ranging from minor duplications of technology to the inability of a corporation to change at the same pace as industry, leading to organizational failure [5].

## The Many Shades of Design Fixation

Interest in what design fixation is, why it occurs, and how it can be avoided has created a bloom of cross-disciplinary research activity, but the "boundary-spanning character" [1] of the phenomena has served as something of a double-edged sword. On the one hand, the interdisciplinary nature of the phenomenon has brought together designers, cognitive scientists, engineers, computational modelers, architects, educators, and many others around the emerging field of *design science*, the scientific study of designing [6]. Design science has revealed important insights into the design fixation phenomena. For example, researchers now speculate that design fixation may occur because of interactions between associative long-term memory systems and working-memory capacity limitations [7]. Researchers also know that some forms of design fixation can be reduced when designers take short breaks [8], use physical materials to prototype [2], incorporate formal design heuristics [9], and potentially, as they adopt computer-based design tools [10].

However, the interdisciplinary nature of design fixation research has also made it increasingly difficult to determine whether or not researchers are all studying the same behavioral phenomenon. Consider one example of design fixation taken from an empirical psychology study where design and engineering students were recruited to compete in a 'Puzzle Box Design Contest' [2]. The contest gave engineering students 90 min to design two original tools that could be operated by hand to retrieve small objects that had fallen into the bottom of a box. Tool designs were restricted to specific rules that prohibited designers from reaching inside the box with their hands, touching the sides of the puzzle box, and so on, and a large cash prize was offered to whoever could create the most original tool design that did not break these rules. Before beginning their own design efforts, participants completed a practice task where they built duplicates of two preexisting tools that had supposedly been created by previous participants. In fact, the preexisting tools were a part of the experiment, and contained ten *fixation features*, easily recognizable design characteristics that could be used to objectively detect fixation effects in later designs. Notably, several of the fixation features of the preexisting tools were negative, that is, they were intentionally designed to *break the rules* of the design competition. Surprisingly, the results of the study revealed that many of the subsequent student designs not only demonstrated high levels of fixation, but they demonstrated fixation to negative fixation features that broke contest rules, disqualifying them from the contest.

Now consider a second case, that of a structural engineer who is designing a beam under bending. Although structural engineers are trained to consider a variety of structural system's construction methods and materials, a common problem for them is their tendency to exploit a single problem-specific body of knowledge to the exclusion of the others they have been trained to employ. This concept, referred to as a *vector of psychological inertia* by engineers, refers to a phenomenon in inventive engineering whereby a designer or a group of designers fixate on a specific class of design concepts, resulting in a tendency to solve engineering problems in the same way over and over again [11, 12]. An engineer who is designing a beam under bending might be said to be following a vector of psychological inertia if she repeatedly designs structures using reinforced concrete beams in spite of the availability of prestressed concrete beams, steel beams, or other potential solutions that do not utilize reinforced concrete beams.

Do both the first and second scenarios represent cases of design fixation? According to many published definitions of the term, the answer is probably 'yes.' In the first empirical study, students blindly adhered to the fixation features of the example designs (even the negative ones), thereby limiting the output of their tool designs. In the second scenario, taken from a real-world example, the structural engineer adhered to one problem-specific body of knowledge (reinforced concrete beams) without consideration of knowledge from other closely related domains of structural engineering, potentially limiting the innovation in his final design solution. In both cases, the designers' past ideas and concepts limited their creative output.

However, there is a critical distinction that should be made between the Puzzle Box Design contestants who fixated in the first example and the structural engineer who always utilizes reinforced concrete beams in the second: the distinction between whether or not the designers were *aware* of their own fixation. In the first example, the designers who fixated were almost certainly unaware that the example tools containing negative fixation features were affecting their work. After all, intentionally copying the negative fixation features disqualified them from a chance to win sizable cash prizes. But in the case of the engineer who chooses to repeatedly design structures using reinforced concrete beams, it becomes much more difficult to determine with certainty whether design fixation is really occurring. If we asked the engineer about the decision to repeatedly use reinforced concrete beams, he or she might react with genuine surprise about their own blind tendency to utilize the same beam materials over and over. Alternatively, he or she might claim to have recognized that their work often incorporated reinforced concrete, but blame the repetition on a genuine inability to think of other materials to use. Finally, he or she might say that that the repetition had nothing to do with some insidious tendency to copy past work, but rather, had to do with the engineer's reliance on his or her own problem-specific body of knowledge on prestressed concrete beams. In this hypothetical, the variety of fixation that the engineer experienced depends largely on their awareness.

Discrepancies between the behaviors that researchers describe using the term 'design fixation' are not limited to the examples we have provided in this paper. An April 2010 symposium entitled *Fixation or Inspiration*? *The Role of Internal and External Sources on Idea Generation* brought together interdisciplinary researchers in Delft, The Netherlands, with overlapping interests in creative problem solving in design and engineering [13]. Attendees of the conference produced seven journal articles on the topic of design fixation that were published in a special edition of *The Journal of Creative Behavior*. Although all seven articles were ostensibly on the topic of design fixation, a quick survey of the examples of design fixation that were put forth by the authors reveals just how *different* many of the examples of fixated designers seemed in comparison with one another. Of the seven articles, two began by referencing examples where fixation was induced seemingly without the designers' awareness by an example design [7, 14]. Two other articles cited examples where designers were aware that they were unable to come up with new ideas because their thinking was blocked by some initial design idea that they were unable to overcome [10]. A third type of example was one where designers actually gained an advantage by intentionally adopting a preexisting design and then transforming it to fit a new design challenge [15]. Finally, one of the two remaining articles theorized that many different types of fixation occur in large corporations or other types of organizations at different stages of the creative process [16], and the other actually highlighted differences between different types of fixation, and repeated a warning that researchers not become 'fixated on our conceptions of what fixation is' [17].

Our point in this review of the conference proceedings is not to champion any one use of the term 'design fixation,' but rather to call attention to just how broadly the term is currently being used. In some ways, the popularity of the term is a good thing; its broad use may be a reflection of the importance of the research as well as the increasing cross-disciplinary research efforts investigating design fixation. However, we argue that the relatively imprecise way in which the term is being used may be doing a disservice to the community by potentially confusing new researchers who are interested in studying design fixation, hurting efforts to educate designers about fixation effects and how they might be reduced or avoided, and complicating efforts to generate a transdisciplinary vocabulary that can be used to describe design fixation behaviors.

To counter recent broadening of the term, we present the following subcategories of design fixation behavior that we recently developed by surveying the current published literature on design fixation and its related behaviors. On the basis of our review, we have identified at least three major forms of design fixation that have been studied, and we recommend that design scientists classify future design fixation research into one of the following categories: (1) studies of *unconscious adherence* to the influence of prior designs, (2) studies of *conscious blocks* to change, and (3) studies of *intentional resistance* to new ideas. We elaborate on the meaning of each category in the following sections of the paper.

## Unconscious Adherence

The idea that a person can be influenced by an encounter with a previous object or system without his or her awareness is not a new idea. The psychoanalysts of the late 19th century assumed that humans were influenced by unconscious internal drives and motivations. In the late 1950s, experimental psychologists who studied attentional processes inferred that unconscious processing of external events in the environment must be taking place in order to explain phenomenon such as the cocktail party effect, the ability for someone to suddenly attend to one's own name when it is spoken across a crowded room by someone in a different circle of conversation [18].

Recently, social-cognitive psychologists studying priming effects have demonstrated to a surprising degree just how often conscious thoughts and actions are influenced by unconscious reactions to the environment. For example, students who share chocolate with their classmates on the same day that their professor is evaluated will irrationally raise their classmates' ratings of their professor [19], and professors who make corrections to students' assignments with red ink will irrationally assign those assignments a lower grade than if they had corrected those same assignments using ink that was blue or black [20]. Presumably, these effects take place without the awareness of either the students or professors, even though they are affecting conscious thoughts and actions. Researchers have even shown that creative problem solving can be improved in insight problems when participants are first primed by seeing an illuminated light bulb, an iconic image representing sudden insight [21].

In design science, there is little reason to assume that these same unconscious influences are also not at play. Admittedly, empirically determining whether some recently encountered environmental example is unconsciously affecting a subsequent design effort is difficult, but one method used by behavioral scientists has been to create studies where example products are shown to designers that contain deliberately *negative* design features. The features may break the rules of the design challenge [2], or may represent designs that failed [1]. The logic of these studies is that no earnest designer should consciously decide to copy such poor features. As a result, any fixation to those features is then thought to be logically attributable to unconscious processes.

In fact, empirical studies by the first author and others have demonstrated that designers will fixate even to negative design features, that is, features that, if copied, would negatively affect the design outcome. One explanation that has been proposed to explain these effects is that humans' associative memory systems store information via associative networks of interconnected concepts in ways that make recently-activated concepts more likely to be retrieved [22]. Design instructors report that students often commit to the design ideas that they think of first [23], and unlike cases of artistic homage or other deliberate references to prior work, designers who experience design fixation may be unaware that they were copying prior examples, leading some researchers to label the effect 'unconscious plagiarism' or 'cryptomnesia' [24–27].

## Conscious Blocking

If design fixation is a blind adherence to a set of ideas or concepts, what happens when a designer becomes *aware* of his or her fixation? While it may seem logical to assume that designers who recognized that they have introduced undesirable fixation into their work would simply eliminate it, psychological studies have long demonstrated that people tend to have difficulty abandoning old mental strategies. Psychologists have demonstrated that creative thinking [28], mathematical reasoning [29], categorization tasks [30], and problem solving [31] all become more difficult when their solutions run counter to previous experience. In these paradigms, people are often frustratingly aware of their inability to avoid fixated thinking, yet their awareness of their own fixated thinking does little to reduce it.

Designers suffer from these same issues; their experiences create familiar solution paths that solve typical design challenges quickly, but that may actually block the generation of new ideas [32]. In a sense, a designer who is consciously fixated is framing the design problem from a problem-specific body of knowledge, and is failing to realize that analogies to past experiences that are outside of his or her problem-specific knowledge base can be sources for design solutions [33]. In engineering and other creative professions, it is obvious that designers gradually expand their experience with practice, i.e. both factual and methodological knowledge regarding their domain increases across time. Methodological knowledge can be understood as methods and decision rules, strong quasi-deterministic rules and very weak rules, often called "heuristics." All these rules represent together what "works" and what "does not work."

With growing practice, the designer accumulates an ever-growing collection of such rules. They allow him or her to easily prescreen many design concepts while considering their feasibility. There is a price, however, which could be called the 'Curse of Experience.' More experience and more decision rules mean that more design concepts are immediately rejected. Therefore, if the goal of a design effort is only to find a satisfactory design concept, experience is helpful, but when a designer is attempting to develop a novel design concept, their experience can actually harm innovation. In this case, experience may lead a designer to discard a large number of design concepts that otherwise would be seriously considered by a less experienced designer because of his or her smaller number of feasibility rules. Such discarded concepts eventually could be evolved and result in novel design concepts.

We could say that a certain amount of experience is helpful in inventive design. There is a common folk belief in engineering that approximately 10 years of experience is necessary to become an inventor (a viewpoint shared by some in the psychology community [34]). However, too much experience may seriously limit the designer's ability to develop novel design concepts. If a designer does not become an inventor around this critical point, each passing year is often thought to decrease the chance that he or she will become an inventor. In terms of fixation, a certain amount of knowledge fixation might therefore be good, but too much of the good think becomes harmful when inventions are concerned.

## Intentional Resistance

Design resistance is the concept that, across a great many different practical domains, there is a prevailing attitude that a previously successful solution is preferable to that of a novel solution. Anecdotally, most people have heard some variant on idioms that warn against 'fixing what isn't broken,' or 'reinventing the wheel.' The point of these sayings is that using past ideas that worked well is preferable to the investments and risks associated with attempting something new. Consider the recently developed Chevrolet Volt, an electric car introduced by the General Motors that contained a novel battery system. After a number of cars were sold, engineers conducting crash tests discovered that the new design was not safe and required costly upgrades. This case of failed new thinking may underscore why designers are sometimes resistant to risk an unproven new technique when a pre-existing design solution is at hand. By adopting an already proven technique, the designer may not have a perfect solution, but they have a workable one. In general, engineers are always concerned about the safety of their products, and it is cheaper and more risk averse for designers to deal with the 'devil they know' rather than to take a risk on some unproven design.

Idioms aside, design resistance may be most rational when viewed in the short term, but it is clearly not optimal when it comes to the long-term development of innovative new designs or ideas. Historically counter-productive examples include the resistance of Americans to adopt the metric system of measurement, the resistance of professional ice-hockey players to adopt safety helmets, and the resistance of sports car manufacturers to adopt automatic transmissions even as their performance became superior to that of manually operated transmissions. Porsche's designers have intentionally kept many of the design features of the Porsche model 911 consistent with the original model introduced in 1963 in spite of the fact that many are not entirely justified in the context of the today's state of the art. Why would someone intentionally choose not to adopt a product or system that is more efficient, safer, or that boosts performance?

One reason might be because a designer genuinely believes that an older system is better. In western education systems, for example, once educators have developed a teaching method that works in the classroom, they may falsely believe that they have developed a method that works *best*. In fact, studies show a strong inverse relationship between teaching experience and innovation of teaching [35]. And although it is rational for someone who mistakenly believes that a design is optimal to resist changing it, design resistance can even occur when designers recognize that a current design is no longer state-of-the art. For example, a designer may recognize that aspects of their design are inferior, but may choose to keep them due to a feeling of envy or competition. A prideful designer may fear that, by abandoning a suboptimal idea they will validate others' claims that the design was, in fact, suboptimal. Further, feelings of nostalgia are common in humans [36], and designers may sometimes prefer time-honored traditional designs regardless of the potential benefits of new systems because of nostalgic feelings.

Is design resistance a true form of design fixation? The answer may hinge on whether it is the design process or the design outcome that is being influenced by outdated beliefs, pride, or nostalgia. Consider what happens when a designer makes the choice to allow design resistance to affect *all* of his or her work, as may be the case when a designer creates an intentional homage to some other artifact. The goal of the designer would not be to improve upon a design, but rather, to mirror as many key elements of it as possible. As such, intentional design efforts to replicate an existing design do not meet the test provided by Jansson and Smith [1], which states that design fixation is a phenomenon that prevents the *consideration* of all of the relevant knowledge and experience which should be brought to bear on any given problem. On that basis, it would seem wrong to suggest that an automobile enthusiast who has succeeded in designing an automobile that referenced other classic cars has fallen victim to design fixation, because the result is not due to a lack of consideration of other ideas, but rather, was intentional.

However, design resistance may, in fact, very much create the types of blind adherences to past ideas or concepts originally described by Jansson and Smith [1], especially if replicating existing design elements is not the goal of the designer. Schon [37] has suggested that "in order to formulate a design problem to be solved, the designer must frame a problematic design situation: set its boundaries, select particular things and relations for attention, and impose on the situation a coherence that guides subsequent moves". Design resistance may therefore affect a final design outcome if mirroring a previous design is not a designer's overall goal, but a past design affects some portion of problem selection, problem framing, designer decision making, or how a designer integrates his or her final design ideas. In this sense, the *intentions* of a designer seem to matter when it comes to determining whether or not design fixation has occurred. We argue that intentional design resistance whereby a designer makes it his or her goal to intentionally replicate elements of a previous design is a class of behaviors that is outside the scope of design fixation research. However, we also stress that tradition or nostalgia may unintentionally bias designers at any stage of their work, leading to cases where designs are fixated without the designers intending them to be.

## Conceptual Versus Knowledge-Based Fixation

Jansson and Smith [1] framed their investigations into design fixation in the context of a theoretical model where the conceptual design process was described as thinking that moves between two mental domains, a configuration space and a concept space. Configuration space contained mental representations of physical design configurations including diagrams, sketches, and combinations of physical elements. Concept space was a mental domain where abstract ideas, relationships, or patterns were considered. Jansson and Smith argued that the conceptual design progress occurred as a designer alternated between thinking in a tangible configuration space and an abstract concept space. Alternating between the two allowed a designer to reveal more about the problem and potential solutions. Barriers to movement between these two ways of thinking would hinder the conceptual design process.

Engineers use a similar framework to that proposed by Jansson and Smith [1] when they talk about the vector of psychological inertia that can lead engineers to suboptimal design solutions. In engineering terms, *conceptual* fixation occurs when a designer, or an entire design group, repeatedly considers only a limited number of concepts. For example, a designer that specializes in the design of underground parking structures might often base all of his or her designs on the single concept of a rigid reinforced concrete frame. As a consequence, if a company wanted to hire the designer to build an underground parking structure at a location where the underground water level was particularly high, the designer might decide to maintain the concept of a rigid reinforced concrete frame by creating a structural system with heavy columns carrying large bending moments and requiring expensive spot foundations, even though a less expensive and less complicated design concept would be a system of shear walls.

Continuing with our example of a designer of parking structures, consider what happens to our structural engineer as he or she accumulates a significant experience (a body of knowledge) related to the analysis and optimization of parking structures based on his or her ever expanding knowledge of reinforced concrete frames. As his knowledge grows, the designer may become less and less inclined to consider alternate knowledge that could inform his or her new designs. This case of *knowledge-based* fixation occurs when a designer, or a team of designers, acquires a substantial body of knowledge in a specific area of engineering and fails to consider knowledge (and the related design concepts) outside of his or her knowledge in this area. Knowledge-based fixation may therefore be thought of as a failure of a designer to consider other tangible physical elements in his or her configuration space.

## Reducing Fixation in Inventive Design

Given the propensity for designers of all types to systematically approach problems, learn by example, and use their knowledge base, all three types of fixation can present serious challenges to creative or inventive thinking. How then can designers hope to best overcome fixation? One approach is to modify *design environments* to decrease the likelihood that designers become fixated on any one concept or knowledge base [2, 38]. Anecdotally, the sense that one's environment is somehow linked to successful inventive design is likely one reason that so many innovative companies invest in creating rich, interactive workspaces designed to foster creative thinking. Engineering educators believe that an academic environment has impact how students learn Inventive Engineering and how creative they become [38–40]. Empirical studies support these notions: working in groups, or working in rich, interactive design environments have been shown to facilitate more original design outcomes [2, 38]. Research also shows that designers who take mental breaks, periods of off-task incubation from a current design effort, may also show less design fixation [8].

Aside from modifying the environment that designers work in, a second potential method for reducing design fixation might be to teach designers to approach design problems in ways that make them less susceptible to fixation effects. For example, in case-based design approaches [41–43], designers are instructed to use their previous experience as building blocks to modify or solve problems in new situations. A structural engineer who is working on the design of a steel roof structure may begin by considering his or her "steel structures" design knowledge acquired through past experience. When a designer is using this knowledge exclusively, then he or she might be said to be using "first order" knowledge, knowledge from within his or her immediate problem-domain experiences and knowledge structure. But an inventive designer might not just consider his or her immediate knowledge when faced with a design challenge, they might also consider knowledge from mechanical engineering, a "second order" knowledge that is closely related to, but separate from, structural engineering knowledge. As the designer continues to think creatively, he or she may consider third order knowledge that is taken from even more distantly related forms of engineering (e.g., chemical engineering), or even "fourth order" knowledge from outside of the engineering profession entirely.

New research on individual differences in cognitive flexibility, the ability to mentally switch between orders of knowledge, suggests that the propensity to switch may also play a role in facilitating creative thinking [44]. What remains to be seen is whether inducing this sort of lateral thinking is possible by utilizing new, or already existing, training techniques, Table 1.

For example, first order fixation, an inability to find solutions within the immediate problem domain, may be susceptible to reduction through Morphological Analysis [45, 46] a method where a problem is broken into subproblems, solutions to subproblems are independently identified, and then randomly

**Table 1** Potential techniques to address different orders of design fixation

| | Morphological analysis | Brainstorming | TRIZ | Synectics |
|---|---|---|---|---|
| 1st order fixation same problem domain | ✔ | | | |
| 2nd order fixation closely related problem domain | | ✔ | | |
| 3rd order fixation distant related problem domain | | | ✔ | |
| 4th order fixation universal knowledge domain | | | | ✔ |

generated combinations of subproblem solutions form potential solutions to the entire problem. Likewise, second or third order fixations, the inability to consider knowledge structures that are not closely related to the problem, may be reducible using Brainstorming methods [47, 48] or TRIZ [11, 12]. Finally, when all available knowledge is being used and fixation still occurs, Synectics [49] provides a knowledge acquisition method called "Excursion" that may be ideal for searching for knowledge within the entire universal knowledge necessary for eliminating fixation.

The authors would like to stress that most real-world cases of design fixation are unlikely to fit neatly into any one of these single categories [5], and we recognize that few empirical studies have tested the effectiveness of morphological analysis, brainstorming, and other training techniques on the reduction of design fixation specifically. However, our point in reviewing these training techniques is to point out that: (1) creative exercises already exist that might be effective with respect to reducing design fixation, and (2) their effectiveness may depend on how well the remedy is tailored to address unconscious adherence, conscious blocking or intentional resistance, see Table 2.

In summary, we are suggesting that designers and design educators should differentiate between different forms of design fixation, and then consider whether different forms of design intervention might not be more or less effective based on the form they are hoping to avoid or eliminate.

More broadly, given the importance of innovation in society, we believe that other interdisciplinary methods for reducing design fixation will be discovered as design science matures, and that it may be helpful for both researchers and design educators to consider couching their research efforts in terms of the types of design fixation under investigation. Specifically, we challenge researchers to consider whether the designers in question are displaying an unconscious adherence to the influence of prior designs, are troubled by conscious blocks to change, or are displaying an intentional resistance to new ideas. We provide Table 2 as a rough guide to design educators who may wish to teach about the various forms of design fixation, and for design scientists who wish to sharpen the focus of their own research efforts.

**Table 2** Types of design fixation with examples and possible remedies

| | Conceptual fixation | Knowledge fixation |
|---|---|---|
| Unconscious adherence | Example: Luchins' 'einstellung' effect (i.e., the use of the same algorithm to solve new problems) [29] | Example: Copying the features (even negative features) of an example [1, 2] |
| | Remedy: Timely warnings to consider all options [29] | Remedy: The inclusion of physical prototyping materials during the conceptual design process [2] |
| Conscious blocking | Example: Perseveration during the Wisconsin card sorting task [30] | Example: Difficulty thinking of new uses for existing object to solve problems [28] |
| | Remedy: Short breaks or 'incubation' [8]. Possibly some design training methods (e.g., TRIZ) [11, 12]. Possibly computer-assisted design [10] | Remedy: Short breaks or 'incubation' [8]. Possibly some design training methods (e.g., TRIZ) [11, 12]. Possibly computer-assisted design [10] |
| Intentional resistance | Example: Thomas Edison's insistence that high power transmission use alternating current | Example: A professional who fails to consider knowledge from outside of his or her own area of specialization |
| | Remedy: No known remedy; possibly systems of cognitive-information feedback [46] | Remedy: No known remedy; possibly interdisciplinary cooperation; possibly creativity exercises; possibly changes in beliefs [50] |

## Conclusions and Future Research

The mental processes responsible for creative behavior have been pondered by some of the greatest minds in behavioral science, including Freud, Skinner, and Newell and Simon. With the relatively recent advent of the field of design science, researchers are gaining ground on some very difficult questions about the nature of human creativity. In this paper, we have argued that design fixation should be thought of as limitations in the inventive design process that occur when designers are biased towards, or are consciously or unconsciously influenced by, a set of conceptual ideas or a previous body of knowledge. This definition may not be the one that researchers ultimately come to rely on, but this updated definition better reflects the various fixation behaviors currently being investigated by the interdisciplinary community of design scientists.

The future of design science research is likely to be influenced by the disciplines of the researchers who study the phenomena, and research questions that are of particular interest to this paper's authors include the potential impact that individual difference in cognitive flexibility may play in designers' ability to resist fixation [44, 51], and how large differences in culture, gender roles, and educational systems may affect fixation rates in an increasingly global society [38]. Extensive behavioral experiments and machine learning studies may both bring important answers to these questions, but these issues also raise the prospect of a new generation of computational design aids that may be able to, for example, spur engineers and inventors to maximize the creative output by comparing their new designs against existing global patents [52, 53].

By categorizing design fixation into six areas, we have highlighted areas that are clearly in need of additional research. For example, the theory that fixation may limit a designers' ability to move between different orders of knowledge, and the possibility that existing creative exercises and methods such as Brainstorming or TRIZ may facilitate movement between them, is certainly worth investigating. Many of these techniques are already taught at universities, although we suspect that many students do not really believe that the methods are very effective. Part of the skepticism surrounding creative exercises may stem from a lack of empirical research documenting their effectiveness, but insufficient knowledge on the part of the students (or faculty) about *when* to use these creative aids and exercises may also erode their effectiveness. Design fixation research may be entering a phase of study where questions about which types of training are most effective under a certain set of circumstances can be more accurately addressed.

An updated definition of design fixation is important to ensure that researchers who study fixation or apply research findings to reduce fixation effects do not conflate one area of fixation behavior with another. There is little evidence, for example, that conscious conceptual blocks and unconscious adherence to negative design features are both caused by the same underlying mechanism, or that the same training methods or interventions would be equally effective in reducing them. However, we worry that others, particularly those without behavioral science backgrounds, might not easily recognize such distinctions, leading to wasted time and efforts. As design science continues to attract researchers and scholars from a variety of technical fields, we believe that developing stronger operational definitions for design fixation phenomena will be important for supporting interdisciplinary cooperation and communication not only between researchers, but also between members of the larger design education community.

## References

1. Jansson DG, Smith SM (1991) Design fixation. Des Stud 12(1):3–11
2. Youmans RJ (2010) The effects of physical prototyping and group work on the reduction of design fixation. Des Stud 32(2):115–138
3. Amabile TM (1996) Creativity in context. West View Press Inc, Bolder
4. Pahl G, Beitz W (1996) Engineering design: a systematic approach. Springer, London
5. Stempfle J (2011) Overcoming organizational fixation: creating and sustaining an innovation culture. J Creative Behav 45(2):116–129
6. Gero JS (2000) Research methods for design science research: computational and cognitive approaches. Doctoral Education in Design, Stafford shire University Press, Stoke-on-Trent, United Kingdom, pp 143–162
7. Youmans RJ (2011) Design fixation in the wild: design environments and their influence on fixation. J Creative Behav 45(2):101–107
8. Smith SM, Linsey J (2011) A three-pronged approach for overcoming design fixation. J Creative Behav 45(2):83–91

9. Yilmaz S, Seifert CM, Gonzalez R (2010) Cognitive heuristics in design: instructional strategies to increase creativity in idea generation. Artif Intell Eng Des Anal Manuf 24(3):335–355
10. Dong A, Sarkar S (2011) Unfixing design fixation: from cause to computer simulation. J Creative Behav 45(2):147–159
11. Altshuller H (1994) The art of inventing (and suddenly the inventor appeared). Translated by Lev Shulyak. Technical Innovation Center, Worcester, MA
12. Arciszewski T (1998) ARIZ 77—An innovative design method. Methods Theor 22(2):796–820
13. Cardoso C, Badke-Schaub P (2011) Fixation or inspiration: creative problem solving in design. J Creative Behav 45(2):77–82
14. Cardoso C, Badke-Schaub P (2011) The influence of different pictorial representations during idea generation. J Creative Behav 45(2):130–146
15. Goldschmidt G (2011) Avoiding design fixation: transformation and abstraction in mapping from source to target. J Creative Behav 45(2):92–100
16. Stempfle J (2011) Overcoming organizational fixation: creating and sustaining an innovation culture. J Creative Behav 45(2):116–129
17. Gero JS (2011) Fixation and commitment while designing and its measurement. J Creative Behav 45(2):108–115
18. Cherry EC (1953) Some experiments on the recognition of speech, with one and with two ears. J Acoust Soc Am 25(5):975–979
19. Youmans RJ, Jee BD (2007) Fudging the numbers: distributing chocolate influences student evaluations of an undergraduate course. Teach Psychol 34(4):245–247
20. Rutchick AM, Slepian ML, Ferris BD (2010) The pen is mightier than the word: object priming of evaluative standards. Eur J Soc Psychol 40(5):704–708
21. Slepian ML, Weisbuch M, Rutchick AM, Newman LS, Ambady N (2010) Shedding light on insight: priming bright ideas. J Exp Soc Psychol 46(4):696–700
22. Collins AM, Loftus EF (1975) A spreading activation theory of semantic processing. Psychol Rev 82(6):407–428
23. Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17(4):363–383
24. Brown AS, Murphy DR (1989) Cryptomnesia: delineating inadvertent plagiarism. J Exp Psychol Learn Mem Cogn 15(3):432–442
25. Marsh RL, Ward TB, Landau JD (1999) The inadvertent use of prior knowledge in a generative cognitive task. Mem Cogn 27(1):94–105
26. Marsh RL, Bower GH (1993) Eliciting cryptomnesia: unconscious plagiarism in a puzzle task. J Exp Psychol Learn Mem Cogn 19(3):673–688
27. Marsh RL, Landau JD (1995) Item availability in cryptomnesia: assessing its role in two paradigms of unconscious plagiarism. J Exp Psychol Learn Mem Cogn 21(6):1568–1582
28. Maier NRF (1931) Reasoning in humans. II. the solution of a problem and its appearance in consciousness. J Comp Psychol 12(2):181–194
29. Luchins A (1942) Mechanization in problem solving: the effect of einstellung. Psychol Monogr 54(6):1–17
30. Grant DA, Berg EA (1948) A behavioral analysis of degree of reinforcement and case of shifting to new responses in a Weigl-type card sorting problem. J Exp Psychol 38:404–411
31. Kershaw TC, Ohlsson S (2004) Multiple causes of difficulty in insight: the case of the nine-dot problem. J Exp Psychol Learn Mem Cogn 30(1):3–13
32. Duncker K (1945) On problem solving. Psychol Monogr 58(5):i–113
33. Dahl DW, Moreau P (2002) The influence and value of analogical thinking during new product ideation. J Mark Res 39(1):47–60
34. Erickson KA, Krampe RT, Clemens T (1993) The role of deliberate practice in the acquisition of expert performance. Psychol Rev 100(3):363–406
35. Ghaith G, Yaghi H (1997) Relationships among experience, teacher efficacy, and attitudes towards the implementation of instructional innovation. Teach Teacher Edu 13(4):451–458

36. Sedikides C, Wildschut T, Arndt J, Routledge C (2008) Nostalgia: past, present, and future. Curr Dir Psychol Sci 17(5):304–307
37. Schon DA (1988) Designing: rules, types and worlds. Des Stud 9(3):181–190
38. Arciszewski T (2009) Successful education: how to educate creative engineers. Successful Education LLC, Fairfax, VA, USA
39. Hou Y (2011) Promoting public spaces of campus for the development of creativity. Chin Electr Power Edu 26:18–19
40. Hou Y (2010) Situating an environment to promote design creativity by expanding structure holes. In: Public places in campus buildings, 3rd Design Creativity Workshop, University of Stuttgart, Germany. Available online http://inspire.usc.edu/dcw10. Accessed 11 July
41. Kolodner J (1993) Case-based reasoning. Morgan Kaufmann, San Francisco
42. Ball LJ, Ormerod TC, Morley NJ (2004) Spontaneous analogising in engineering design: a comparative analysis of experts and novices. Des Stud 25(5):495–508
43. Maher ML, de Silva Garza AG (1997) Case-based reasoning in design. IEEE Expert: Intell Sys Appl 12(2):34–41
44. Figueroa IJ, Youmans RJ (2013) Individual differences in cognitive flexibility predict poetry originality. In: Engineering Psychology and Cognitive Ergonomics. Understanding Human Cognition, pp 290–296. Springer, Berlin
45. Zwicky F (1969) Discovery, invention, research through the morphological analysis. Macmillan, New York
46. Arciszewski T (1988) Stochastic form optimization. J Eng Optim 13(1):17–33
47. Taylor DW, Berry PC, Block CH (1958) Does group participation when using brainstorming facilitate or inhibit creative thinking? Adm Sci Q 3(1):23–47
48. Lumsdaine E, Lumsdaine M (1995) Creative problem solving: thinking skills for a changing world. McGraw-Hill, New York
49. Gordon W (1961) Synectics: the development of creative capacity. Harper & Row, New York
50. Ohlsson S (2011) Deep learning: how the mind overrides experience. Cambridge University Press, New York
51. Youmans RJ, Figueroa IJ, Kramarova O (2011) Reactive task-set switching ability, not working memory capacity, predicts change blindness sensitivity. In: Proceedings of the 55th annual human factors and ergonomics society. Las Vegas, NV pp 914–918
52. Arciszewski T, DeJong K (2001) Evolutionary computation in civil engineering: research frontiers. In: Topping BHV (ed) Civil and structural engineering computing, pp 161–185. Saxe-Coburg Publications, Stirlingshire, UK
53. Kicinger R, Arciszewski T, De Jong KA (2005) Evolutionary computation and structural design: a survey of the state of the art. Comput Struct 83(23–24):1943–1978

# Part III
# Design Creativity

# A Systematic Approach Towards Creative Urban Design

**Kinda Al Sayed**

**Abstract** The last few decades have witnessed a shift from utopianism towards systematic approaches in urban design thinking. The shift has been faced by challenges emerging from the mutual belonging of architecture to both art and science domains. In addition to the widely held claims that a knowledge-based urban design approach would restrain creativity, systematic approaches have been challenged by the complex nature of cities. A full account of the conflicting and overlapping variables in urban design is seen to be unfeasible due to the linear nature of design process. For that, we present a prioritized structure model of design thinking that builds on the generic function of movement in cities. On this ground, we prioritize spatially-determined variables over other quantitative and qualitative variables. We implement the prioritized structure in designing a hypothetical city. From our experiment, we conclude that a knowledge-based design approach can help defining the parameter constrains for solution space. In this process, a creative design input is seen to be inevitable to further define design features and allocate functional relationships. It is seen, however; that by externalizing this process we make explicit the dialectic of design hermeneutics. This approach can be of high value as it enables users and other parties to engage in determining the course of actions required to reach to desirable design criteria.

## Introduction

Design as a practice of human cognition in the reproduction of space and recognition of its partially-dependent components continues to be a fertile ground for speculations and experimentation. The domains that played part in researching this

---

K. A. Sayed (✉)
University College London, London, UK
e-mail: k.sayed@ucl.ac.uk

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_8,

subject range from engineering, architecture, computation to cognitive sciences and psychology. Mainly there is a divide between these domains based on research-centered approaches and practice-centered ones. Research-centered approaches present a top-down view of the design process. The focus is often on quantifying design knowledge and creativity by looking at what designers say rather than what designers do. The reason for that is the hardship in providing comprehensive account of the criteria against which design performance can be assessed. This is also due to the nature of design process itself, seeing that designers unlike researchers tend to be selective about the knowledge they incorporate in their design decisions. Designers can have different preferences about the criteria they reason with. While the criteria might be determined partially by the design problem; background knowledge might also play a non-trivial part in shaping the design course of actions. This is in itself is presenting a problem given that for a design to satisfy different criteria it has to involve different types of knowledge in different capacities. Practice-centered approaches would be more focused on the experiential part of design and often are on the skeptical side about the possibility of modeling design process. The skeptical standing is mainly reasoned by the impossibility to account for all the variables that make designs possible seeing that many of such variables are qualitative rather than quantitative. There is also the argument that design in architecture relies mostly on intuition and is largely an irrational process. To verify this claim, there is a need to reflect in action on the boundaries of rationality in design thinking. To start with that, the nature of design problem needs to be scrutinized and an investigation needs to be held on how defined or ill-defined urban design problems are [1]. From there, the process; whether rational or irrational needs to be elucidated to reflect on the concept of ‘bounded rationality’ in design. As Simon defines this concept, he grounds his theory on the subjectivity of a designer and the cognitive limitations presented by the design situation and the background knowledge [2]. In an approach to tackle this limitation; problem-solution approaches can be classified as to respond to well-defined problems or ill-defined problems. For a well-defined problem, an automated process can be adapted to seek for optimum solutions. For an ill-defined problem, designers could seek solutions that satisfy the solution criteria by means of heuristic methods (trial and error). The implementation of Simon’s methods has been mainly limited to solving engineering problems. Due to the uncertain nature of architectural and urban design, the concept of *bounded rationality* has not been fully explored. Namely, the boundaries for rational and irrational reasoning have not been clearly identified. The identification of these boundaries can only be done with extra caution by isolating the logics of a designer, a situation and the external parameters that influence the design. In dialectical theory, the relationship between the former logics is what makes seemingly irrational decisions reasonable [3]. The nature of a designer’s actions is limited to ‘endogenous logic’ and is framed by ‘exogenous forces’. The design situation acts as an interface between a designer and the world. The ‘exogenous forces’ are described to be partially independent parameters that restrain the ‘free will’ of an actor, or a designer in our case. Mandel identifies the dialectic between the 'endogenous logic' and 'exogenous forces' as a duality that aligns progress in history. This identification echoes the dialectic between a designer and the world in a given design situation as a form of design hermeneutics. These hermeneutics might be internal in a solo design process or partially externalized in collaborative design. Whether, internalized or externalized, design hermeneutics are habitually practiced in the form of question and answer [4]. In a systematic reading of the duality that is inherent in design hermeneutics we might need to force the isolation between what is endogenous and what is exogenous and observe the instants at which an interaction between the two is inevitable. This is not to break the duality between the subject and the object that makes design possible but to understand the notion of 'bounded rationality'. In other words, by unwrapping and disbanding this duality we might better understand the boundaries between objective rationality and subjective rationality. With the term 'subjective rationality' we identify subjective acts of reasoning to be another level of rationality at which decisions are based on qualitative criteria. To explore the boundaries of rationality in architectural design, a structure of priorities needs to be extracted from urban dynamics to inform design reasoning. In such a structure, preference is given to exogenous forces to frame and better define the design problem and to ensure a functional design outcome. A designer's endogenous logic interferes at certain stages to direct the course of design and to further shape design solutions.

Exogenous forces can be a makeup of both the physical space and the social set up of society that inhabits space. In a biased reading of the built form and its embedded social structure, we might see the first to be the materialization of the second. To simplify the problem we might consider the spatial structure of the built form and its associated economic activity as the material manifestation of human activities. With this theoretical perspective, we assume that human activities in space conform to certain patterns for which space is their physical imprint. The interactions between the realms of space, society and economy comprise what Jacobs' terms as the 'organized complexity of cities' [5]. Attributing organization to urban complexity might be suggestive of a certain systematic process that governs what appears to be stochastic. While an explanatory model of this systematic process can provide an understanding for the operative mechanisms of urban complexity all at once, an urban design model would need to reconstruct this process. Both an explanatory model and an urban design model should be based on observed scientific evidence for them not to be alien to the urban phenomena. An urban design model however should be able to effectively project the explanatory model on the course of design process to form a comprehensive approach towards problem-solution definition. The projection of urban complexity on a linear design model can only be enabled by prioritizing certain preferences over others.

In this paper, we present a knowledge-based model that aids urban design decisions. We reflect on this model in action to observe where the boundaries exist between evidence-based rationality and designers' internal logic. The model reflects on a prioritized structure of design thinking. Priority is given to spatial structures to account for movement as the fundamental function of urban form. A generative model is devised to produce different spatial structures. The process of generating street segments is guided by the centrality and extension rules [6]. The structures are then evaluated against properties that are seen to identify the geometric development of cities [7, 8]. After the evaluation, one spatial structure is selected. The configurational properties of the selected spatial structure act as control variables for the parameter space of other partially-dependent form-function variables. This relationship is determined by a parametric model that is extracted from observed relationships between space and form-function parameters in Barcelona and Manhattan. Form parameters include building height and density as well as street width. Functional parameters define the relationship between spatial structure and the overall zoning of the associated areas. Building on a hypothetical sequence of form-function dependencies we can outline several possibilities in which a composite design model operates. We will test the application of both the generative and the parametric models in design and outline events where the model can be fully automated and events where designer input is needed. In doing so, we investigate the role of the composite model in setting the ground for a knowledge-based urban design approach. We also reveal the limited capacity of a purely automated model that is based on spatial criteria in defining the features of design solutions. This limitation does not go against the hypothesis, as we assume that such models are devised to guide designers rather than determine their course of actions.

## From an Analytical to a Synthetic Space Syntax

The presence of explanatory theory of cities architecture that reads movement potentials as a function of space is central to any sensible urban design approach. With this contribution, the role of Space Syntax theory in understanding the science of cities cannot be underestimated [9, 10]. Developments on the theory have further supported the theoretical propositions of Space Syntax about a fundamental relationship between space and society. Efforts in this field have often been devoted to further test the theory on different contexts stressing the existence of cross-cultural invariants that govern socio-spatial behavior. Research efforts have also been directed towards modeling invariant correspondences between the spatial structure of cities and their formal and functional attributes [11]. For these correspondences to be devised in design a comprehensive modeling approach needs to be undertaken. Such an approach would necessarily require projecting observed parametric relationships between Space, form and function on design as a sequential course of actions. Aligning this implementation to design would lead to a major change in the functionality of Space Syntax [12]. It would transform it from an analytical model that decodes urban form and contribute to the knowledge about its functioning mechanisms to a synthetic model that encodes this knowledge into design [13]. A synthetic reading of Space Syntax would fulfill early promises by Alexander [14] on a synthetic reading of form. The outcome of the design is an artificial product that, given its reliability on a functioning urban structure; is expected to be less costly in the process of adaptation to natural growth. This is seeing the cities—in spite of human interventions—self organize their spatial structure to enforce global accessibility into planned areas [7]. Along that process, the planned areas subdivide and deform to imitate natural growth. By embedding natural rules in our reproduction of street structures, we aim at minimizing the effort with which cities counter the disruptions made by human interventions to the Parts-whole structural unity. Along the lines of devising scientific models into urban design there is the designer's worry that by erecting designs on an existent reading of space novelty of designs might be limited. There is the argument that such an approach might lead to a pure reproduction of conventional city spaces. On the side of a scientist, there is another worry. The instrumentalization of knowledge into design decisions that are normative in essence might impose risks on the rigor that has been originally ensured by building on a scientific theory. An argument for the first type of concerns could be that despite the emphasis made on knowledge-based reasoning in systematic design, there is enough space for creative connections to be made. In the meanwhile, concerns raised on the side of science claiming that testing knowledge in the ambiguous logic of design might threaten its profound credibility should not stand against the stream that empowers theory by application. In order to elaborate on how and in which capacity creativity might feed into design without risking the rigor that associates a systematic approach, we need to reflect on that by means of design experimentation.

The decoding of urban dynamics serves as to expose dependencies between variables. Variables that appear to have more control over others can be prioritized in the process of modeling. In doing so, the complexity of urban form and function interdependencies could be partially projected on a hierarchical structure of dependencies and priorities. Such a structure, while reflecting on the dynamic process that directs the functioning of urban form, can better inform the linearity of design process. The nature of design as a process can be read in this course as the set of actions required to reconstruct urban dynamics. The incongruity in setting up a prioritized structure comes from describing a correlation as causal [2]. For that there is a need to distinguish between causal correlations and what is statistically recognized as 'spurious' correlation. Whether a causal correlation means that a variable is fully determined by another variable is also to be questioned. In the theory of 'cities as movement economies' the configurational settings of space are seen to raise movement potentials for certain street segments [11]. This chain is followed by retail activity emerging along routes that have high movement activity. This if described as a causal chain of relationships might lead to the erroneous understanding that space can fully determine movement and its consequent economic activities. Another reading for that is the demand and supply model [15]. In this model, space defines movement potentials. Movement flows while restrained by spatial configurations will then form the demand for certain form-function requirements. The consideration of space-form and space-function relationships in a demand and supply model might structure a directional relationship in which space encourages movement, and movement forms a demand for certain form-function variables to fulfill. A direct relationship between one variable and another should always be taken with caution as to bear in mind the interdependencies between variables, especially those that form the supply for movement. In adapting such models for design support systems, there is a need to rank certain preferences for supply variables. The hierarchical structure that represents space-form-function relationships might be better elucidated as to see space as a control parameter that partially determine natural movement and partially constrain configurational movement.

The preference of a spatial knowledge in constraining design probabilities is not a particularly novel proposition. In fact, Hillier [10] recognized that this knowledge could be built into a design model that prioritizes the 'generic function' of movement and occupation in spatial structures. He defines three design filters that help constraining a design process. The 'generic function' is considered as the first design filter that defines the spatial genotype and characterizes spatial permeability connecting all spaces in a system. By this functionality, it filters design probabilities to define possible solutions. The space of design possibilities can be further filtered by two filters. These filters are to do with the phenotype criteria of design solutions that are determined by individual or communal cultural identity. They constrain design by means of qualitative criteria that is defined by makers or users. The first design filter can be interpreted in a set of 'discursive techniques'. In urban spaces, discursive techniques might be read as the tendency to minimize depth hence conserve on movement from all origins to all destinations. To reflect on urban dynamics we need to examine how patterns of transformations in cities are produced by situated spatio-temporal conditions of the network elements and parts [7, 8]. The key design dimension for this process is to see how the network configurations would shape the urban environment and set the ground for certain economic activities to occupy space. Following this logic, the correlations reported by Space Syntax between space, form and function become instrumental in informing design decisions. An evidence-based design approach that utilizes Space Syntax into design decision making has long been established [16]. The approach has predominantly been applying Space Syntax as an analytical and evaluative tool. For further engagement of Space Syntax in the making of design solutions, the model needs to be adapted to serve in synthesizing designs. The correlations if hypothetically read from one side, can be seen as relationships between control variables and partially dependent parameters. We identify this adaptation of the model as central to the new synthetic functionality of Space Syntax. On this basis, we update the model initiated by Hillier with a prioritized structure of design filters. The ideal implementation of that would be to consider an urban network as a parametric estimator of urban volume and function. We base these parametric relationships on rules extracted from urban regions [17]. With the application of evident rules in design we reconstruct the relationship between space, movement and their economic byproducts. Hence, we entwine the layers of urban complexity by a spatial preference.

To reflect on a prioritized structure of design thinking, we update Hillier's theoretical model of design filters to include four sets. The first set of design filters defines the generative laws of urban space. These laws have been addressed in [6–8]. The laws are extracted from the evolution of urban form. The second set of filters, are inferred from the first set and define parameters that are directly estimated from the temporal state of spatial structure. The third set of design filters are not directly related to space but are determined by other types of quantitative criteria. Examples for that are environmental constrains, construction constrains and emergency planning. The fourth set of design filters are then purely determined by designers or users and encompass all the qualitative criteria that defines the features of a design solution. Qualitative criteria are normally associated with cultural, aesthetic singularities or idiosyncrasies.

Designer and user's involvement in this design filtering process is inevitable for a solution to be defined. Designers are central in making decisions to direct the course of actions and select relevant criteria. They are also involved in making decisions where higher degrees of freedom leave larger space for uncertainty or where overlap and conflict between different variables interrupts design progress. These issues are expected to arise at any stage. In this paper, we aim at presenting a design experiment in the form of a reflective practice [18] where we report the stages at which designer's input is required. We also report difficulties with regards to the automation of certain criteria. We mainly present a design process where we involve the first two filters in determining the universe of design probabilities. We then present creative variations on the outcome of the constrained process. The objective of this tactic is not to claim that a design process can be fully automated but to present a structured approach towards design development and discuss the difficulties that might be encountered in this approach. This is taking into account that by externalizing design thinking we can allow for self-criticism and user participation with the scope to democratize design process. The approach is structured in such a way as to maximize certainty about design decisions at a stage that is considered to be fundamental for a spatial structure and its associated form-function criteria to operate. With the gradual shift towards uncertainty, there is an increasing need for a designer/user creative input to further shape design outcomes. This filtering process is structured to conserve on problem-solving where automation can be an option. It allows for defining decision points where selection and allocation is needed. While the form-function parameters described in [17] appear to be strongly determined by spatial configurations, we expect overlapping and interrelationships between the parameters themselves. More importantly, we expect that these parameters involve internal evolutionary dynamics that link them to their prior states. The sequence in which the form-function parameters operate to shape urban form can be inferred from the process of urbanization itself and can be tested through simulations.

## The Dynamic Geometry of Cities

Simulation of growth behavior in cities has been a domain where urban geographers have invested for decades [19, 20]. Models that have been developed to simulate growth were based on combined CA-agent techniques. Structural properties of the street network have not been represented in these simulations. Taking an analytical approach, early Space Syntax experiments [9] have presented a generative pattern of organization on the local scale of an urban area. The approach was further pursued by investigating the emergent structural properties that result from the repetitive process of block alignments. This has led to evidence-based assumptions about the characteristics of local and potentially generative dynamics in organic grid. Hillier [10] recognizes the tendency of longer lines to continue straight and shorter lines to be blocked forming near right angles with other lines. By identifying that process as the "centrality and extension" rule he sets the ground for the assumption that local rules will have an effect on the global pattern of urban structures. Whether, a centrality and extension rule on its own can lead to the generation of cities is something that needs to be questioned, provided an evidence on different orders of urban growth and its stationary effects [8]. In search for generative dynamics that governs cities, a process of preferential attachment has been outlined [7] to operate on the local and global structures given their spatio-temporal conditions. The process seems to also involve the pruning of weakly integrated local structures. The system's integration values are apt to be normally distributed if an organic grid pattern prevails. Stationary patterns that were seen to be conserved by the system throughout growth might be considered as a side effect of this process [8]. The conservative patterns were recognized as the steady state that the system arrives at in a process of reaction-diffusion. In this process, the phenomenon of equally distributed metric patches that resembles dissipative solitons is marked. The patterns emerge as the system is in a continuously updated state of equilibrium where wave like structural change spread from the original core towards the edges and bounce back into the system. Both generative processes and their steady state effects constitute the criteria against which we identify a spatial structure to belong to the class of urban street networks. At this stage it is difficult to rule out the role and sequence in which these laws generate urban structures, we therefore take them as criteria for urban pattern recognition.

## Form-Function Parameters

In our suggested model for a prioritized structure of design thinking, we have referred to the second set of filters as those that are determined by the geometric configurations of the street network. Analytical approaches that looked for correlations between spatial structure and urban form and function have occupied the validation front of Space Syntax theory. An attempt to go beyond the validation to modeling has been made by Banister et al. [15]. As explained earlier, the demand and supply model they propose interprets the state of the urban street structure as the demand and interprets the supply to be the corresponding street width and landuses. The demand can be determined by the functionality of the street network topology as a regulator for movement flows. The supply can be read as in how the parameters of street width and landuses would respond to estimated movement rates that are provoked by the network properties. Research has followed this approach to outline a comprehensive model [17] that brings together all the associations between form-function properties and space. This model updates the deriving point where space is considered to be the demand by accounting for its topo-geometric properties. The associations between the demand and the supply are further translated into parametric constrains. Following that an evidence-based parametric model is outlined. In this model, space acts as a proxy indicator for urban volume and its overall zoning functionality. To extract the parameters, an intermediate layer, the *pixelmapper* is devised to translate the values of the spatial structure and the data points of the form-function maps into a certain resolution. The resolution is defined by the length of the polygons' edges in this layer. In order to test the parametric model, we take it to be the second set of design filters. The solution space defined by such filter is to be further refined by a designer's selection of third and fourth order of filters.

## Generative Variations and the Geometric Filter

In this section we evolve a number of growth iterations for hypothetical urban patterns using Hillier's centrality and extension rules with a margin of randomness. For that longer lines are encouraged to continue in the system and intersect with other lines forming semi-continuous patterns. Shorter lines are more likely to stop at the first line they intersect with forming near-right angles where possible. The structures produced present varying syntactic properties and patterns. In order to recognize structural patterns that match those of cities we assess these iterations by looking for properties that identify the generative processes outlined earlier. We particularly resort to the property of normally distributed integration values and steady state metric patches as criteria for urban pattern recognition. The question is whether differentiation and self-organization that mark these generative and steady-state patterns can be a natural product of a local generative rule. The experiment's first objective is to verify by means of simulation whether this process of centrality and extension can on its own generate urban structures that match the configurations of cities geometry. The second objective is to look at this process from a design perspective and identify stages at which a designer's input is needed.

By running Choice SLW (segment length weighted) analysis, we can extract a structure that has the highest 10 % values and evaluate its continuity. The

| | | | |
|---|---|---|---|
| ∑TD/NC | 15726.2 | 11366.29 | 7747.37 |
| ∑ SL | 51730.92 | 46011.49 | 39664.84 |

**Fig. 1** The first row represents growth iterations using the centrality and extension rule. The second row displays segment choice SLW analysis

**Fig. 2** Rendering metric mean depth values within different radii for the three growth iterations

evaluation can be made by measuring its normalized cumulative total depth values and cumulative segment length. Choice as a measure of ‘shortest putative journey’ and integration as a measure of depth in the system are angular-based graph properties of the street network [21]. Steady state patterns can be recognized by running metric mean depth analysis (MMD). The measure simply represents average physical distance from each street segment to all neighboring segments in a network [22]. Integration values are expected to be normally distributed in a grid that presents a differentiated structure. The normal distribution can be evaluated through measuring the goodness of fit KSL test. The evaluation measures of choice indicate that iteration 3 performs better than iterations 1 and 2, Fig. 1.

Calculating MMD for different radii does not identify clear patchwork patterns in the background network of any of the three variations, Fig. 2. We might need to run this measure on a larger scale to verify this, but for the scope of this analysis we could report that a local rule on its own is incapable of producing steady state

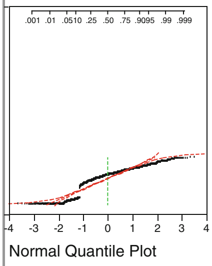
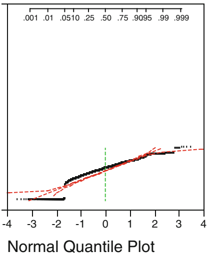
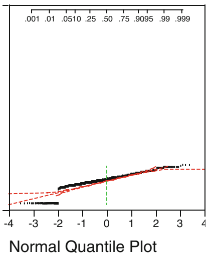
| | **Iteration1** | **Iteration2** | **Iteration3** |
|---|---|---|---|
| Intelligibility $R^2$ | 0.126 | 0.139 | 0.179 |
| Distribution of Integration Rn values |  |  |  |
| — Normal | (653.1,265.102) | (750.239,242.664) | (586.137,117.541) |
| Goodness-of- Fit KSL Test | | | |
| Prob>D | D | D | D |
| 0.0100 > | 0.149932 | 0.095711 | 0.03652 |

**Fig. 3** Testing the distribution of integration values for different iterations of growth

patchwork patterns. The patterns we target are identified as equally distributed patches that are byproducts of a self-replication process, a property of reaction-diffusion systems [8]. We believe that the absence of clearly defined patchworks is due to the fact that the directional mechanism of the current model not accounting for reinforcing feedback. Judging on KSL test we find that Iteration 3 fits best with normal distribution, Fig. 3. Given the findings we have, iteration 3 prevails as it presents an optimum path in the foreground structure that conserves physical distance and angular turn costs. It also presents a structural differentiation that match better urban form. We therefore choose it and proceed by applying the parametric model to define design features.

## A parametric Framework for Form-Function Definition

In this part of the experiment, we implement the parameters extracted from space form and function relationships to deduce future states that maximize these correspondences. It must be emphasized that the correspondences between street structure and form-function parameters is highly effective on the street level itself and loses its significance when it comes to higher and lower street levels. An exception for that is the building height parameter, which for construction convenience might not exhibit huge variations. To extract the parameters from existing urban cases, we mainly used an intermediate layer that we called the *pixelmapper* to translate the spatial information into a certain resolution. The future states are then defined within that resolution level and might be interpreted as the target space for a maximized association between spatial configurations and form-function attributes. The realization of this association will raise the effectiveness in which the physical domain of urban space responds towards form-function. We outline target spaces separately for each form-function variable and discuss the conflicts and overlaps in-between them. We then present some variations on the target space determined by these parameters.

Parametric constrains explained in [17] can be summarized as follows;

- Segments marked by highest 10 % values of choice SLW are more likely to be wider than others.
- Higher aggregate values of connectivity within a *pixelmapper* unit mark a proxy indicator of higher block density.
- Higher buildings are more likely to be on an intersection point between high choice SLW elements, the likelihood is enhanced by a proximity to dense patches within walkable distance (R1000meters).
- High integration values Rn is a proxy indicator for commercial zones.
- Islands defined by highest 10 % values of choice SLW that have high metric mean depth R3000 are more likely to be zoned as industrial.

Plotting the target space for these parameters from the measures defined by the spatial structure we can highlight the target space for each parameter on its own (see Fig. 4a). The parameters can be fully automated given the spatial measures, with the exception of industrial zones and high-rise buildings. These two parameters are based on the metrically-defined patchwork patterns that seem to render outliers in the analysis. These outliers need to be excluded from the calculation for the patches to be recognized by an automated process. It might be perceivable, looking at the target spaces that the parameters overlap leaving more space for hypothetical assumptions about the interrelationships between these parameters. At this stage, a designer or user input is needed to make decisions and define the relationship between these parameters. The relationship can be defined by furthering the investigation on the association between the variables and real urban scenarios to extract second order parametric rules. Taking that into account, we list several assumptions regarding dependencies between form-function variables as follows;

1. Street width is predominantly determined by high choice SLW values, however continuities are more likely to preserve consistent street width.
2. Block density is predominantly determined by local structures but is more likely to be concentrated around commercially active centres.

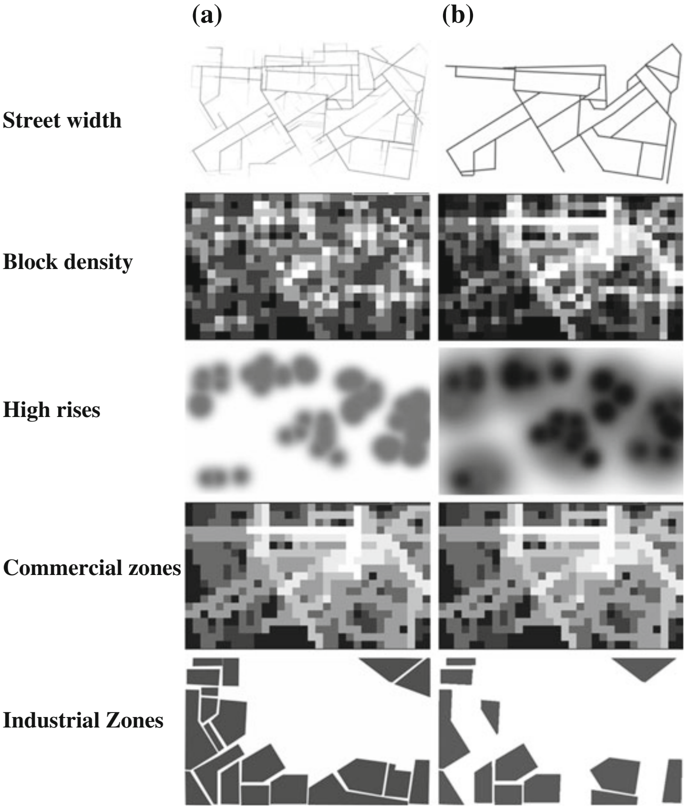


**Fig. 4** Measures of spatial structure as proxies for form-function parameters. Spatial structures analyzed using UCL Depthmap [23]: **a** target spaces for form-function parameters based on spatial configurations; **b** target spaces considering interrelationships between form-function variables

3. High-rise buildings are determined by the configurational structure. Buildings are more likely to be higher in commercially active zones. Buildings are assumed to be lower close to industrial zones.
4. Industrial zones have second priority when overlapping with commercially active zones. An overlap with dense areas reduces that effect.

Taking these assumptions into consideration, and given that the parameters were initially extracted from the spatial measures, further refinement might be needed for the parametric model itself. This means that the likelihood for each parameter might need to include positive or negative multiplier effects given an overlap or a conflict with other variables. Parametric constraints on this level of detail cannot be read within clear thresholds. Fuzzy boundaries and Gaussian decay are expected to mark the landscape of the solution space. A preliminary attempt is made in these regards to refine the target space by taking the previously mentioned assumptions into account (Fig. 4b). Given the first assumption, a continuous foreground structure is traced to represent major road network that links the overall structure and thus demand wider street segments. By applying the second assumption, considerable alteration is made to the target space. In this case, a verification of the assumption is needed through an evidence-based approach before considering this assumption as second order parametric rule. With the third assumption, we can further distinguish differentiations on the building height parameter that would help us approximate a target space for that variable. The fourth assumption is found to aid on decision-making regarding the percentage of zones given a preference for commercial activity. Functional constraints are intended to describe an overall property of a zone mainly on the ground level rather than a precise functional type of the identified building blocks. The level at which this functionality is concentrated in certain zones depends on the type of functionality itself and the overall zoning requirements for a city. Commercial and industrial areas normally concentrate in lower levels. Residential and office spaces are more likely to occupy higher storey levels. Regardless of that, the zoning of areas does not determine the programme in which the local functional organisation would operate. It does only imply that there are higher percentages of certain functionalities within an area compared to others.

## Ensuing an Exploration into the Universe of Design Possibilities

While we ignore at this stage the third set of design filters given that such a procedure would require multidisciplinary expertise, we aim at presenting design variations after having gone through a knowledge-based design process. The process we identified thus far can be automated given that all the parameters are quantified, validated and generalized to reflect on real case evidence. There are limitations however, in the precise definition of the boundaries at which correlations between space and form-function parameters converge. Before defining convergences there is a need to generalize the correspondences that outline the parametric model itself. This goes beyond our scope for this paper since we consider such model as an aiding tool to frame objective knowledge rather than a model to shape design solutions. To proceed with this experiment, we model design output variations building on the estimated targets (see Fig. 4). We recognize that unless we translate these constraints into a rigid orthogonal design outcome (see Fig. 5a) there is hardly any recognized definite urban form that can be erected on these principles. The exploitation of all degrees of freedom regarding the directions that buildings might align to or the z dimension of the street level in relation to buildings can lead to interesting variations on the design outcome (see Fig. 5b). The model on its own cannot assign specific features to design solutions without designers input. The role of designers in this process is to identify the

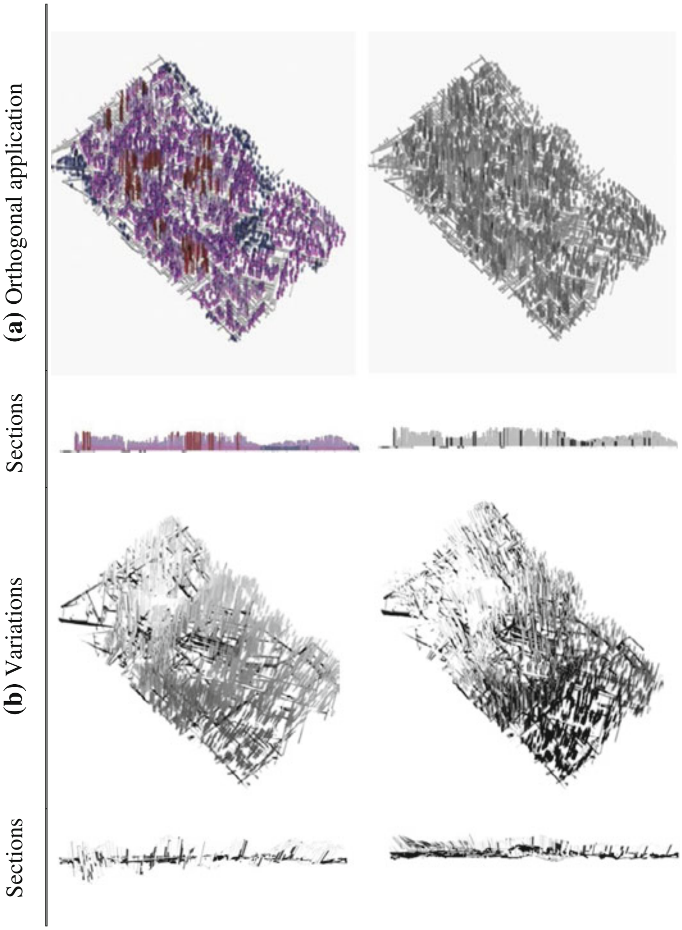


**Fig. 5** **a** A direct orthogonal application of the volume-function parameters determined directly by the configurations of the spatial structure; **b** 3D variations on the target estimated volumetric outcome constrained by the first two sets of design filters and further defined by a designer

characteristics of the elementary proportions and shapes of the blocks along with the dimensionality of the street structure to setup the base model for design. Even if we take the second set of design filters to be fixed, high-rise buildings could grow in all directions and cross over presenting multi-dimensional complexities. Taking that to the extreme, blocks could grow horizontally if the infrastructure affords for such an inclination. Street network could also form wave patterns linking to higher and lower levels depending on predefined criteria. Similarly, blocks could exhibit different densities on different levels. Functions could mix accordingly or could be programmed themselves to produce formal variations. Form-function variations could be a product of the third set of design filters or could be arbitrarily defined by designers. The variations presented yield with the idea that, even with the implementation of a partially constrained process, creative design input is not restrained from defining and tuning the features of design outcomes.

## Conclusions

This paper presents a theoretical model for a prioritized structure of design thinking and associates the theory with an experimental approach that elucidates the role of a designer in an evidence-based approach. Taking that into account, we review the automated procedures and identify where design input was needed to narrow solution space. With this we outline the boundaries between exogenous forces and endogenous logic in relation to the design problem. We make a distinction between an automated process where objective rationality is fully enrolled and a process where designers are needed to direct the course of actions either by selecting relevant quantitative criteria or by engaging qualitative judgment into design. The objective rationality is yet subject to the constructs of measurement and representation. We emphasize the fact that the attributes discussed in the previous sections will filter designs given a certain resolution of a *pixelmapper* grid unit. The generative process, in which the first set of design filters are applied, is directed to produce a functional network structure that affords for permeable movement all through the system. This process can be fully automated, however further dynamics should be attributed to the algorithm for it to present negative feedback effects hence for the outcome to be recognized as an urban pattern. In the second stage, where form-function parameters define the second filtering process, a designer's input might be needed to determine the influence range for each of the parameters. Yet, the target models can be devised to direct design decisions and to maximize the correspondence between form-function and the network structure of streets. For the third set of design filters, multidisciplinary knowledge needs to be incorporated to reflect on other non-spatially determined quantitative criteria. The final filtering process would be fully overtaken by designers or users who may determine the outlining qualitative features of design solutions.

Following experimentations on the model we have suggested to structure design thinking to take the functionality of an urban structure as a priority condition for urban design solutions. Going from that level of certainty to face uncertainty by applying constraints that further define the solution space we find that creativity is not restrained by our structured and knowledge-based design approach. This comes in response to the claim that scientific approaches in design thinking would hold designs from being creative. For that, we review the determinism of the previously discussed parameters over design. As the parameters aid initial design decisions and ascertain the first steps towards formalizing design solutions, they provide no unique design solution by themselves. The space of design creativity is open for infinite types of variations. The parameters therefore constitute the first two sets of constrains towards narrowing the solution space. Hence, this approach aids design reasoning by prioritizing the knowledge that defines a functional framework. The constraints outlined in this approach react towards the temporal conditions of the parts-whole city structure and partially contribute to the problem definition. For a more correct reflection, the parameters might need to respond to real-time dynamics of the structure and what this implies on the volume and function. Given that such dynamics might be associated with a slower pace dynamics on the level of the form-function parameters, we can ignore the latter dynamics at this stage. Considering these variables as static, more evidence is needed to validate the second order assumptions that speculate about the interrelationship between different form-function parameters. On the computational side, effort should be made on presenting a better definition for the target space and further enable a more responsive modeling approach to visualize the results. With this in mind, developments can be made on this model that may consequently lead to conserve problem solving effort especially when handling complex large scale urban developments. Creative outputs might challenge architectural skepticism about using knowledge-based models in design. This is seeing that through exploring different variations, there are unlimited degrees of freedom in the universe of design possibilities for an architect to innovate and involve personal input into design process.

**Acknowledgments** This paper is part of ongoing PhD research at the Bartlett, UCL. The PhD is funded by Damascus University. The author wishes to thank Alasdair Turner, Sean Hanna and Alan Penn for their supervisory support. Partial funding was given by the UCL Space Group platform grant.

## References

1. Simon HA (1984) The structure of ill-structured problems. In: Cross N (ed) Developments in design methodology. Wiley, New York, pp 317–327
2. Simon HA (1957) Models of man: social and rational. Wiley, London
3. Mandel E (1985) Partially independent variables and internal logic in classical Marxist economic analysis. Social Sci Inf 24(3):487–488 (reprinted in Ulf Himmelstrand, Interfaces in Economic and Social Analysis, London 1992)
4. Snodgrass A, Coyne R (1997) Is designing hermeneutical? Architectural Theor Rev 2(1):65–97
5. Jacobs J (1964) Death and life of great American cities—the failure of town planning. Penguin Books, Harmondsworth
6. Hillier B (2002) A Theory of the city as object: or, how spatial laws mediate the social construction of urban space. Urban Des Int 7:153–179
7. Al Sayed K, Turner A, Hanna S (2012) Generative structures in cities. In: Greene M, Reyes J, Castro A (eds) Proceedings of the 8th international space syntax symposium, Santiago, Chile
8. Al Sayed K, Turner A (2012) Emergence and self-organization in urban structures. In: Proceedings of AGILE 2012, Avignon, France
9. Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge
10. Hillier B (1996) Space is the machine. Cambridge University Press, Cambridge

11. Hillier B (1996) Cities as movement economies. Urban Des Int 1:41–60
12. Penn A (1993) Intelligent analysis of urban space patterns: graphical interfaces to precedent databases for urban design. In: Proceedings of the international symposium on computer-assisted cartography, vol 1, pp 53–62
13. Penn A (2006) Synthetic networks—spatial, social, structural and computational. BT Technol J 24(3):49–56
14. Alexander C (1964) Notes on the synthesis of form. Harvard University Press, Cambridge
15. Penn A, Hillier B, Banister D, Xu J (1998) Configurational modelling of urban movement networks. Environ Plan 25(1):59–84
16. Karimi K, Amir A, Shafiei K (2007) Evidence-based spatial intervention for regeneration of informal settlements: the case of Jeddah central unplanned areas. In: Proceedings of the 6th international space syntax symposium
17. Al Sayed K, Turner A (under review) Parameterizing Space Syntax: does space determine urban volume and function? Environ Plan B Plan Des
18. Schon D (1983) The reflective practitioner. Basic Books, New York
19. Batty M (1991) Cities as fractals: simulating growth and form. In: Crilly AJ, Earnshaw RA, Jones H (eds) Fractals and chaos. Springer, New York, pp 43–69
20. Batty M (2005) Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals. MIT Press, Cambridge
21. Turner A (2000) Angular analysis: a method for the quantification of space. Working Paper 23, Centre for Advanced Spatial Analysis, UCL, London
22. Hillier B, Turner A, Yang T, Park H (2007) Metric and topo-geometric properties of urban street networks: some convergences, divergences and new results. In: Proceedings of the 6th international space syntax symposium
23. Turner A (2011) UCL Depthmap: spatial network analysis software, version 10. University College London, VR Centre of the Built Environment, London

# Quantified Study of the Aesthetic Appeal of the Formal Conceptual Elements in New Products Design Through Conjoint Analysis

**Fernán Acevedo López and Jorge Alcaide Marzal**

**Abstract** Would it be possible to know exactly what do people want regarding the form of the products they buy? Different studies were created to give us such answer, but those studies tend to use images of already existing products and because of that, such studies throw much more than the answers that we are looking for. Besides what people want, those studies reveal what the brands want people to want. That is why the following study makes an analysis of some of the various formal elements used in the design of new products, highlighting some representative ones to use them in a conjoint analysis study that will show which are the most appreciated formal attributes in a specific demographic group.

## Introduction

Design methods and the study of form in product design processes are important because they allow designers to structure their work in a methodical and reasoned way while, at the same time, leave room for intuition and creativity to flourish. The study of the perception of the products and their relation to the observer's emotional response is one of the newest disciplines and there are already numerous works published.

Most of these studies are focus on the analysis of the response that causes the image of some product in the consumer. At present time the aesthetics of the products is the key factor in new product design even more than before, and this has come to be because the market give us a large scope of products to buy that have the same function and they are only set apart from each other because of how each one looks. Thus, aesthetics is use to give products an added value.

---

F. A. López (✉) · J. A. Marzal
Universidad Politécnica de Valencia, Valencia, Spain
e-mail: facevedolopez@gmail.com

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_9,

On his book about the design of objects, Norman [1] makes a reasoned analysis of why design works best with attractive shapes and concepts.

He gives an explanation from a psychological point of view, focusing his research in three processing stages or levels of the human psyche: First, the visceral, this level could also be called the intuitive. Second, the behavioural or the level of customs and manners, where the decision making is made almost immediately for we do what we are used to do. And third, the reflexive level, that, as it name implies, this is the process in which we analyse the thinking process. What Norman proposes on this book is to make designs that appeal more to the first levels because those are the ones that could be seen as instinctive or intuitive and there for these are the easiest to approach and the harder ones to change as individuals.

Other authors, such as: Veryzer [2], explore the systematic nature of aesthetic responses to products and propose a conceptualisation of the aesthetic response to design principles, and he operate this principles on internal transformation as algorithms.

Hsu et al. [3] conducted a study that investigated the difference in the perception of the form of products between designers and the users. They used the semantic differential method, for this they showed images of telephones to the two groups and analysed their response to the formal elements of the designs.

Mokarian [4] developed a study that tested the aesthetic value of formal balance in objects. The author highlights the problem that presents the presentation of engineering design objects to the public, as it primarily focus is on the functional aspects, not the aesthetic ones. None the less, the study has two main hypotheses: the first is that you can make a classification of formal variables from the design in terms of their functional relevance, ergonomics and aesthetics, and the second hypothesis is that the formal element of balance helps to achieve product aesthetics, resulting in attractive objects. The paper concludes that to have an effective integration of the functional, ergonomic and aesthetic variables in the design of new objects, there is the need for an analysis and a classification of the formal variables in the design.

In a different line of work, Sharmin [5] presented a study in which he analysed the patterns of eye movements of people seeing design objects, with the help of a tracking eye movement device, he was able to measure the number and duration of stops that the eye of the subject had while seeing a picture of a product. These stops were identified as points of interest or attraction to the observer.

The work of Wrigley et al. [6] proposes that the design process should start by targeting the hedonistic visceral experience that the buyer have when introduced to a new product. The importance of analysing the hedonistic visceral reactions according to the authors is that these reactions will be responsible for creating guidelines to design attractive products, and by knowing how the costumer it's going to react is how one can create designs that will have better market acceptance.

There are other studies that have attempted to measure the formal appeal of the products. For example, Osborn et al. [7] performed a selection of features underlying the form, and these features were then used as attributes in a utility equation. It concludes that, once preferences are outlined in the utility equation, the results can be used as a base for the generation and modification or design verification on new products.

Following this line of research, the present study undertook the task of identifying some general formal elements of products and general elements of forms and shapes that were thought to be attractive to the consumers, to be used later on a conjoint analysis to see which are the most appreciated in a trial group. To achieve this it was necessary to identify a set of general elements that could constitute a form (shape). Such elements could be the contrast, rhythm, colour, texture, etc. The explanation and uses of these elements were based on the design fundamentals book by Scott [8].

## Hypothesis and Objectives

The main observation with the studies and methods for the appreciation of needs, tastes and likes on products is that they are always tampered by the subjectivity of the person that the study is analysing, almost all studies try to capture and understand through an objective way a sensation or feeling that are by definition subjective. It would be interesting to develop a method that could show what people are really felling without them being any the wiser about it, this way we could really know what do they really want.

What this study proposes is to change the approach to the participant, and to achieve it this study will be using the same type of methods proven accurate by others, but changing the object of study: in this case, instead of pictures of products, the participant will be seeing abstract images with no references to any product. This way we will be able to analyse the elements of the product form instead of the paradigms around the products. The goal of this is to prove that people give a sincerer answer when they are asked whether they like or dislike something abstract, and also to show that if the information throw by the survey were used on the design of a new product, that product would have better acceptance among potential buyers.

Thus, the aim of this quantified study is to determine which are the attributes (between the selected ones for the survey) that people find to be more attractive. To achieve this we need to break off the elements of the form from the products that contains them, and to do so this study bases itself in the classification of elements found in Scott [8] and Wong [9] book. If these elements can be extracted so they can’t be linked to the original object, we will get a separate design criteria and guidelines for the cataloging and quantification of products and objects.

## Methodology

In order to know the individual contributions of each formal element to the attractiveness of a product, it was decided to conduct a survey that was processed through the Conjoint Analysis (AC). A short and concise explanation for this survey method is given by Joseph Curry:

Conjoint analysis is a popular marketing research technique that marketers use to determine what features a new product should have and how it should be priced. Conjoint analysis became popular because it was a far less expensive and more flexible way to address these issues than concept testing. […] three steps: collecting trade-offs, estimating buyer value systems, and making choice predictions, form the basics of conjoint analysis. Although trade-off matrices are useful for explaining conjoint analysis as in this example, not many researchers use them nowadays. It's easier to collect conjoint data by having respondents rank or rate concept statements [10].

The survey was applied to a specific market niche: young people between 20 and 30 years old, from an A, B+ and B demographic group. Also the survey was developed in paper and pencil format with the system of paired data comparison collection to facilitate the survey and make it faster. Once the data was collected, it was processed using the software: SMRT Sawtooth Software, Inc. The data analysis was conducted using multiple linear regressions.

The preference model used was the partial benefit and the selected attributes shown in the Fig. 1.

The choice of levels in the "Form" attribute was made searching for two opposite elements of the form. The conclusion was that, within the lines that make any figure, there are two types: the straight and the curve lines, from now on such lines will be referred to as inorganic and organic according to the notion that says that organic shapes have a natural look with a flowing and curving appearance, whereas the inorganic would be the exact opposite. In the case of the levels under "Contrast" the choosing of these elements were made with the idea in mind of showing one element that was easy to see and another one that wasn't. Finally, for the last attribute "Proportion (rhythm)" the levels were chosen to give a wider range of variations, choosing, once again, two opposites and one element in between. All this attributes were chosen because they were the easiest ones to apply on the survey since it was thought of to be done on paper, although other attribute that would have being interesting to examine was texture but, to be able to do so there was the need to add samples of different materials which weren't available.

Using these attributes and levels we generated the following images to be used in the conjoint analysis study (Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 and 14).

As it can be seen, and for reasons of consistency with the study, the presented images are formed by sequences of squares and circles. With the idea that the respondent shouldn't be able to easily think his or her answer, the corners of these circles and squares were cut with a cross, in order for them to not be obvious

| ATTRIBUTES | | |
|---|---|---|
| **Form** | **Contrast** | **Proportion** |
| **LEVELS** | | |
| Organic | Hi | Simetry |
| Inorganic | Low | Progretion |
| | | Asimetric |

**Fig. 1** Attributes and levels

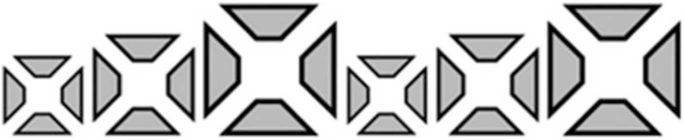


**Fig. 2** Inorganic, high contrast with progression image

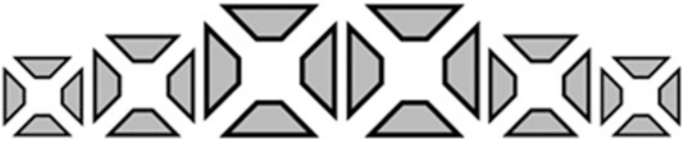


**Fig. 3** Inorganic, high contrast and symmetric image

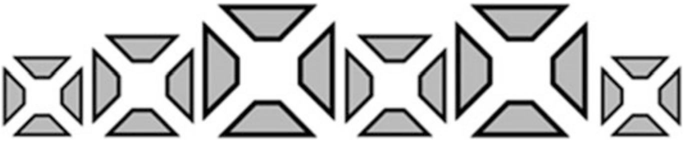


**Fig. 4** Inorganic, high contrast and asymmetric image

**Fig. 5** Inorganic, low contrast with progression image

**Fig. 6** Inorganic, low contrast and symmetric image

**Fig. 7** Inorganic, low contrast and asymmetric image

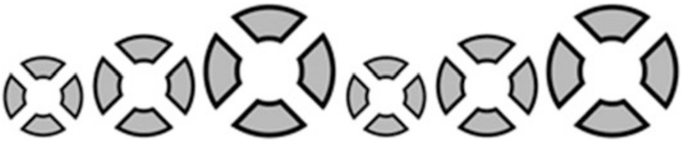


**Fig. 8** Organic, high contrast with progression image

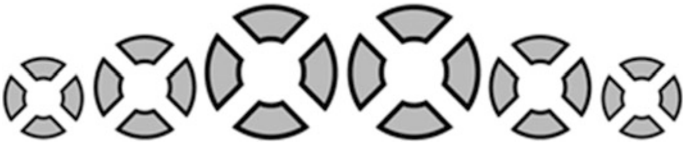


**Fig. 9** Organic, high contrast and symmetric image

**Fig. 10** Organic, high contrast and asymmetric image

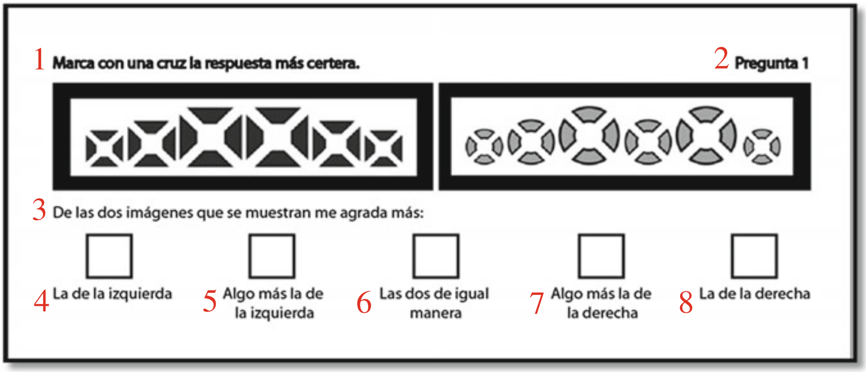
circles and squares, through thanks to the principle of wholeness (Gestalt), our mind adds the missing elements to complete a figure and that's why the figures will still be perceived as circles and squares (the principles of organic and inorganic shapes) [8].

The survey was applied to a pilot group of twenty people at a shopping mall on a working day at a working hour (12 men and 8 women).

**Fig. 11** Organic, low contrast with progression image

**Fig. 12** Organic, low contrast and symmetric image

**Fig. 13** Organic, low contrast and asymmetric image



1 **Marca con una cruz la respuesta más certera.**

2 **Pregunta 1**

3 De las dos imágenes que se muestran me agrada más:

4 ☐ La de la izquierda
5 ☐ Algo más la de la izquierda
6 ☐ Las dos de igual manera
7 ☐ Algo más la de la derecha
8 ☐ La de la derecha

**Fig. 14** Example of the survey card

Translation of the original survey card:

1. Cross the most accurate response.
2. Question 1.
3. Which of the two images do you like the best.
4. The one on the left.
5. A little bit more the one on the left.

6. I like both images the same.
7. A little bit more the one on the right.
8. The one on the right.

## Survey Results

This survey was analysed using the SMRT Sawtooth Software and the results from a 40 % women and 60 % men group are in Fig. 15.

The goodness of fit, represented by square R is 0.8235 and therefore we considered acceptable the result of this survey. The utilities are shown in Fig. 16.

Figures 17, 18, 19 and 20 show the results in a graphic mode.

## Confirmation Survey

To confirm the reliability of the results a second survey was developed. This survey showed images of two objects of design that the original group could have been interested in according to their ages and demographics. We asked people within the same specifications than the first survey group, which of the two objects they liked most, if the symmetrical, contrasted and square design, or the arrhythmic, without contrast and circular one. These combinations were chosen because they were the two extremes of the results of the original survey.

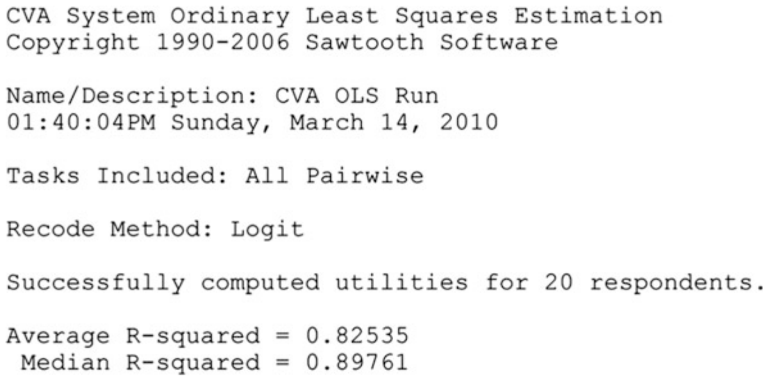
The images displayed to people are shown in Figs. 21 and 22.

To do the second survey we interview twenty new people and from those eleven were men and the rest were women. Once again, the survey took place at a shopping mall on a working day at a working hour.

Thirteen people of the survey preferred sound system 1 and seven sound system 2. That means that 65 % of the people did like the sound system that the study said had to have better acceptance and only a 35 % choose the opposite which is consistent with the original survey.

## Discussion

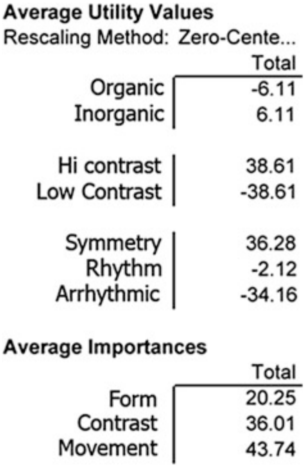
The study showed that objects that have a square shape, or have predominating straight lines, are symmetrical as a whole and have contrast between the colours of its elements, or components, shall, in accordance with the experiment, have a greater acceptance. On the other side, objects that are arrhythmic with circular or curved forms prevailing, and with a low colour contrast, will be less accepted by people.



```
CVA System Ordinary Least Squares Estimation
Copyright 1990-2006 Sawtooth Software

Name/Description: CVA OLS Run
01:40:04PM Sunday, March 14, 2010

Tasks Included: All Pairwise

Recode Method: Logit

Successfully computed utilities for 20 respondents.

Average R-squared = 0.82535
 Median R-squared = 0.89761
```

**Fig. 15** Survey results

**Fig. 16** Utilities



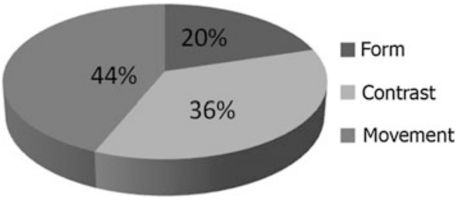
**Average Utility Values**
Rescaling Method: Zero-Cente...

| | Total |
|---|---|
| Organic | -6.11 |
| Inorganic | 6.11 |
| Hi contrast | 38.61 |
| Low Contrast | -38.61 |
| Symmetry | 36.28 |
| Rhythm | -2.12 |
| Arrhythmic | -34.16 |

**Average Importances**

| | Total |
|---|---|
| Form | 20.25 |
| Contrast | 36.01 |
| Movement | 43.74 |

**Fig. 17** Importance of the attributes for the users



One of the most interesting things is the importance of the attributes for the respondents. In theory and according to the survey, people shall, first of all, appreciate the apparent motion of an object, and then appreciate the contrast, and finally the form. This is interesting because it forces us to consider that the form is not the most valued attribute.

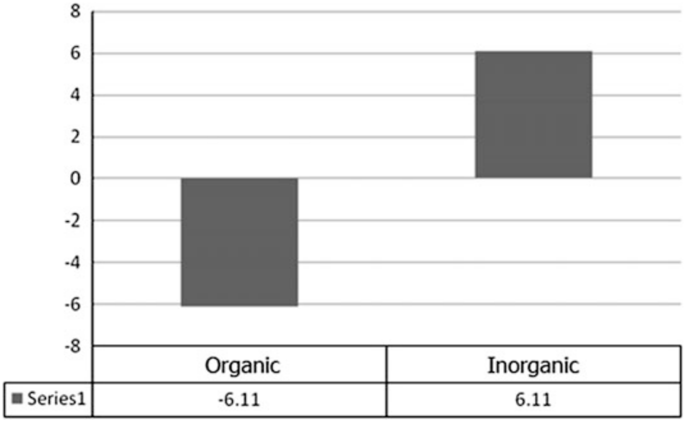


Fig. 18 Partial preferences per level (form)

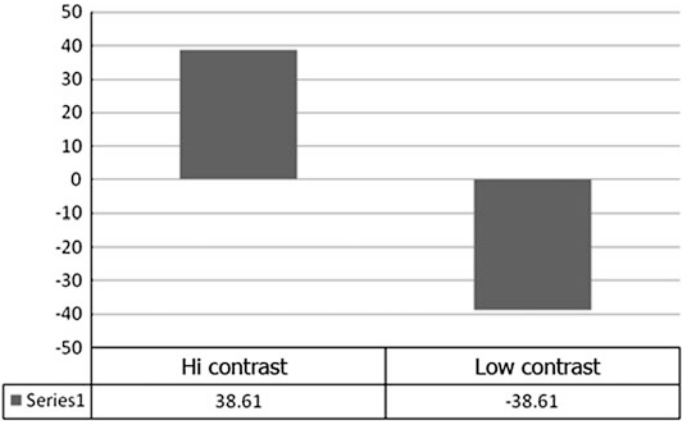


Fig. 19 Partial preferences per level (contrast)

Movement was the most important attribute to respondents and that can be explained by the fact that symmetry is something that human seeks instinctively [11]. It was, therefore, expected that the survey reinforced that premise.

Regarding the contrast, it was found that respondents preferred an obvious contrast in images over the low contrast and more subtle ones.

Finally, regarding the formal elements, although this attribute had the lowest weight in the decision of the respondents, people preferred the square forms over the circular ones.

The validation survey confirmed the results obtained in the first survey: Stereo 1 was the one that had the best acceptance and was more pleasing to the respondents, this sound system was the one that had the characteristics of symmetry, contrast and squared shape, which were determinant in the first survey as the resulting

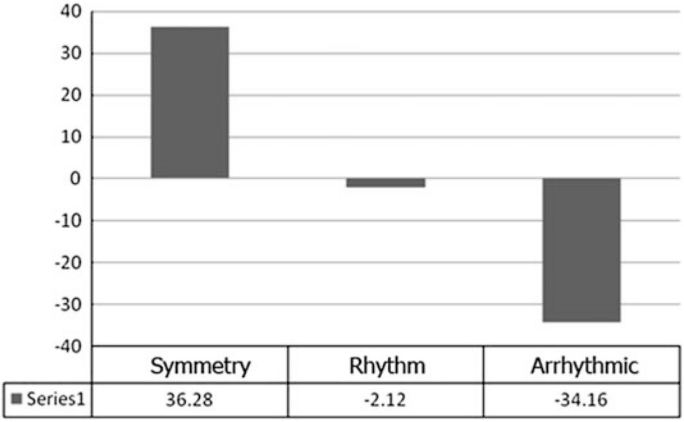


**Fig. 20** Partial preferences per level (movement)

**Fig. 21** Stereo 1

preferences. This allows us to say that the perceptual response of the user to the abstract forms corresponds with the respond given when evaluating specific things, in this case, sound systems.

With this we can say that, for the purposes of a first attempt at analysing the aesthetic appeal of the conceptual elements of the form, the results presented in this study are accurate and valid from a methodological perspective. However, they are only an approximation to the overall vision of the market, since it only covers the perspective and scope explained on this text. Further studies will be needed to refine the results.

**Fig. 22** Stereo 2

## Conclusion

In this study we could observe and verify the existence of the relationship between the perception of abstract forms and the concrete forms of the products in the minds of consumers, which can generate a whole new line of research according to this findings.

Specifically, the study outcome support the conclusion that the proposed method yields results consistent with the general hypothesis. It also allows us to say that, as in any other conjoint analysis study, the attributes to be analysed can vary, by this we mean that we can use different elements to do the study for different cases.

Within the possible areas to continue this study we see the possibility of varying the way of presenting the stimuli to people. It would be possible, for example, to prepare physical samples with different materials, finishes, contrasts, shapes and silhouettes (proportions), this would allow us to go from two-dimensional visual stimuli to visual and tactile stimuli in three dimensions.

Another variation of this study could be applying it in the field of ethnography. In this case, one would have to change the collection data mode of the survey by observing people interacting with different objects in a store (the objects are to be previously classified within the different elements of the form) and the tabulation could be done depending the reaction to these objects, this way one can study the change between present images to an object itself. The values could be given by the approaching of people to the objects, so if people do not approach an object, this object would get a score of zero, if the subject approaches the object and looks at it, the score would be 1, and if he comes and touches it, the score would be 2. This scoreboard is proposed to measure the level of interest of people with objects and hence whether they like them more or less.

Finally, it is important to clarify that the intention of this study wasn't to elaborate a set of attributes to be use in the design of new products, this exercise had the task to try a hypothesis and meant to open the path to a new research line, therefore the survey had a very reduced amount of participants, the intention being to first confirm if the idea could be apply to a small pilot group. With this in mind, this study is rather a pre-study, so to speak: the confirmation survey was conducted in an expedite manner just to see if the hypothesis could be proven, now that we know it has work, the hole experiment should be re done to expand its grasp.

## References

1. Norman DA (2005) El diseño emocional: Por qué nos gustan (o no) los objetos cotidianos. Ediciones Paidos, Barcelona
2. Veryzer RW (1993) Aesthetic response and the influence of design principles on product preferences. Adv Consum Res 20:244–248
3. Hsu SH, Chuang MC, Chang CC (2000) A semantic differential study of designers and users product form perception. Int J Ind Ergon 25(4):375–391
4. Mokarian MA (2007) Visual balance in engineering design for aesthetic value. A thesis submitted to the College of Graduate Studies and Research in Partial Fulfillment of the Requirements for the Degree of Master of Science in the Department of Mechanical Engineering. University of Saskatchewan. Saskatoon
5. Sharmin S (2004) Studies of human perception on design products. M.Sc. thesis, University of Tampere, Department of Computer Sciences
6. Wrigley C, Popovic V, Chamorro-Koc M (2008) Visceral hedonic rhetoric: emerging research in design and emotion. In: 6th design & emotion conference, dare to desire, The Hong Kong Polytechnic University, Hong Kong
7. Orsborn S, Cagan J, Boatright P (2009) Quantifying aesthetic form preference in a utility function. J Mech Des 131(6):061001-1–061001-10
8. Scott RG (2002) Fundamentos del diseño. México D.F. Editorial Limusa
9. Wong W (2001) Fundamentos del diseño, 3ª edn. Ediciones G. Gili, Barcelona
10. Curry J (1996) Sawtooth software, research paper series: understanding conjoint analysis in 15 minutes. Available online: http://sawtoothsoftware.com/download/techpap/undca15.pdf
11. Euclides (1996) Libros X–XIII: Elementos, vol 3. Gredos, Madrid
12. Aguayo F, Soltero VM (2003) Metodología del diseño industrial. Un enfoque de la ingeniería Concurrente, Ed. Rama, Madrid
13. Aicher O (2002) El mundo como proyecto (4ª). Barcelona. Ediciones G. Gilli
14. Alcaide J (n.d.) Tema 3: Análisis conjunto de valor. Obtenido de la Universidad Politécnica de Valencia del Programa de diseño, fabricación y gestión de proyectos industriales para la clase de Métodos avanzados en ingeniería de proyectos
15. González M, Lama JR, Pérez JR (2009) Ingeniería Kansei para un diseño de productos centrado en los usuarios. Técnica Industrial. 280. Obtenido de http://www.tecnicaindustrial.es/TIFrontal/a-2491-Ingenieria-Kansei-diseño-productos-centrado-usuarios.aspx
16. Cai H, He W, Zhang D (2003) A semantic style driving method for products appearance design. J Mater Process Technol 139(1):233–236
17. Léon D (2005) Metodología para la detección de requerimientos subjetivos en el diseño de producto; Universidad Politécnica de Cataluña: Tesis doctoral
18. Lobach B (1981) Diseño industrial. Gustavo Pili, España
19. Potter N (1999) ¿Qué es un diseñador? Editorioa Paidos, Barcelona

# Evaluating Creativity in Parametric Design Processes and Products: A Pilot Study

**Ju Hyun Lee, Ning Gu, Julie Jupp and Sue Sherratt**

**Abstract** Parametric design is an emerging research issue in the design domain. However, our current understanding of creativity in relation to either a process or product standpoint is limited. This paper presents a formal approach for the description and identification of creativity from both perspectives. The framework combines: (1) protocol analysis for encoding cognitive design activities, providing a process-based evaluation of creativity, and (2) consensual assessment of parametric products, providing a product-based evaluation of creativity. The coding scheme is based on the creative acts: Representation, Perception, and Searching for a Solution. The consensual assessment technique is based on a series of creativity evaluations undertaken by an expert panel. The effectiveness of this approach was examined in a pilot study. Findings show the capture of cognitive activities and identification of creative patterns, revealing how they correspond to the creativity levels of parametric design products. The results identify conditions that have the potential to enhance creativity in parametric design. This research provides a promising procedure not yet available and contributes to the development and verification of a formal approach for evaluating creativity in parametric design.

## Introduction

Parametric design has become an increasingly popular approach to Computer-aided Design (CAD), resulting in the emergence of a global architectural style, known as 'parametricism' [1]. The popularity of this approach is due, on the one

---

J. H. Lee (✉) · N. Gu · S. Sherratt
University of Newcastle, Newcastle, NSW, Australia
e-mail: Juhyun.Lee@newcastle.edu.au

J. Jupp
University of Technology Sydney, Sydney, Australia

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_10,

hand, to the generation of unusual forms, often with complex geometries and increased technical sophistication; on the other hand, by virtue of the recognition of these designs being novel, useful and, arguably, creative—important goals in non-routine design.

Whilst a profusion of parametric design solutions have been constructed and widely illustrated in architectural magazines, studies of parametric design activities and outputs is relatively nascent. Consequently, our understanding of the generative and evolutionary aspects of parametric design and the role of creativity—from either a process or product standpoint—is limited.

Our investigation explores this knowledge gap. This paper presents a formal evaluation framework developed by the authors for assessing creativity in parametric design. This framework is based on a two-pronged approach: (1) a protocol analysis procedure—providing a process-based evaluation of creativity, and (2) a selective criteria-based assessment method, providing a product-based evaluation of creativity. The remainder of this paper is divided into four parts. Section "Background" explores the related literature. Section "Framework for Evaluating Creativity in Parametric Design" then presents a conceptual framework for evaluating creativity in parametric design. A pilot study illustrating the research approach and framework is presented in section "Pilot Study", detailing its implementation and resulting empirical data. Finally, section "Discussion and Conclusion" concludes with a discussion of the empirical evidence and outlines the directions for future work.

## Background

Creativity has been described in terms such as "creative thinking", "problem solving", "imagination", or "innovation" [2]. Design activities involve problem solving [3], which could be characterized as a cognitive process [4]. Creativity should be a natural component of the design process [5]. In the design domain, process and cognition are core themes [6], and much of the related research has aimed to enhance design creativity. Researchers [5, 7–9] have studied designers' sketching activities in attempts to understand the creative process in the early conceptual design stages, revealing that sketching activities are linked in various ways to creativity. In other approaches, researchers have compared sketching activities with those used in the application of traditional CAD tools [4, 10]. These studies show that sketching, with its flexible and intuitive characteristics, has advantages over CAD tools in terms of creativity. Ibrahim and Rahimian [11] illustrate through their studies that both sketching and conventional CAD tools have limitations in supporting conceptual design. Sketching has limitations in constructing complex designs. Conventional CAD tools can hinder creativity due to their inflexibility.

Design studies indicate that sketching enhances creativity through reconstructing [4, 10, 12] and regulating [8] mental imagery. Thus, in order for digital design techniques to become more flexible and intuitive, they should provide interactive imagery so as to enable the designer to perceive visuo-spatial features and organizational relations, and generate alternative solutions [4]. Such techniques need to also consider the management of part-whole relationships, design hierarchies, topology-geometry relationships, structuring of ill-structured problems, and restructuring of problem parameters [8]. Research has shown these characteristics are evident in parametric design [1, 13, 14]. Advances in such digital design techniques are providing an increasing capacity to encode and evaluate generative processes that can intuitively support design exploration whilst maintaining flexibility. With the complexity and power of contemporary parametric design software, the impact of such tools on creativity in the conceptual design phases remains relatively unexplored. However, recent research has illustrated that parametric design can support creativity [15, 16], with some parametric design tools providing intuitive restructuring and flexible regulating environments [15, 16], thereby allowing interactions that can engage complexity [13, 14].

Lee et al. [2] explained that parametric design is related to both divergent and convergent thinking, two of the most important processes in models of creativity. Divergent thinking is connected to the parameters and generative rules available in parametric design environments, while convergent thinking comprises the rules which define constraints for the most correct (or satisfactory) answer to a design question. Iordanova et al. [15] argue that generative modeling using parametric design tools can contribute to creativity. Chusllp and Jin [17] proposed a cognitive activity model consisting of three loops for problem redefinition, idea stimulation, and concept reuse. Generative parametric-based design solutions are evolved through extensive iteration and regeneration by modified parameters and rules (algorithms). Scripting or coding activities are regarded as a channel for creativity and a means of representing design ideas [16]. Thus, whilst generative parametric modeling presents a different context for design during the conceptual stages than that of sketching, divergent and convergent thinking is maintained. From this perspective, parametric design is capable of supporting not only the generation of ideas but also their evolution, leading to creativity. This paper therefore focuses on the interactive and algorithmic activities of parametric design that are significant in the creative process which underlie the generation of creative outcomes.

## Framework for Evaluating Creativity in Parametric Design

The framework for evaluating creativity in parametric design is based on the key work of Rhodes [18]. Rhodes aimed to deal with both a cognitive approach to the design process and a confluence approach to design products. This approach [18] classifies strands of creativity into four perspectives—known as the four Ps of creativity: person, process, press (environment), and product. This conceptualization of creativity recognizes that design environments are influenced by physical and social contexts that can affect the creative design process, the quality of the

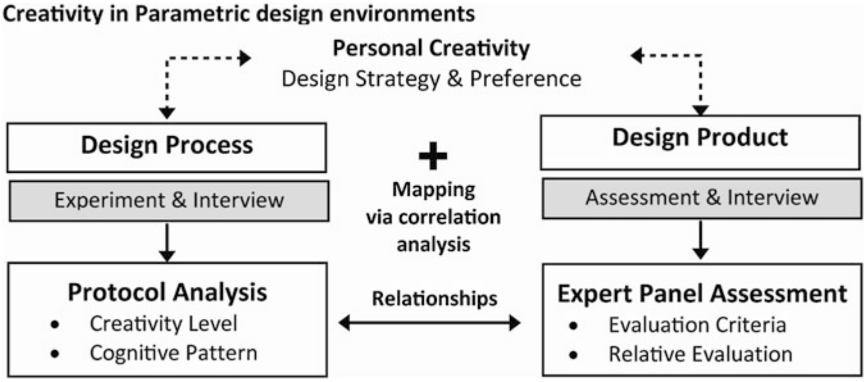


**Fig. 1** A framework for evaluating creativity in parametric design

design product, and levels of personal creativity. However, it is difficult to evaluate and investigate all design environments within a single research framework. Simonton [19] suggests that all three perspectives except for ‘press’ should be integrated into a unified view of scientific creativity. Although Hasirci and Demirkan [5] present a framework derived from Rhodes’ four Ps, they also paid attention to the relationships among the three essential components of creativity (still excluding ‘press’). They claimed that person and product follow process. Design process must therefore be a core theme in the study of creativity as it can reveal the characteristics of creativity. This approach is significant to the study of creativity in parametric design.

Figure 1 illustrates our proposed framework for evaluating creativity in parametric design.

Taking into account Rhodes’ four P’s, the framework will be applied to designing (process) and design (product) in parametric design environments (press), whilst also accounting for design strategies and preferences as part of Rhodes’ personal creativity (person). The framework is divided into two components, viz. design process and design product. The evaluation of the creative design process uses protocol analysis and the evaluation of the design outcomes adopts an expert assessment approach based on the Consensual Assessment Technique (CAT) [20].

This research argues that by encoding segments of the design process in sequence and using the coding scheme developed here, cognitive patterns can be identified which will provide an evaluation of creativity in parametric designing. Furthermore, by mapping these results with the creativity assessments provided by an expert panel, it will be possible to enhance our understanding of process-based creativity relative to product-based creativity. Hence, it was important that the proposed coding scheme be capable of describing levels of creativity and identifying cognitive patterns throughout the creative process.

## *Coding Scheme and Protocol Analysis*

To evaluate creativity in the parametric design process, we have adopted protocol analysis [21–23] which has been widely used to explore cognitive activities in the design process. In protocol analysis, a coding scheme for describing activities of creativity is essential and should be devised to be suitable for the purpose and the design environments. Therefore, for effective analysis, a coding scheme should be specifically developed to suit the process of parametric design.

The two coding schemes developed by Suwa et al. [23] and Gero and Neill [24] have been widely used in the design domain. The former targets the content-oriented aspects of designing, whilst the latter targets the process-oriented aspects of designing [25]. For this research, the code developed by Suwa et al. [23] is suitable for examining the representation and perception aspects in parametric designing. Similarly, the code developed by Gero and Neill [24] is valuable for studying aspects of searching for solutions in parametric designing. The coding scheme for analyzing how parametric design process supports creativity (see Table 1) is based on the adaptation from these two influential coding schemes.

For our coding scheme, three levels of cognitive processes identifying creativity in the design process are defined: 'Representation', 'Perception', and 'Searching for a Solution'. These levels are informed by the cognitive processes in creativity identified by Hayes [26]. To generate a cognitive pattern suitable to the conditions of parametric design, each level is then separated into two subcategories: (1) Geometry, and (2) Algorithm, whilst 'Searching for a Solution' comprises three subcategories, viz.: Finding Idea, Evaluation, and Adopting Idea. Subclasses were then defined for detailed actions based on both design [27] and cognitive actions [22].

## *Expert Panel Assessment*

The use of 'expert panel assessments' [28, 29] based on CAT [9, 16] is a valuable and often-used approach for measuring the creativity of design products. The CAT technique is based on the notion that a measure of the creativity of an artifact is the combined assessment of experts in that field. Unlike other measures of creativity, CAT is not based on any particular theory of creativity, which means that its validity (which has been well established empirically) is not dependent upon the validity of any particular theory of creativity [30]. In employing this technique, a basic requirement of an expert panel assessment is that its members are familiar with the design domain and the techniques required to produce the design [28, 31].

From this standpoint, a series of assessment tasks were designed to evaluate the parametric design models, namely:

1. Independent non-criteria based assessment—using their own judgment of creativity, panel members must evaluate the creativity of each design model independently from the other models.

**Table 1** Coding scheme for exploring the parametric design process

| Creativity level | Category | Subclasses | Description |
|---|---|---|---|
| Representation | Geometry | RG-Geometry | Create geometries without an algorithm |
| | | RG-Change | Change existing geometries |
| | Algorithm | RA-Parameter | Create initial parameters |
| | | RA-Change Parameter | Change existing parameters |
| | | RA-Rule | Create initial rules |
| | | RA-ChangeRule | Change existing rules |
| | | RA-Reference | Retrieve or get references |
| | | R-Generation | Make generation (or variation) |
| Perception | Geometry | PG-Geometry | Attend to existing geometries |
| | Algorithm | PA-Algorithm | Attend to existing algorithms |
| | | PA-Reference | Attend to existing references |
| Searching for a solution | Finding Idea | SF-Initial Goal | Introduce new ideas (or goals) based on a given design brief |
| | (Geometry) | SF-Geometry SubGoal | Introduce new geometric ideas extended from a previous idea |
| | (Algorithm) | SF-Algorithm SubGoal | Introduce new algorithmic ideas extended from a previous idea |
| | Evaluation | SE-Geometry | Evaluate primitives or existing geometries |
| | (Algorithm) | SE-Parameter | Evaluate existing parameters |
| | | SE-Rule | Evaluate existing rules |
| | | SE-Reference | Evaluate existing references |
| | Adopting Idea | SA-Geometry | Adopt new ideas to geometries |
| | (Algorithm) | SA-Parameter | Adopt new ideas to parameters |
| | | SA-Rule | Adopt new ideas to rules |
| | | SA-Reference | Adopt new ideas to retrieve or get internal/external references |

2. Comparative non-criteria based assessment—using their own judgment of creativity, panel members must evaluate design models relative to one another.
3. Criterion-based assessment—using specific evaluation criteria, the panel must evaluate each design model.

Evaluation criteria were selected by reviewing the literature on creativity assessment [20, 32]. Criteria are often comprised of three categories: novelty, usefulness, and aesthetics. Based on these, a number of subscales can then be derived. For example, in the Creative Product Semantic Scale, or CPSS, Christiaans [29] used a large number (70 in total) of bipolar subscales. However, such a large number of criteria would be time-consuming and even confusing for assessors. The subscales in our research were selectively adopted from the CAT [20, 32] and CPSS methods [33]. Both instruments have been widely used in design creativity assessment. Table 2 shows the evaluation criteria selected here to assess the creativity of parametric design products.

**Table 2** Criteria for assessing the creativity parametric design products

| Novelty | Usefulness | Aesthetics |
|---|---|---|
| Originality | Technical quality | Attractiveness |
| Innovation | Functional quality | Expressiveness |
| | Integration Capacity | Complexity |

The pilot study, presented in section Pilot Study, utilizes all three assessment tasks and evaluation criteria, including Novelty, Usefulness and Aesthetic-based Complexity. However the subscales shown in Table 2 were not tested. Whilst the subscales require further development, the pilot study focuses on the parametric design process in relation to evaluations of product-based creativity relative to the three main criteria mentioned above.

### *Mapping via Correlation Analysis*

Finally, the framework includes mapping the cognitive activities of the design process and the expert panel's assessment. Correlation analysis is used to identify particular process patterns in parametric designing that may enhance or hinder different aspects of creative design. Participants with similar results on the expert panel assessment are grouped together. Correlation analysis is then used to examine significant similarities or differences in the protocol analysis results within and across groups.

## Pilot Study

The pilot study tests the two components of the formal framework and the correlation analysis mapping technique so as to assess their effectiveness. Thus, a parametric design experiment and an expert panel assessment were conducted.

### *Process-Based Evaluation of Creativity*

#### Design Experiment Procedure

A written design brief was given to participants and described verbally by the experimenter. The brief concerned the conceptual design of a high-rise building. This is a simple form generation design task, containing five specific design requirements, viz. that the high-rise building will (1) have two main functional areas of offices and a hotel; (2) have a maximum floor area of 2,500 $m^2$ (50 m × 50 m) per floor; (3) be over 40 stories high; (4) reflect transformations of structural forces using external data (optional); and (5) be a designated regional landmark. Participants were instructed to "think-aloud" or provide a running commentary of their actions and thoughts.

Participants were given 1 h to undertake the design task and were video-recorded utilizing parametric modeling tools of their choice, including, e.g., Rhino, Grasshopper, Maya, and Python scripting. Further, as parametric scripting could be very technical for designers and the time restriction could be a potential source of stress for participants and limit their designing, all participants were informed that they could continue if more time was required. Design deliverables were specified in the brief, namely: (1) a 3D model and related files, and (2) rendered or captured images to clearly represent the conceptual design solution and illustrate its main attributes.

After each design session, the students then participated in a recorded post experiment interview with researchers so as to report and explain their retrospective thoughts and activities in terms of their think-aloud protocols, whilst watching the recorded video. In addition, four specific questions to describe their experiences and preferences on creativity in parametric design environments were posed. Each student used a different parametric design environment: Grasshopper, Maya Script Editor (SE), and Python.

## Protocol Analysis Procedure

The protocols were firstly divided into smaller segments. Suwa et al. [23] categorized methods of segmentation in two approaches, firstly, using pauses or syntactic markers, and the second based on the participant's intentions or the content of thoughts and activities. This study employs the latter technique and segmentation was undertaken by a single researcher. An intention (or content of thought and activities) represents a segment and it was encoded as at least one or as several of the subclasses described in Table 1. Furthermore, segmentation was based on the video recording of the computer screen. The protocols of each recorded video were directly transcribed using NVivo 9 research software and automatically segmented into smaller episodes. The coder encoded the transcriptions using the scheme presented in section "Coding Scheme and Protocol Analysis". The average value of the number of segments was 263.5 (Student A: 142, Student B: 286, Student C: 368). Over 90% of each protocol was encoded in our coding scheme.

The coded data and transcriptions were transferred to Excel 2007 spreadsheets to encode the data series again and inspect the codes themselves. The coded data were visualized in simple graphical forms to facilitate the exploration and identification of patterns and characteristics.

**Table 3** The percentage of coding result of each student's protocol

| Creative Level | Subclass | Student | | | Mean | SD |
|---|---|---|---|---|---|---|
| | | A | B | C | | |
| Representation | RG-Geometry | – | 1.2 | – | 0.4 | 0.7 |
| (RE-Geometry) | RG-Change | – | 2.8 | 0.3 | 1.0 | 1.5 |
| (RE-Algorithm) | RA-Parameter | 5.9 | 2.8 | 0.1 | 2.9 | 2.9 |
| | RA-Change Parameter | 11.2 | 0.9 | 4.6 | 5.6 | 5.2 |
| | RA-Rule | 17.7 | 19.2 | 21.5 | 19.5 | 1.9 |
| | RA-ChangeRule | 10.2 | 26.6 | 11.9 | 16.3 | 9.0 |
| | RA-Reference | – | 2.7 | 0.2 | 1.0 | 1.5 |
| | R-Generation | 0.9 | 2.8 | 12.6 | 5.4 | 6.3 |
| Perception | PG-Geometry | 1.1 | 0.4 | 1.7 | 1.1 | 0.7 |
| | PA-Algorithm | 2.3 | 8.2 | 3.8 | 4.8 | 3.1 |
| | PA-Reference | – | 1.9 | – | 0.6 | 1.1 |
| Searching for a Solution | SF-Initial Goal | 3.9 | 1.0 | 0.3 | 1.7 | 1.9 |
| (Finding Idea) | SF-Geometry SubGoal | 7.1 | 2.2 | 2.4 | 3.9 | 2.8 |
| | SF-Algorithm Subgoal | 6.5 | 1.0 | 4.3 | 3.9 | 2.8 |
| (Evaluation) | SE-Geometry | 18.6 | 8.7 | 16.2 | 14.5 | 5.2 |
| | SE-Parameter | 4.9 | – | – | 1.6 | 2.8 |
| | SE-Rule | 6.3 | 17.0 | 19.0 | 14.1 | 6.8 |
| | SE-Reference | – | – | 0.2 | 0.1 | 0.1 |
| (Adopting Idea) | SA-Geometry | 0.5 | 0.6 | 0.6 | 0.6 | 0.1 |
| | SA-Parameter | 2.5 | – | 0.2 | 0.9 | 1.4 |
| | SA-Rule | 0.4 | – | 0.1 | 0.2 | 0.2 |
| | SA-Reference | – | – | – | 0.0 | 0.0 |
| Sum | | 100 | 100 | 100 | 100 | – |

## Coding Results of the Parametric Design Process

Table 3 shows the coding results describing the percentage of the frequency weighted by time span (calculated by time duration of each code). This allows us to figure out time usage in stages of design process [5] as well as main activities of each participant. On average, the coverage of 'Representation' accounts for 46.4 %, 'Perception' accounts for 22.0 %, and 'Searching for a Solution' accounts for 52.4 %. 'RA-Rule' and 'RA-ChangeRule' were dominant features in each student's parametric design process. There was little 'RG-Geometry'. This suggests that the algorithmic representation is a preferred medium for progressing in the parametric design. The protocol of student A, using a graphical algorithm editor, explicitly made and changed parameters, while in student B's and C's protocol it was difficult to distinguish between the use of parameters and rules. They used a script editor which writes the programming language to progress design.

However, the protocol of Student C shows that when generating a design solution or its variation, existing parameters were changed rather than the rules themselves. Making rules (RA-Rule) accounted for approximately 20 % of all cognitive design activities and changing rules (RA-ChangeRule) accounted for an average of 16.3 %. There was some differentiation among participants regarding the coding result of ‘changing rules’. Students A and C made new rules rather than changed existing ones, while Student B tended toward changing existing rules. This indicates that there are significant differences in approaches to parametric designing in relation to scripting. The third most dominantly occurring activity is the code ‘SE-Geometry’.

SE-Geometry (evaluating geometries in a 3D view) has the highest frequency in the protocols of Students A and B, and has the second highest frequency occurring in Student C’s protocols. This indicates that in parametric design environments, the design solution should be repeatedly examined using a 3D view to support conceptual design processes. Bilda and Demirkan [4] claim that it is time consuming to switch between the different visual representations that digital environments provide. In the conceptual design stage, sketching activities produce rich 2D and 3D imagery, while designers using CAD tools must switch to 3D view options. Furthermore, Students B and C’s design strategies employed the writing of scripts as the main method to generate their designs, and produced many instances of the SE-Rule code (evaluating existing rules) in the verification of their algorithms. Parametric design processes produce results and generate solutions that may be unexpected and/ or complex. Consequently, what is generated during the parametric design process must be evaluated in the 3D view as well as in the scripting view.

The summed coverage data in each 10 min interval was visualized using the normalized coverage value (normalized A = (A − mean)/SD) [9, 34] to facilitate the exploration of sequential patterns. Figure 2 shows the normalized coverage of geometric and algorithmic codes over time. It enables the representation of the cognitive design activity relating to geometric and algorithmic actions in sequence and to compare the changes over time.

A closer examination of Fig. 2 reveals that only Student B has activities coded as ‘RE (representation)-Geometry’ which is ‘to create or change geometries without an algorithm’. This is because Students A and C used algorithmic methods to achieve the manipulation of geometry. Student B has a different design strategy using both geometric and algorithmic methods to represent the design solution. This student used algorithmic methods for designing a skin pattern and a combination of geometric and algorithmic methods to model the cylindrical form of the building. In this protocol, the pattern of ‘SF-Geometry Subgoal (introducing new geometric ideas extended from a previous idea)’ overlaps that of ‘RE-Geometry’.

The graphs in Fig. 2 describe the normalized coverage of ‘SE-Geometry’ and reveal a relatively high value in the middle of each timeframe. Whilst ‘RE (representation)-Algorithm’ is a dominant feature in Table 3, the normalized value of these codes decreases over time. In the case of ‘SF-Algorithm Subgoal’, both Students B and C utilized a script editor and have the same pattern of the normalized value. Other activities based on algorithmic methods also have a similar pattern. To solve the design problem, Students B and C introduced new algorithms to represent design ideas which were extended from a previous idea at the earlier

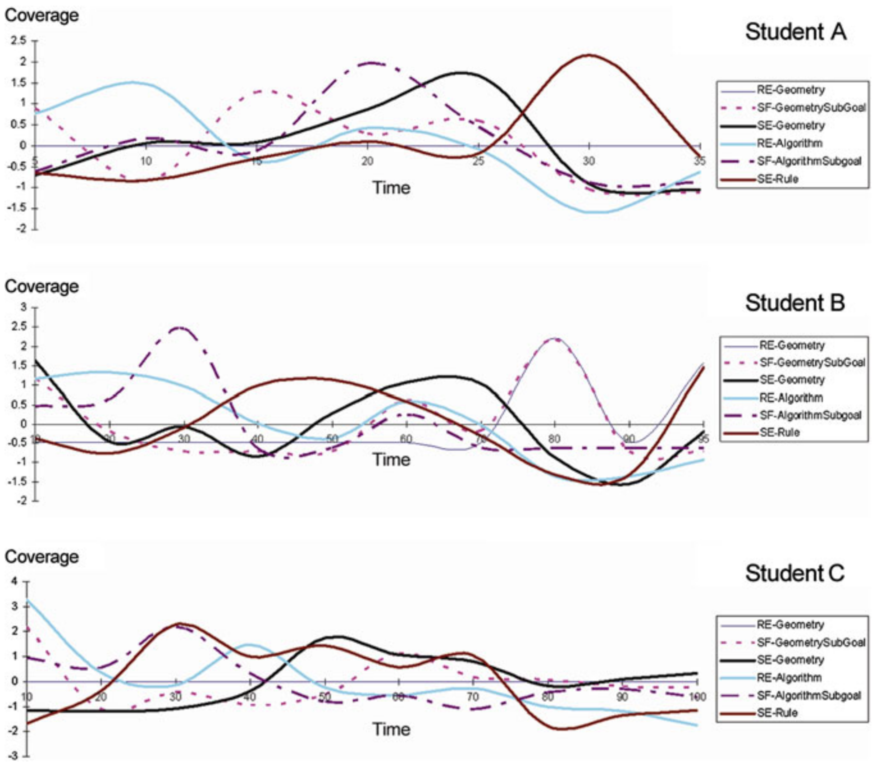


Fig. 2 The normalized coverage of geometric and algorithmic activities over time

stage of each timeframe. Activities coded as ‘RA-Rule’ in Student C’s protocol often occurred later in the design process.

Figure 3 is intended to present the normalized coverage of the different levels of creativity (i.e., Representation, Perception and Searching for a Solution) over time and the sequential changes of these three levels of creativity. While Fig. 2 illustrates the cognitive activity related to geometric and algorithmic protocols in detail, Fig. 3 deals with creativity levels to provide a better understanding of the creative process.

The coding scheme consists of five subcategories within the three main creativity levels: Representation, Perception, Finding Idea, Evaluation and Adopting Idea. The last three subcategories are at the level of ‘Searching for a Solution’ (refer to Table 1). Figure 3 illustrates the normalized value of the different levels for each design participant. In the case of the ‘Representation’ level, the value decreases over time and each student produces a similar pattern. Whilst the code ‘Perception’ shows a relatively small amount of coverage in Table 3, Fig. 3 reveals that there are different patterns within each timeframe. The ‘Finding Idea’ subcategory in all protocols reveals a similar pattern to the ‘Adopting Idea’

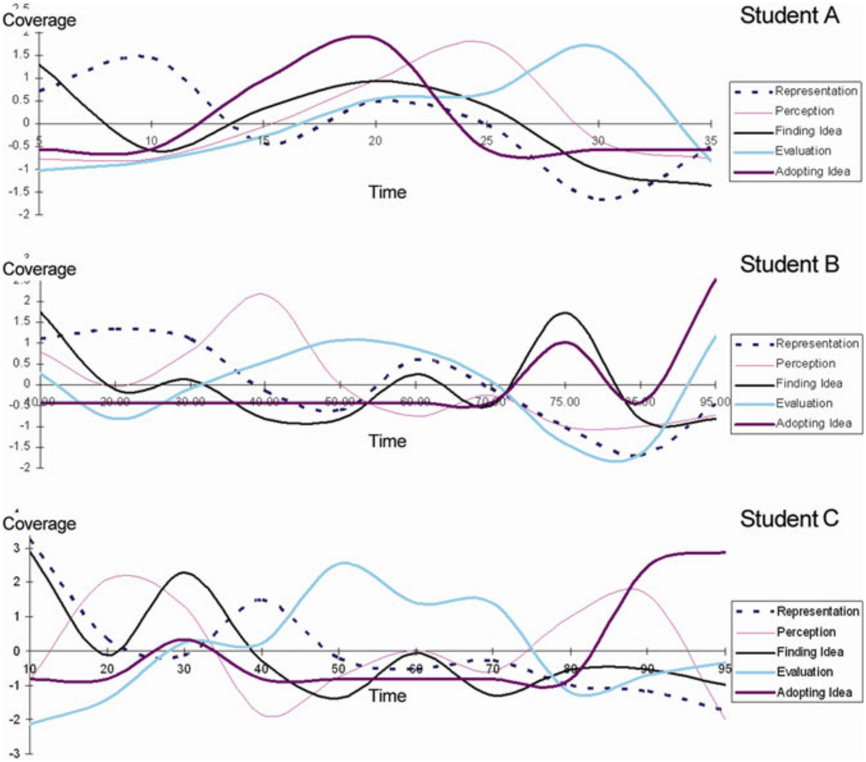


Fig. 3 The normalized coverage of the creativity levels over time

subcategory. We can assume that each designer's introduction of ideas follows a decision making activity. Both Students B and C who utilized a script editor produced a relatively high value of the 'Evaluation' subcategory in the middle of each timeframe, while Student A's increased over time.

These results reveal that the Design Process component of the framework is able to capture specific values of creative activities (see Table 3) and reveal a number of cognitive patterns (see Figs. 2, 3) in each student's parametric design process. These results enable further analysis of the data in comparison to the rating values of the expert panel assessment.

## Product-Based Evaluation of Creativity

This section presents the implementation of the second component in the framework, namely the expert panel assessment. Figure 4 shows an overview of each student's design process relative to solutions generated. It is expected that the coding results and the design behavioral patterns will correlate with the rating values of the expert panel assessment.

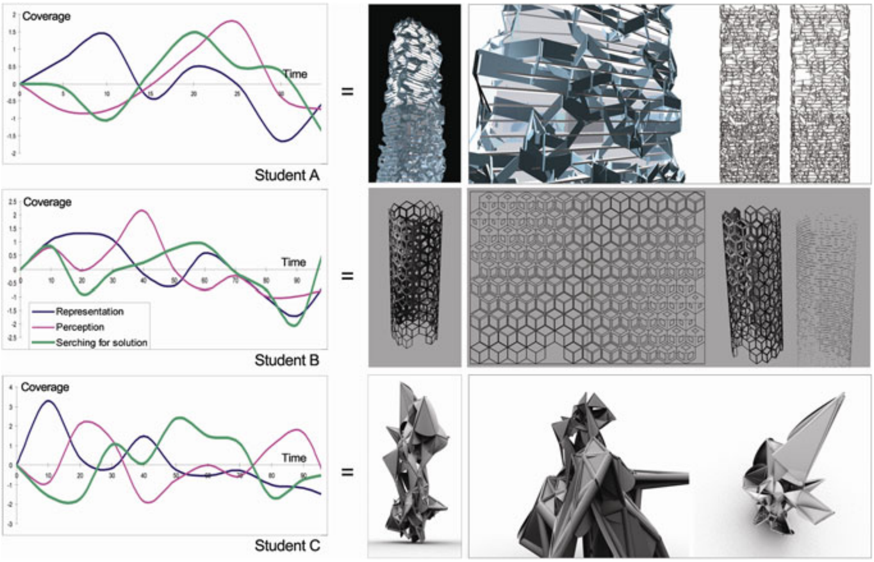


**Fig. 4** Students' design processes and products

## Assessment Experiment Procedure

To evaluate the creativity of the design product, a panel consisting of five expert judges provided assessment of the three parametric design models produced by the students. These judges fulfilled the following criteria:

- A tertiary degree in architecture design.
- Minimum 5 years of professional architectural design experience or 5 years of architectural design studio teaching experience.
- Familiarity with parametric design.

Parametric design solutions were presented as a collage of images on A4 size papers so that all design products were similarly scaled. The judges had access to all images in a de-identified form. They were aware that the designs had been produced using a parametric design environment and were asked to assess designs using three evaluation forms viz.

- Assessment 1 (A1): independent non-criteria based assessment of creativity,
- Assessment 2 (A2): comparative non-criteria based assessment of creativity, and
- Assessment 3 (A3): criterion-based assessment of creativity using—*(A3i)*: Novelty; *(A3ii)*: Usefulness, and *(A3iii)*: Complexity.

**Table 4** Assessment scores of the three parametric models

| | A1 (Indp. non-criteria based) | A2 (Comp. non-criteria based) | A3 (Combined criteria) | A3i (Novelty criterion) | A3ii (Usefulness criterion) | A3iii (Complexity criterion) |
|---|---|---|---|---|---|---|
| Student A | 5.4 | – | 4.8 | 4.6 | 5.2 | 4.6 |
| Student B | 4.8 | Lowest | 3.6 | 2.8 | 3.4 | 4.6 |
| Student C | 6.4 | Highest | 6.0 | 6.4 | 5.2 | 6.4 |
| Mean | 5.5 | – | 4.8 | 4.6 | 4.6 | 5.2 |
| SD | 0.8 | – | 1.2 | 1.8 | 1.0 | 1.0 |

Each assessment task used a seven-point Likert scale (where 1 is lowest and 7 is highest). In addition, in tasks *A1* and *A2*, judges were asked to list the assessment criteria used to evaluate the level of creativity, thereby providing insights into their rationale and design focus.

## Results of Expert Panel Assessment

The three students satisfied four of the five design requirements, provided in the brief. The final requirement (that the design solution be designated as a regional landmark) was included as a condition of the *A3* evaluation tasks, i.e., the level of novelty, usefulness, and complexity was conditional on the requirement that it reflects the characteristics of being a regional landmark. Table 4 shows the results of the judges' assessments of the three parametric design models.

The values shown in Table 4 are the average grades of the five judges' assessments. The final column shows the design solution's (i.e., student's) average grade received for each criterion. It also shows that the average values for the non-criteria based assessment are slightly higher than the average values in the criterion based assessment for each of the three criteria assessed.

The results consistently show that the level of creativity of Student C's model is assessed by the expert panel as being the highest across all evaluations, with the exception of task *A3ii*, where Student B's model has an equivalent score for the level of usefulness as a regional landmark. This is confirmed in the average of all assessment tasks (the final column).

For tasks *A1* and *A2* each panel member listed a range of criteria used to evaluate the level of creativity in these non-criteria based evaluations. Responses were categorized as shown in Table 5 and revealed a number of variations and similarities in the criteria used. The two most common responses, highlighted below, included 'Technical complexity' and 'Aesthetic attractiveness'.

**Table 5** Main patterns emerged in the comparison between the three students

| Patterns | Code | Student A | Student B | Student C |
|---|---|---|---|---|
| Process (Normalized values for each code) | | | | |
| $A > B > C$ | RA-Parameter | 1.02 | −0.05 | −0.98 |
| | SF-Initial Goal | 1.13 | −0.37 | −0.76 |
| $A > B \approx C$ | SF-Geometry SubGoal | 1.15 | −0.61 | −0.54 |
| | SE-Parameter | 1.15 | −0.58 | −0.58 |
| $A < B < C$ | RA-Rule | −0.92 | −0.14 | 1.06 |
| | R-Generation | −0.72 | −0.42 | 1.14 |
| | SE-Rule | −1.14 | 0.42 | 0.72 |
| $A < B > C$ | RA-ChangeRule | −0.67 | 1.15 | −0.48 |
| | RA-Reference | −0.64 | 1.15 | −0.51 |
| | PA-Algorithm | −0.80 | 1.12 | −0.32 |
| $A > B < C$ | PG-Geometry | 0.05 | −1.02 | 0.97 |
| | SF-Algorithm Subgoal | 0.93 | −1.06 | 0.13 |
| | SE-Geometry | 0.79 | −1.12 | 0.33 |
| | RA-Change Parameter | 1.08 | −0.89 | −0.19 |
| Product (Assessment scores) | | | | |
| $A > B < C$ and $A < C$ | A1-Indp. non-criteria based | 5.40 | 4.80 | 6.40 |
| | A3-Combined Criteria | 4.80 | 3.60 | 6.00 |
| | Mean | 4.92 | 3.84 | 6.08 |

## The Mapping of Process and Product Creativity Evaluations

Mapping the cognitive activities of the design process and the expert's rating of the design product was performed by correlating the following sets of data: coding results (see Table 3), non-criteria-based creativity and criteria-based creativity evaluation scores (see Table 4). Some visualized patterns in Figs. 2 and 3 were also used to inform the mapping process. In order to effectively deal with different types of data and values in Tables 3 and 4, the normalized values for each code and rating were calculated and presented relative to each student's cognitive activities. Various types of comparisons have been produced to explore the implication of the results. They are shown in Fig. 5 and Table 5.

Possible patterns emerging from the comparisons are listed in the left-hand-side column of Table 5. For example, '$A > B > C$' means that Student A's is higher than Student B's, and Student B's is higher than Student C's; '$A > B \approx C$' means that Student A's is the highest, and Student B's and Student C's are similarly lower; '$A < B < C$' means that Student A's is the lowest, and Student C's is the highest; '$A < B > C$' means that Student B's is the highest, and Student A's and Student C's are similarly lower; '$A > B < C$' means that Student B's is the lowest, and Student A's and Student C's value are similarly higher. The system allows us to easily relate the coding results of the design process to the rating results of the design product.

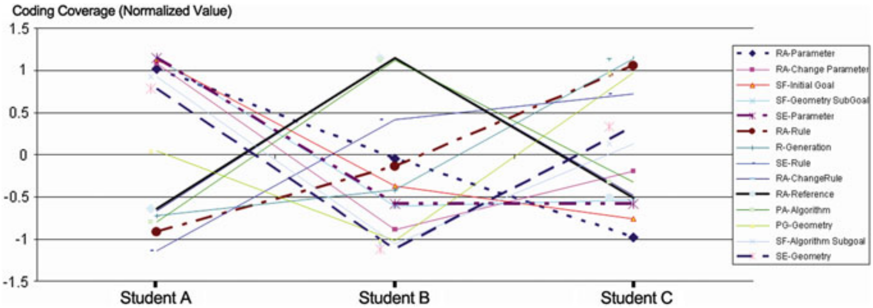


**Fig. 5** Comparison between the three students based on the normalized values of each code

Due to the limited sample size of the pilot study, we selected those patterns that are consistent with the expert panel's results for a surface discussion on the correlation between the parametric design process and product. The fifth process pattern (A > B < C) in Table 5 is consistent with the results, i.e. activities 'PG-Geometry', 'SF-Algorithm Subgoal', 'SE-Geometry', and 'RA-Change Parameter' are revealed as important aspects for supporting creativity in parametric design. More detailed implication analysis and other correlation analyses will be explored and generalized in future work.

## Discussion and Conclusion

This paper aims to explore and evaluate creativity in parametric design. The framework for this evaluation presented here shows that the underlying cognitive activities of the creative process can be described by the encoding schema and identified via protocol analysis. It further demonstrates that parametric design outputs can be formally evaluated using CAT. A mapping process through correlation analysis was used to identify the relationships between designers' cognitive patterns during the design process and the evaluation scores of the product. The preliminary evidence shows that the combined process-product approach is capable of revealing the conditions that potentially enhance creativity in parametric design. Overall, based on the correlation analysis (Table 5), this two-part framework performs well, revealing insights into the creative process and product in parametric design.

As a pilot study, the research reported here comes with limitations in the generalization of results. Whilst the authors acknowledge the small sample size, the results show some unique characteristics of creativity in parametric design. First of all, the paper highlights that the coverage of 'Searching for a Solution' is dominant and changing existing parameters or rules, as an algorithmic activity, is related to the potential production of creative outcomes. 'RA-Change Parameter' is an important activity for supporting creativity as shown in Table 5. In parametric design, rules are often regarded as graphical programming tools that carry constraints, while parameters are related to generating alternatives and reconstructing the design process. This implies that parametric design provide tools that can better support divergent thinking as well as providing reconstructing [10, 13] and regulating [8] processes, which enhance creativity.

'SE-Geometry' was also revealed as another dominant activity in parametric design. It can be assumed that more frequent evaluation activities potentially refine the design solution. The results of correlation analysis also support 'SE-Geometry' as potentially one of the most essential activities supporting creativity in parametric design. Furthermore, as shown in Figs. 2 and 3, different cognitive patterns can be indirectly related to creativity. For example, Students B and C who use a script editor have a similar pattern as indicated by the normalized values revealed in these figures. Student C who receives higher scores from judges shows a clear sequential patter of 'Finding Idea', 'Evaluation', and 'Adopting Idea'. These activities have previously been identified as common processes in a number of models of creative processes [26]. This shows that parametric design has the potential to support a more flexible and intuitive design process as well as a creative mode of designing. The criteria used by the panel to assess the product (see Table 5) still show the duality of both technical fitness and aesthetic attractiveness [35]. The reciprocal aspect of parametric design may be one of the most interesting features in terms of creativity.

In summary, the pilot study presents promising results demonstrating the effectiveness of the framework in evaluating parametric design creativity. Such a conceptual framework has not previously been available. The paper demonstrates and verifies the framework using the two-part pilot study. Using protocol analysis, the expert panel assessment and a correlation analysis by mapping the two, illustrate the framework's potential to investigate and contribute to our understanding of the nature of creativity in parametric design. However challenges remain; the key challenge lies with further testing of the framework using larger sample sizes in both the parametric design and the expert panel assessments. Future work therefore includes (1) verification of subscale criteria in expert panel assessment (see Table 2), which have not been tested exhaustively by this pilot study; (2) further investigation of the correlation results (see Table 5), which are not fully explored here due to the limited scope of this paper; and (3) a larger study to generalize results of both evaluation methods.

## References

1. Schumacher P (2009) Parametricism: a new global style for architecture and urban design. Architectural Des 79:14–23
2. Lee JH, Gu N, Sherratt S (2011) Developing a framework for evaluating creativity in parametric design. In: From principles to practice in architectural science-ANZAScA2011. The University of Sydney, Abstracts Book, pp 17

3. Casakin H, Kreitler S (2011) The cognitive profile of creativity in design. Thinking Skills Creativity 6:159–168
4. Bilda Z, Demirkan H (2003) An insight on designers' sketching activities in traditional versus digital media. Des Stud 24:27–50
5. Hasirci D, Demirkan H (2007) Understanding the effects of cognition in creative decision making: a creativity model for enhancing the design studio process. Creativity Res J 19:259–271
6. Chai K-H, Xiao X (2012) Understanding design research: a bibliometric analysis of design studies (1996–2010). Des Stud 33:24–43
7. Benami O, Jin Y (2002) Creative stimulation in conceptual design. In: Proceedings of the ASME design engineering technical conference, pp 251–263
8. Akin O, Moustapha H (2004) Strategic use of representation in architectural massing. Des Stud 25:31–50
9. Kim JE, Bouchard C, Omhover JF, Aoussat A (2010) Towards a model of how designers mentally categorise design information. CIRP J Manufact Sci Technol 3:218–226
10. Verstijnen IM, van Leeuwen C, Goldschmidt G, Hamel R, Hennessey JM (1998) Sketching and creative discovery. Des Stud 19:519–546
11. Ibrahim R, Rahimian FP (2010) Comparison of CAD and manual sketching tools for teaching architectural design. AutoCon 19:978–987
12. Liu Y-T (2000) Creativity or novelty? Des Stud 21:261–276
13. Madkour Y, Neumann O, Erhan H (2009) Programmatic formation: practical applications of parametric design. Int J Architectural Comput 7:587–604
14. Aish R, Woodbury R (2005) Multi-level interaction in parametric design. In: Butz A, Fisher B, Krüger A, Olivier P (eds) Smart Graphics. Springer, Berlin, p 924
15. Iordanova I, Tidafi T, Guité M, Paoli GD, Lachapelle J (2009) Parametric methods of exploration and creativity during architectural design: a case study in the design studio. In: Tidafi T, Dorta T (eds) Joining languages, cultures and visions: CAADFutures 2009, pp 423–439
16. Salim F, Burry J (2010) Software openness: evaluating parameters of parametric modeling tools to support creativity and multidisciplinary design integration. Computational science and its applications—ICCSA 2010. Springer, Berlin, pp 483–497
17. Chusllp P, Jin V (2004) Cognitive modeling of iteration in conceptual design. In: Proceedings of the ASME design engineering technical conference, pp 473–485
18. Rhodes M (1961) An analysis of creativity. Phi Delta Kappan 42:305–310
19. Simonton DK (2003) Scientific creativity as constrained stochastic behavior: the integration of product, person, and process perspectives. Psychol Bull 129:475–494
20. Amabile TM (1983) The social psychology of creativity: a componential conceptualization. J Pers Soc Psychol 45:357–376
21. Akin O (1986) Psychology of architectural design. Pion, London
22. Kim MJ, Maher ML (2008) The impact of tangible user interfaces on designers' spatial cognition. Hum–Comput Inter 23:101–137
23. Suwa M, Purcell T, Gero J (1998) Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions. Des Stud 19:455–483
24. Gero JS, Neill TM (1998) An approach to the analysis of design protocols. Des Stud 19:21–61
25. Coley F, Houseman O, Roy R (2007) An introduction to capturing and understanding the cognitive behaviour of design engineers. J Eng Des 18:311–325
26. Hayes JR (1989) Cognitive processes in Creativity. In: Glover JA, Ronning RR, Reynolds CR (eds) Handbook of creativity Perspectives on individual differences. Plenum Press, New York, pp 135–145
27. Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Des Stud 18:385–403
28. Amabile TM (1983) The social psychology of creativity. Springer, New York
29. Christiaans HHCM (2002) Creativity as a design criterion. Creativity Res J 14:41–54

30. Baer J, Mckool SS (2009) Assessing creativity using the consensual assessment technique. Creativity Res J: 1–13
31. Wallas G (1926) The art of thought. Harcourt Brace, New York
32. Hennessey B, Amabile T (1999) Consensual assessment. In: Runco M, Pritzker S (eds) Encyclopedia of creativity. Academic Press, San Diego, pp 346–359
33. Besemer SP, O'Quin K (1987) Creative product analysis: testing a model by developing a judging instrument. In: Isaksen SG (ed) Frontiers of creativity research: beyond the basics. Bearly Limited, Buffalo, pp 341–357
34. Bilda Z, Gero JS (2007) The impact of working memory limitations on the design process during conceptualization. Des Stud 28:343–367
35. Goldschmidt G (1992) Criteria for design evaluation: a process-oriented paradigm. In: Kalay YE (ed) Evaluating and predicting design performance. Wiley, New York, pp 67–79

# Interaction in Optimisation Problems: A Case Study in Truss Design

**Simon de Timary and Sean Hanna**

**Abstract** This paper is a preliminary study to explore the benefits of user interaction in topology optimisation by attempting to support two distinct claims: the first claim states that there exist similarly optimal, yet visually different designs based on the exact same parameters. The second one supports that accurately predicting the outcome can guide the program to faster convergence by skipping intermediary steps. For the purpose of this research a program based on Sigmund's *99-line MATLAB code for topology optimisation* was developed to implement real-time interaction. The programming language chosen was Java® for its flexibility and ease of scripting as well as its global efficiency. Both claims were tested through two distinct sets of experiments. The first one modified the designs by adding and/or removing material and proved the existence of similarly optimal yet different designs. The second one explored the use of pseudo-filters to simulate intuition and managed significant decreases in the amount of iterations necessary for convergence. This second experiment also produced slightly stiffer designs. Both experiments led to the conclusion that user interaction, when used responsibly, helps topology optimisation in generating creativity and in speeding the process.

## Introduction

The present work explores the benefits of human interaction in a deterministic process such as topology optimisation. The benefits of real-time human input are explored through two different statements. The first claim—responding the need

---

S. de Timary (✉) · S. Hanna
University College London, London, UK
e-mail: simon.de.timary@gmail.com

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_11,

for several design options[1]—states that, since the solution to the problem of minimising the objective function is a local minimum [1], there should exist similarly optimal, yet visually different designs based on the same parameters. The second claim—addressing the time constraints of design practices—supports that accurately predicting the final design can guide the program to faster convergence by skipping intermediary steps.

In order to assess both statements, two sets of experiments were conducted. The first one modified the designs by adding and/or removing material and the results showed the existence of similarly optimal yet different designs. The second one explored the use of pseudo-filters to simulate intuition and managed significant decreases in the amount of iterations necessary for convergence. Furthermore, 94 % of the designs produced in this second experiment had greater stiffness than those produced by standard methods.

Before getting any further, it is important to notice that the practical context of this research would involve an understanding of the degree to which the solutions resulting from user interaction compare to the theoretical optima (or as found by unaided gradient descent). As with any optimisation technique, there is potentially more research to be done to understand this with respect to various levels of complexity or specific problem domains. More specifically, two phenomena are relevant here: both the optimisation method itself and the nature of how a user's intuition might lead them to make changes based on their expertise. In this paper, we focus on the former, and so the rules adopted to stand in for real user interaction are used so that the experiments can be controlled. Further work on the nature of real human learning processes and intuition about structures would be a useful further addition.

For the purposes of this research, a topology optimisation tool based on Sigmund's *99-line MATLAB code for topology optimisation* [2] was implemented to enable real-time user-interaction. The program is a classical topology optimisation in the sense of material distribution [3]. The design is iteratively adapted to respond the displacements provided by a finite elements analysis until convergence is reached. This process is thus deterministic in that it will always produce the same design when run with the same parameters. This program is still in its developing phase and should, in its present state, mainly be considered as an education tool.

The first part of this paper will be dedicated to a brief explanation of the finite element method, a short review of the theoretical background constructed over the last 20 years by the leading authors in topology optimisation's field and a presentation of the used algorithm's structure. The second part will describe more thoroughly the research question and the two claims developed to answer it. The fourth part will describe the methods used to generate and analyse the results. These will then be presented and discussed. Finally, will be introduced the limitations of this study along with suggestions for future researches and the conclusion.

---

[1] In order to be able to make a choice based on aesthetical reasons.

# Theoretical Background

## Finite Element Method

The finite element method is a calculation technique based on discretising a structure or a continuous domain in a finite number of simple geometrical shapes which are called the finite elements [4]. This system greatly simplifies calculations since all of these elements can be calculated individually before the whole system is put together again. The stresses and strains are calculated locally at the nodes from the potential energy equation KU = F [5] where K is the stiffness matrix, F the force vector and U—the displacements vector—the unknown. This local equation can be extended to the whole system and thus enables the calculation of the displacements of every node in the system. Since the displacements are exact at the nodes [4], it implies that a larger amount of elements (a finer mesh)—therefore a larger amount of nodes—in the system, enables better accuracy of the calculated stresses and strains [4].

This calculation technique is perfectly well suited for the topology optimisation system provided by Sigmund [2], which is based on modifying the elements' density according to their nodal displacements [6]. Indeed, the density can be modified for each element individually and, as a result, the element stiffness matrix becomes the product of the original matrix by the density: $K_e^{new} = \rho_e K_e^{orig}$.

## Topology Optimisation

Amongst several existing types of shape optimisation problems, the present work uses that of shape design of optimal material distribution [3]. As Bendsøe [7] declared: "Shape optimization in a general setting requires the determination of the optimal spatial material distribution for given loads and boundary conditions". The problem thus is, for each point in the domain Ω, whether there is material or not (amongst others: [3, 7]). If this problem is solved using finite elements, the domain becomes that of the elements and each element is either solid or void [7, 8].

Optimal shape design's aim is to minimise the objective function:

$$\min L(u),$$

as described by many authors (e.g.: [9] ), where $u$ is the displacement vector. In the present case, the objective function is the compliance, which is defined as "the work done by the external loads" [1]:

$$L(u) = \int_{\Omega} f_i u_i dx + \int_{\Gamma} g_i u_i d\Gamma.$$

Furthermore, since the compliance is equal to "twice the additive inverse of the potential energy" ([10]: 206) and to the inverse of the stiffness, minimising the compliance can be considered as maximising the stiffness [1]. One can thus express the objective function through the energy bilinear form [3]:

$$a_E(u, v) = L(v).$$

$$\text{s.t: } a_E(u, v) = \int_{\Omega} E_{ijkl} \varepsilon_{kl}(u) \varepsilon_{ij}(v) dx,$$

where the elasticity tensor $E_{ijkl}$ belongs to the set of admissible elasticity tensors $U_{ad}$. In these terms, the goal of the optimal design is to find the optimal choice of elasticity tensor in $U_{ad}$ [3] or if finite elements are used to solve the equations, the optimal stiffness matrix $K$.

## Program Presentation

The code used in this research was assembled from the following two codes, both available online. The first one is a translation in Java® of Sigmund's *99-line MATLAB code for topology optimisation* [2]. The second code is Nikishkov's *Programming the finite elements method in Java®* [5]. While Sigmund's code is self-sufficient in MATLAB, the matrix operations are not taken care of in Java and Nikishkov's LDU (lower, diagonal, upper) solver was implemented.

The optimisation process is run on a rectangular beam (domain) which is then discretised in the desired amount of squared quadratic 4-node elements (Q4). These elements all have the same material properties (the same element stiffness matrix) and are associated with a density factor when the global stiffness matrix is assembled. At the first run of the algorithm each element in the global stiffness matrix has its density initialised to the target volume (e.g. 0.5). With this particular setup is then run a first finite elements analysis which returns the nodal displacements. The elements' density is then adapted to match the displacements before filtering the sensitivities of the new density in order to make the design mesh-independent and checkerboard-free. Finally, if the maximal difference between the previous and current stiffness matrix is over a certain threshold (e.g. 0.001), the process starts again from the finite element analysis (see Fig. 1).

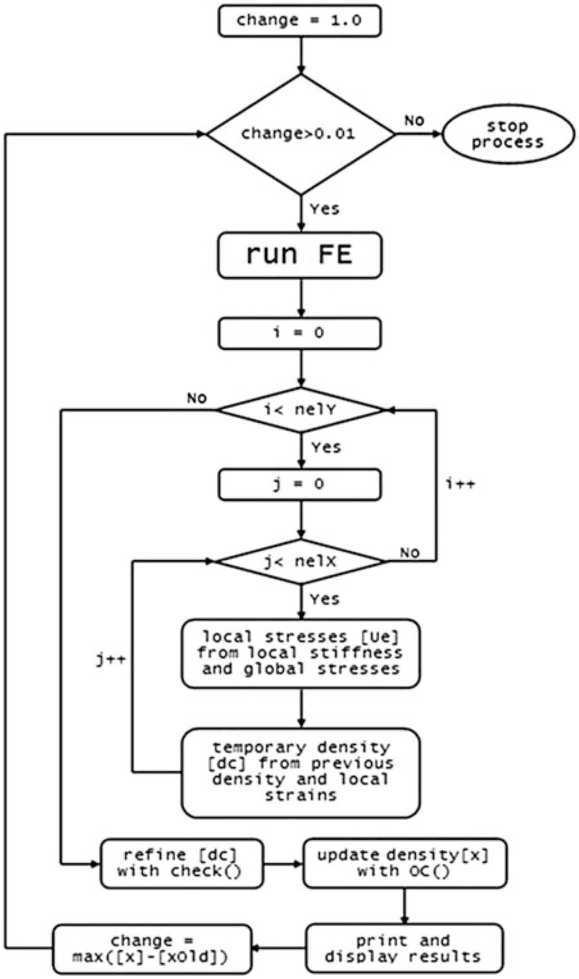


**Fig. 1** Diagram of the optimisation process of a domain of nelX × nelY elements [16]

## Research Question

Topology optimisation has been extensively studied in the past 25 years and for more information on the mathematical concepts behind it, one can refer to the existing literature in the field. The aim of this paper is thus not to develop new techniques in topology optimisation but rather to study the opportunities offered by new computational resources in user interaction. To do so, the impact of user interaction as a means for design and efficiency was explored. The type of topology optimisation presented here relies on a finite element analysis coupled with a hill climbing approach, neither of which involves randomness. This process therefore leads consistently to the same results for the same parameters. This phenomenon might be unwanted when the program is used in a design practice that is likely to look for several different options. The time factor is also a crucial aspect and constraint of designing a project. In regard to these issues, the present research will try to assess whether user interaction can reduce computational time as well as generate various designs responding the same conditions. In order to explore the interrogations of this research question, two claims were developed.

- **Claim 1**

The first claim is related to the issue that topology optimisation is a deterministic process. Indeed, practices are likely to wish for several different designs produced on the same parameters and since the problem of minimising the compliance only gives local minima [1], one might wonder whether it would be possible to produce different designs that are similarly optimal. The claim is as follows.

C1: *There exist visually different designs, solutions to the same design problem, which have comparable compliances to the optimal shape produced by the program on its own*

The way to supporting this claim is to try and produce different designs on the same boundary conditions and then compare their objective function (the compliance) with the originals.

- **Claim 2**

The second claim emerges from the fact that topology optimisation is an extremely time-consuming process. One might thus wonder whether these processes could be sped up by a user. The second assumption is as follows.

C2: A*ccurately predicting the outcome guides the program to early convergence, skipping a large number of steps, thus reducing computational time*

This claim will be assessed by filters mimicking intuition in such a simple way that real intuition should provide better results.

## Methods

In order to assess the validity of both claims, two series of tests were conducted. Automated processes were chosen over real interactions for comparison and accuracy purposes as well as for resource constraints.[2] Five base cases with different loads, boundary conditions, proportions and mesh refinement were created. These cases were named: *4-corners*, *cantilever*, *MBB*, *Michell-1* and *Michell-2*. They can be seen in their final form in Figs. 2, 3, 4, 5, 6.[3]

---

[2] See the parts on future improvements and limitations for more information on resource constraints.

[3] Triangles represent hinges; bars represent boundary conditions for a whole side and the thin arrows under those bars show the direction in which displacements are blocked; thick arrows represent a downwards unitary force.

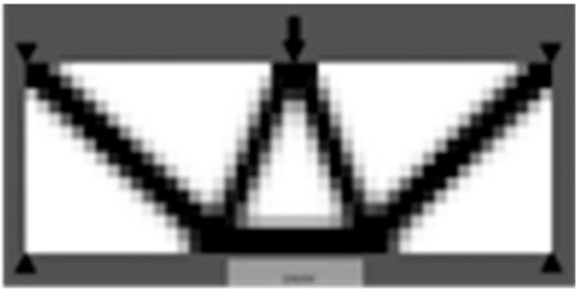


**Fig. 2** 4-corner

**Fig. 3** Cantilever

**Fig. 4** MBB-beam

**Fig. 5** Mitchell-1


**Fig. 6** Mitchell-2


**Fig. 7** Pattern 1 (Test 1, Exp.3)


**Fig. 8** Pattern 2 (Test 2, Exp.3)

The first wave of tests was trying to assess the validity of *Claim 1* and their aim was to produce designs as different as possible from the originals. In order to do so, series of ‘blind’ tests were realised. These were gathered into three different experiments. While each experiment followed the same patterns, the first one solely added material, the second one strictly removed material and the third one was a combination of both.[4] Six patterns were drawn once onto the designs at different times with a geometric progression in the loop where the change was made (1, 2, 4, 8, 16, 32 and 64). These patterns can be observed in Figs. 7, 8, 9, 10, 11, 12.

These three experiments comprised a total of 528 tests amongst which 231 visually different designs (44 % of the sample) were selected arbitrarily by the researcher. However, one must note that, even though 231 tests led to different designs, there were not 231 different designs produced. Most of them were similar

---

[4] It might useful to note that the intervention of a user mid-way through the optimisation process is a special case of changing the initial start point for optimisation.

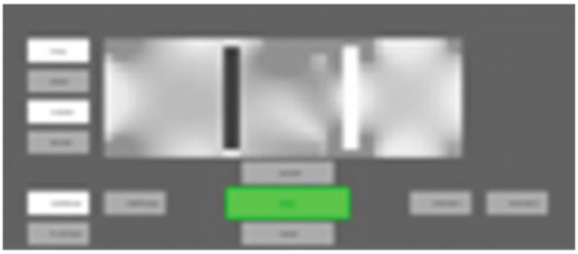


**Fig. 9** Pattern 3 (Test 3, Exp.3)

**Fig. 10** Pattern 4 (Test 4, Exp.3)

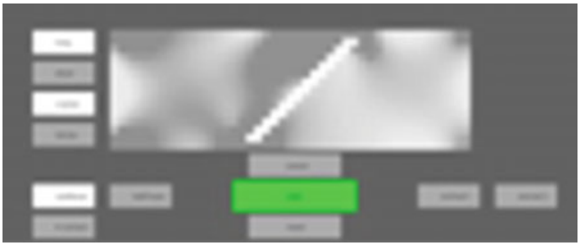


**Fig. 11** Pattern 5 (Test 6, Exp.3)

**Fig. 12** Pattern 6 (Test 4, Exp.3)

to each other and only 28 genuinely different families of designs were produced. The comparisons in the results analysis section were made on one example per family. These can be observed in the appendix.

The second series of tests simulated intuition through the use of two different pseudo-filters[5] both based on the assumption that a user trying to help the program would remove the excess material where it is obvious that there will not be any at the end of the run and add up material on the structural members. This type of filter could be considered as an explicit penalty on the densities along with SIMP [8, 9]. Thirty-two tests were conducted for the second hypothesis (16 per filter). The first filter was a simple threshold that removed the weakest elements (under 0.2 of density) and reinforced the strongest ones (over 0.8 of density). The second filter was looping through the density matrix and retrieved the weakest and strongest elements and changed them to the lower limit $\underline{\rho}$ (0.001 to avoid a singular stiffness matrix [11] ) and to the upper limit $\bar{\rho}$. This filter was launched only after the fourth loop when enough elements had been able to evolve towards extreme values. The filter started by selecting the 10 % most extreme values on either side and this amount increased by 1 % per loop until it reached the lowest value between 35 % and the target volume.

Every test of every experiment returned the last density matrix, and a record of the evolution of compliance and change through the loops. The change in density (deviation) from the original design was calculated by the following formula:

$$\frac{\sum_{e=1}^{N}\left|x_e^{orig} - x_e^{modified}\right| \times 100}{N}$$

where $x_e^{orig}$ and $x_e^{modified}$ are the density matrix of the original and modified designs respectively and $N$ is the number of elements. Similarly, the change in compliance and iterations had to be calculated the following way:

$$\frac{\left(X^{orig} - X^{modified}\right)}{X^{orig}} \times 100$$

where X is the value being compared.

## Results Analysis and Discussion

**Claim 1** *There exist visually different designs, solutions to the same design problem, which have comparable compliances to the optimal shape produced by the program on its own.*

The first statement tries to assess whether structures produced from the same boundary conditions and contrived to be different could have a similar compliance. In order to do so, the first part of the experiments was trying to generate visually

---

[5] These pseudo-filters have a role similar to actual filters but work with threshold values rather than neighbouring cells.

different solutions by applying a series of blind tests to the structures at different time intervals. Out of 528 tests, 231 led to visually different designs ($\sim$44 %). The first and second experiments were both made of 198 tests and led respectively to 69 ($\sim$35 %) and 92 different designs ($\sim$46 %). The third experiment comprised 132 tests and led to 70 significantly different results ($\sim$53 %). One must note though that the last two experiments only led to 168 (85 %) and 117 (89 %) designs. The remaining tests crashed and only produced a fully black domain. This shows that subtraction (and a combination of both addition and subtraction) of material is more efficient in achieving change than simple addition. This can be explained by the fact that removing material allows for better redistribution of material. The volume must stay constant at the end of the process and this means that the removed material must be redistributed in the design and, since removed material cannot sustain any stresses, these need to be displaced elsewhere on the design. Material will not be redistributed where it was removed but where the stresses and displacements have shifted. Yet, it is also a more destructive process. Indeed, when the alterations are made late in the process and slash through the structural elements, this leads to singularities in the stiffness matrix and the result is a black beam. On the other hand, when simple addition is performed, unless there is significant room for change (blurry design, homogeneous material distribution) the changes cannot be taken into consideration since the filtering of sensitivities takes it as a numerical instability (i.e. checkerboard effect[6]). Moreover, on the contrary of subtractive modifications, material must be removed and usually the added material is removed immediately, as the displacements have not shifted and the original structure is still optimal in regard to those.

Most of the 28 different designs achieved quite a significant change in material distribution (over 90 % of the sample achieve more than 10 % of change). Furthermore, as can be seen on Fig. 13, 75 % of the sample achieved a compliance close to the original compliance (less than 5 % of difference or stiffer solutions). An obvious result was that minimal change in the structure leads to minimal change in the compliance (all tests under 10 % of change led to a change in compliance of less than 1 %). Another result is that after a certain point of change, the resulting compliance is not even remotely close to optimality (this may be observed on Fig. 7, zone 2). The limit seems to lie at about 25 % of change (red vertical line) after which the increase in compliance goes over 20 % almost systematically. This leads to the conclusion that while it is possible to achieve significant changes in topology and still keep the same or better structural quality, if the design is too different from that provided by the program alone, the structure is no longer optimal. However, finding the change limit after which the design is no longer optimal would require further researches since only the Michell topologies showed such an increase in compliance.

---

[6] Defined as "areas where the density jumps from 0 to 1 between neighbouring elements" (Pedersen et al. 2006: 1), the checkerboard patterns are a common numerical instability that occurs particularly when using Q4 elements [12].

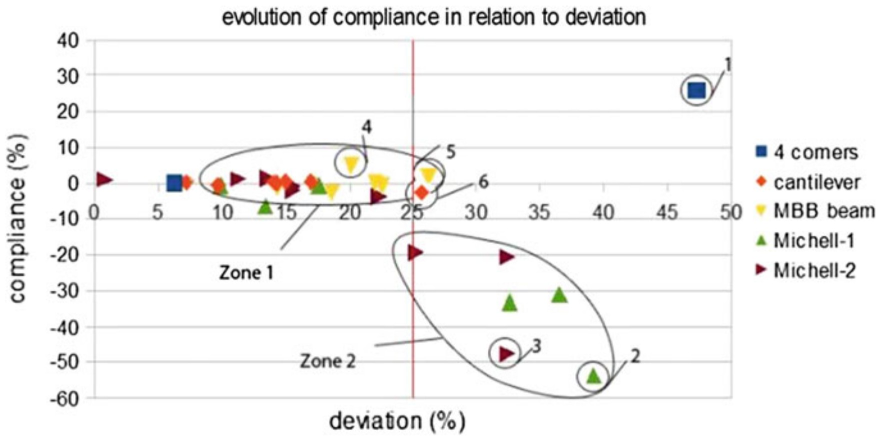


**Fig. 13** Compliance in relation to deviation

From these results, claim 1 may be confirmed since it is clearly possible to obtain significantly different topologies (>10 % change) that have comparable or better stiffness. All results in zone 1 (see Fig. 13) qualify for these criteria, since they produced over 10 % change and have less than 5 % increase in compliance. Out of the 28 families, 17 (61 %) are in zone 1 and out of the 11 remaining, three produced marginally better results without accomplishing significant change and one had a spectacular percentage of change and stiffness improvement (circled 1 on Fig. 13[7]). There is also a limit to change that the program can accomplish and, if the program manages to cross that limit, it leads to non-optimal designs. Crossing this limit should thus be avoided. No specific method was developed to avoid crossing this limit; however, good sense (i.e. not isolating the structure from its supports) is usually enough to prevent high rises in the compliance. One might conclude that user interaction can provide valuable help for topology optimisation but is at the user's discretion and not all results provided are optimal.

**Claim 2** *Accurately predicting the outcome guides the program to early convergence, skipping a large number of steps, thus reducing computational time.*

The second statement claims that user interaction would speed up the optimisation process by removing intermediary steps. The results analysis confirmed this statement. Indeed, using pseudo-filters to mimic intuition produced results in a much shorter amount of iterations. However, what is suggested in this research is not that user interaction is the only way to reduce the amount of iterations and therefore improve the speed factor of an optimisation process. Indeed, it is important to first optimise the speed factor of the programme computationally. As

---

[7] This case is the 4-corners case for which the solution provided by the program is far from the optimal one. It is only a marginal result and should thus be disregarded.

an example of such speed improvement, Andreassen et al. reworked Sigmund's 99 line MATLAB code to an 88 line code that showed a speed improvement factor of 100 and enabled the calculations of highly dense meshes [13]. Tovar et al. also generated a new technique relying on hybrid cellular automata that showed a considerable reduction in the amount of iterations [14, 15]. Only then, when using the most efficient structural optimisation technique available, one could add user interaction to it and enable for the designers' intuition to guide the programme to faster convergence, therefore further reducing the amount of iterations. The first simulation of intuition, which was a poor simulation in regard to obtaining smooth looking results, provided astounding results with decreases in the number of iterations of up to 54 % (Michell 2, dense). As a result, convergence is indeed sped up by modifying the densities already clearly solid or void. Intermediary steps are thus removed by the user. These tests also resulted in lower compliance. This can mostly be explained by the fact that the lowest densities (that have very little contribution to the stiffness by their low value and the penalisation scheme or SIMP [12]) are set to $\underline{\rho}$, which allows for more material to be added to the structural parts, thus enhancing the stiffness. The final design is closer to a 0-1 design while keeping the same volume and is thus logically stiffer. The use of simple pseudo-filters to simulate a simplified intuition based on the idea that users will try to get rid of intermediate densities by erasing the weakest elements and strengthening the strongest proved successful. Both filters showed a sharp decrease in the number of iterations as well as a relative decrease in the compliance. Several tests reached savings in iterations of over 60 %. All tests were very receptive to the filters apart from the first Michell topology with a coarse mesh which was already at a very low amount of iterations. This process turned out to be very efficient by itself and could not be improved by these filters. All the others were very receptive and out of the 32 conducted experiments, only 5 of them (amongst which the two Michell-1 coarse experiments) did not lead to a decrease in the amount of iterations (84 % of the experiments conducted to a positive result). This can easily be observed in Fig. 14. The mean decrease in the number of iterations was 28.1 %.

Moreover, along with a decrease in the number of iterations, the filters led to a decrease in the compliance (a higher stiffness). While the decrease in compliance was not as impressive as that of the iterations, it was still noticeable. Indeed, the results reached a decrease of up to 15 % with the first Michell truss in the second experiment and even reached 23 % in the long and dense 4-corners structure.[8] Only two examples achieved a higher compliance (an increase of less than 3 %) out of the 32 experiments (94 % positive results). These results can be observed on Fig. 15. The mean decrease in compliance was 5 %.

In order to verify the obtained results, the material distribution's deviation between the original design and the filtered ones was analysed. Some of these results were rather different. The 4-corners example was quoted in previous

---

[8] This result corresponds to the one mentioned in the previous footnote.

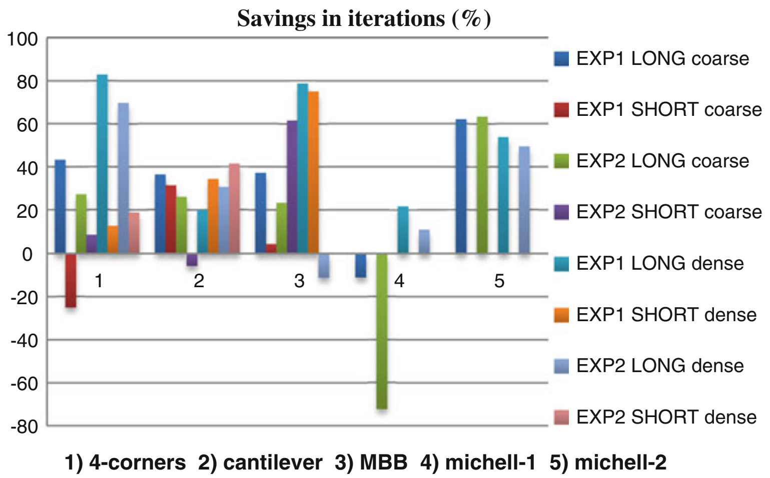


**Fig. 14** Savings in iterations

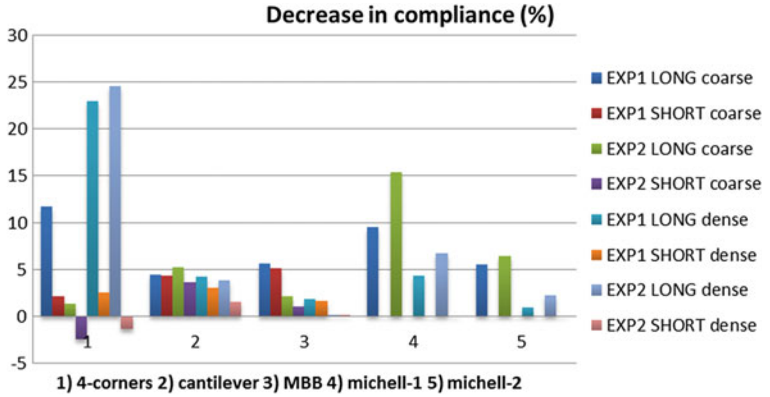


**Fig. 15** Decrease in compliance

developments. The cantilever beam,[9] long and dense of the second experience had a deviation of 10 % and was visually very different (see Figs. 16, 17). As can be observed, the modified cantilever beam would be easier to predict and seems more natural. However, the lines are not as smooth and this phenomenon could be observed on all filtered structures (these filters somehow seemed to counteract the blurring effect of the filtering of sensitivities). This effect might be responsible for the decrease in compliance.

---

[9] The reason the original design is unsymmetrical is the slight upward shift in the concentrated load. Indeed, it is not applied on the middle node but on the one above it.

**Fig. 16** Original cantilever (densemesh)

**Fig. 17** Resulting cantilever, exp. 2

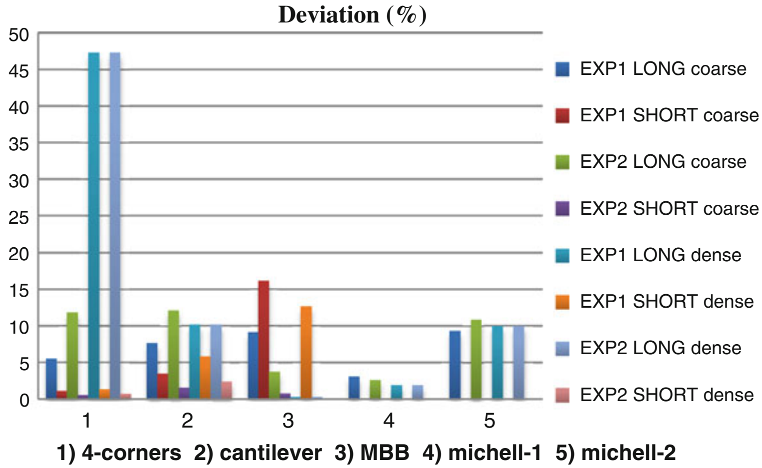


**Fig. 18** Percentage of change

The 4-corners, cantilever and Michell-2 structures all had a deviation of ±10 %. However, the results were extremely close for the short examples both visually and quantitatively. These cases resulted in smaller decrease in the compliance and accounted for some of the negative results both in compliance and in iterations. As can be observed on Fig. 18, the deviation generally stayed under

10 %, one can thus conclude that these structures stayed globally the same with a better compliance and a lesser amount of iterations. As for the structures that deviated by more than 10 %, they were all slightly stiffer and thus contribute to demonstrating *claim 1* as well. These results clearly indicate that predicting the outcome (even poorly such as was presented here), helps the program drastically.

To conclude the part on the results analysis, one can see that both claims seem to hold and that real-time user interaction may be considered useful in a topology optimisation process when it is used to help or find other designs. However, this process can lead to non-optimal designs when the user is not careful and can also vastly increase the number of steps.

Designers should hence always be cautious and make sure the modified topologies provided are optimal. However, a good knowledge of the program truly guides intuition both in generating valid different designs and in driving the process to faster convergence. Further research would enable to determine the threshold to which change leads to non-optimal designs, inducing “rules of thumb” and threshold values for the compliance. Furthermore, despite the fact that one might need to run the program on its own in order to assess the validity of their own solutions, most of the time the program’s use will take place in a large design process and will usually be the very first run before alterations are made.

Finally, one should add that real interaction would have further reduced the amount of iterations. Indeed, using the filters at every loop often prevented the programme from reaching convergence.

## Limitations

The presented results, discussions and interpretations have to be considered under some limitations. Firstly, in the experiments linked to the first statement, not only was the selection of visually different designs made arbitrarily, but so was the selection of the different families. This inevitably resulted in information losses. Due to time and hardware constraints, only a limited amount of tests were conducted and the mesh refinement was limited to a rather coarse mesh. A more extensive range of boundary conditions could have revealed different results and a more refined mesh would have enabled better interaction and a larger deviation of the modified designs from the untouched ones. Indeed, coarse meshes tend to be too stable.

Further improvements could be made to the code in order to make it more user-friendly: incorporate it as a plug-in into a CAD environment in which the designer could simply draw the boundary conditions, select the number of elements in one direction and then generate the sequence and interact with it; play it on a touch-screen, etc. Other improvements should be made in order to enable more possibilities of the code (i.e. enable 3D, variable boundary conditions, variable forces, take into account structural instabilities such as torsion or buckling, etc.).

Using designers and structural engineers unfamiliar with the design in the process of testing would have broadened the range of results and made possible input of real intuition. For instance, the rules used in these experiments were much too simple to accurately simulate user interaction and if it really were possible to simulate user interaction, then user interaction would not improve what can be generated by a computer. The real interest lies in genuine human input. However, the aim of this paper was to determine whether user interaction could improve the process of structural optimisation, therefore human learning processes were not taken into account. Future research should thus include tests with designers and engineers, and assessment of genuine intuition in order to demonstrate that this would lead to even better results. Tests should also be conducted to determine the limit of change at which the modified design becomes non-optimal.

Finally, this paper and the supporting tool should only be considered as preliminary work in interaction for structural optimisation. The program is intended as a starting point in the reflexion on structural design and should be followed by further design and more extended structural calculations taking into account material properties, uncertainty, connexions and national building codes. Moreover, the programme should be optimised in regard to speed and robustness issues. Only then could it be considered as a real design tool. Until then, it should mostly be considered as a teaching tool intended for beginning designers to grasp the concepts behind structural optimisation and develop their intuition in the field. This research is not in itself a revolution in design but merely suggests leads as to how structural optimisation should evolve in the future.

## Conclusion

The present research tried to assess whether the introduction of human interaction could help make topology optimisation a better design tool for structural designers. Two claims were developed to investigate this and were both verified in the present work. It was demonstrated that designs can be bent towards different solutions and still be optimal. This result implies that designers should be able to produce a wide range of optimal designs that fulfil their requirements so they can choose the “best one” on aesthetic criteria. This study also showed that predicting the final design significantly reduces computational times. This would be useful when trying to meet the time constraints of a design process.

However, if this program is not used properly, it can lead to designs far from optimal and is also liable to crash without returning any design at all. Designers should thus be conscientious when interacting with a structural optimisation software and pay attention not to break the structure without providing it with an alternative. They should also ensure that the structure is indeed optimal.

Hence, a robust and fully functional version of this program would be a useful tool to incorporate at the beginning of any design process. It provides a wide range of optimal designs and leads the designers in several different directions. It can

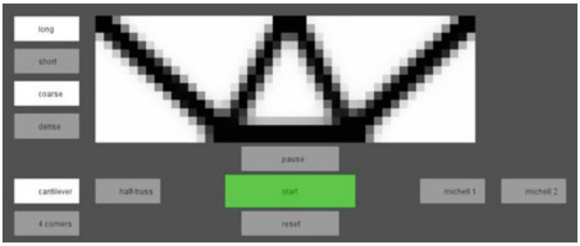


**Fig. 19** 4-corners, family 1, deviation 6.2 %, compliance +0.13 %

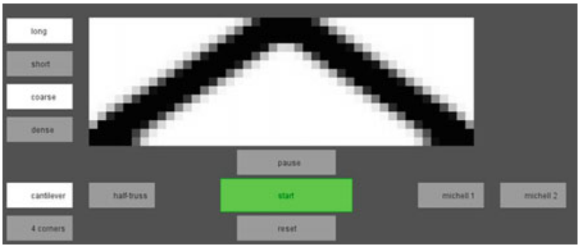


**Fig. 20** 4-corners, fam. 2, dev. 47.3 %, comp. −25.8 %

**Fig. 21** Cantilever, fam. 1, dev. 7.1 %, comp. −0.12 %

also be used as an education tool to develop starting designers' intuition in topology optimisation as well as in general structural design.

Finally, the solving of complex topology optimisation problems tends to exclude radically any human input and the process usually ends up by an unsupervised computer calculating a solution overnight. If human responsibilities are limited to feeding a program with boundary conditions before even that program is launched, then to what extent can topology optimisation be considered a part of design? It is hoped that this research and the program that supports it will contribute to a greater integration of intuition and designer creativity in the use of topology optimisation.

## Appendix: Design Families

See Figs. 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 and 30.

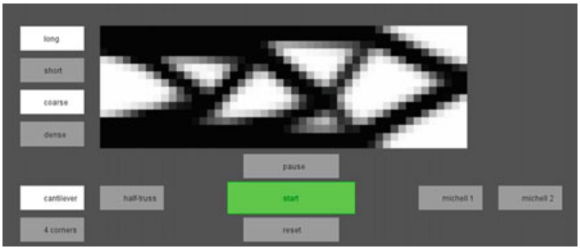


**Fig. 22** Cantilever, fam. 2, dev. 15 %, comp. −0.25 %

**Fig. 23** Cantilever, fam. 3, dev. 14 %, comp. −0.11 %

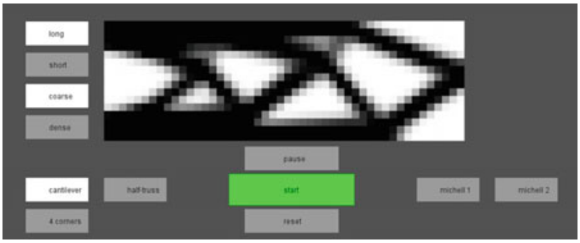


**Fig. 24** Cantilever, fam. 6, dev. 25.2 %, comp. +2.9 %

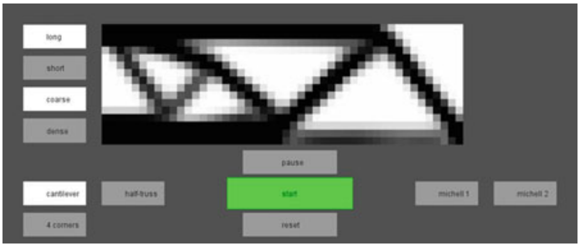


**Fig. 25** MBB, fam. 1, dev. 22 %, comp. +0.05 %

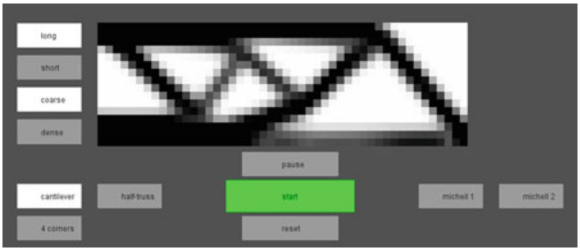


**Fig. 26** MBB, fam. 2, dev. 14.3 %, comp. +1.0 %

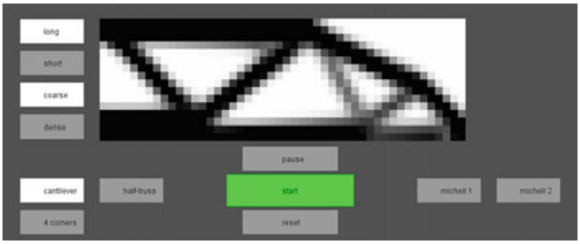


**Fig. 27** MBB, fam. 5, dev. 26.2 %, comp. −1.73 %

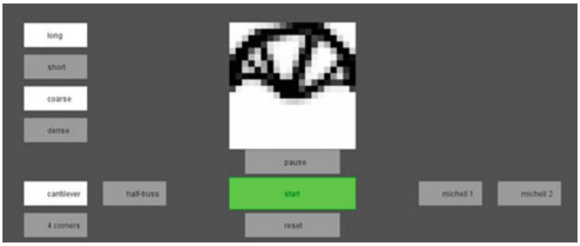


**Fig. 28** Michell-1, fam. 5, dev. 39.1 %, comp. +53.8 %

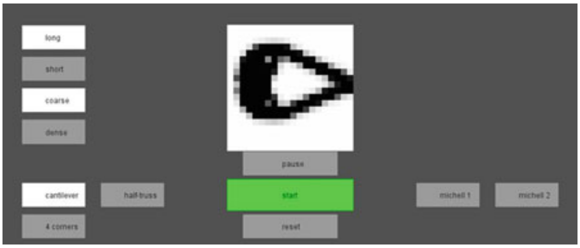


**Fig. 29** Michell-2, fam. 4, dev. 13.6 %, comp. −1.3 %

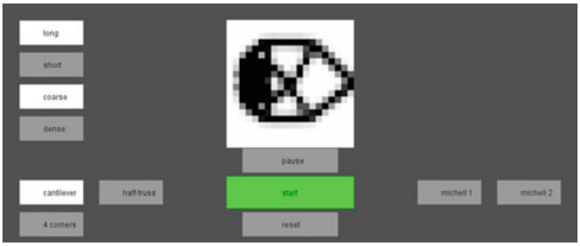


**Fig. 30** Michell-2, fam. 5, dev. 15.5 %, comp. +2.1 %

## References

1. Reiss R (1976) Optimal compliance criterion for axisymmetric solid plates. Int J Solids Struct 12:319–329
2. Sigmund O (2001) A 99 line topology optimization code written in MATLAB. Struct Multi Optim 21:120–127

3. Bendsoe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71:197–224
4. Warzee G (2010) Méthode des éléments finis (structures), 7th edn. Presses Universitaires de Bruxelles, Brussels
5. Nikishkov G (2010) Programming finite elements in Java. Springer, London
6. Bendsoe MP, Sigmund O (2003) Topology optimization. Theory, methods and applications. Springer, Berlin
7. Bendsoe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1:193–202
8. Rozvany GIN (2001) Aims, scope, methods, history and unified terminology of computer-aided topology optimization in structural mechanics. Struct Multi Optim 21:90–108
9. Liszka T, Ostoja-Starzewski M (2003) Effects of microscale material randomness on the attainment of optimal structural shapes. Struct Multi Optim 25:1–10
10. Chandrasekhar SJ, Haberb RB (1996) Stability of finite element models for distributed-parameter optimization and topology design. Comput Methods Appl Mech Eng 130:203–226
11. Bruns TE (2005) A re-evaluation of the SIMP method with filtering and an alternative formulation for solid-void topology optimization. Struct Multi Optim 30:428–436
12. Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multi Optim 33:401–424
13. Andreassen E, Clausen A, Sschevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in MATLAB using 88 lines of code. Struct Multi Optim 43:1–16
14. Tovar A, Wilson IQ, Patel NM, Renaud JE (2005) Hybrid cellular automata with local control rules: a new approach to topology optimization inspired by bone functional adaptation. In: 6th World Congress of structural and multidisciplinary optimization, 30 May–03 June 2005, Rio de Janeiro, Brazil, pp 1–11
15. Tovar A, Patel NM, Niebur GL, Sen M, Renaud JE (2006) Topology optimization using a hybrid cellular automaton method with local control rules. J Mech Des 128:1205–1216
16. dE Timary S (2011), Exploring interaction as an optimisation tool, research dissertation, adaptive architecture & computation. University College London

# Part IV
# Design Cognition—2

# The Role of Design Team Interaction Structure on Individual and Shared Mental Models

**Matthew Wood, Pinzhi Chen, Katherine Fu, Jonathan Cagan and Kenneth Kotovsky**

**Abstract** The interaction structure of problem solving teams in design and other domains, and its effects on ideation outcomes is a well-explored topic in the study of team cognition in problem solving and design. Much less is known on how changes in team interaction structure influence the development of mental models over the course of work on a problem. This study aims to understand the relationship between team interaction structure and mental model development by measuring the similarity of individual mental models across time with respect to the individual and other group members. Three-member design teams from upper-level engineering design courses worked either independently or interactively on a mechanical design task for either the 1st half or the 2nd half of the design process. Participants were periodically interrupted for a written description of their mental models of the design process. Descriptions were analyzed with Latent Semantic Analysis to assess mental model convergence. Results show working together has a substantive impact on shared mental models of the design process, and team interaction was associated with more self-consistent mental models of individual team members across time. Working independently was also associated with mental models that were more similar to final design outcomes. Implications for team interaction structure, mental model development, and design fixation are discussed.

## Aims

The goal of this research is to better understand the relationship between team interaction structure (whether members collaborate or not) and the development of similarities in individual team members' representations, or mental models [1], of

M. Wood · P. Chen · K. Fu · J. Cagan (✉) · K. Kotovsky
Carnegie Mellon University, 5000 Forbes, Pittsburgh, PA 15213, USA
e-mail: cagan@cmu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_12,

a problem during design problem solving. This overlap in individual mental models of a task or process is known as a shared mental model [2]. Ideation research suggests that divergence in shared mental models is important for problem solving outcomes [3–5]. This area of inquiry has focused on differences in the quantity, quality, and variety of ideas developed by problem solving teams as a function of working independently or interactively on ideation tasks. In contrast, research in engineering design and team cognition emphasizes convergence on a single problem representation, and focuses on team interaction and collaboration. Such work focuses on the development of one or a small set of ideas into a single design or problem solving solution [6–9]. While both domains provide unique contributions to our knowledge of team design and problem solving processes, their distinct methodologies and literatures make it hard to synthesize research findings [10].

The primary aim of this research is to learn more about how structuring interaction within teams influences the development of mental models at the individual and team level over time. Ideation research finds that a collection of individuals working alone typically produce more ideas than an interacting team, but it is unclear whether individuals or teams produce ideas of higher quality [3, 11]. Although the research on team structure and ideation is robust, less is available on team structure and problem solving processes generally [9]. In particular, little has been done on the role of factors like team interaction structure on the designs or solutions that teams produce [10]. Teams can be better than individuals at evaluating and selecting ideas when fewer alternatives are available, but this is only true when the products of an earlier ideation session are already available for evaluation [12], and does not consider ideas that are created, developed, and refined on-the-fly. Most research focused on team interaction structure is only interested in comparing performance of interacting teams to nominal teams (those whose members work independently) to each other [9]. Such research has little to say on how periods of interaction or independent work may be combined to produce outcomes different from that of collective or independent work exclusively. This research aims to contribute to the literature on structured team interaction, ideation, and problem solving.

A second major aim of this work is to develop new metrics to track differences in individual and shared mental model representations as a function of time and team interaction structure. In this study, we instructed team members to work individually or collaboratively during the 1st half and/or the 2nd half of their problem solving efforts. We aimed to develop new metrics of mental model development using latent semantic analysis (LSA) [13, 14] in an experimental setting. LSA gives an estimate of contextual similarity between two segments of text (called *documents*) by comparing the relative frequencies of words within those texts to a larger corpus of representative content. We aim to extend previous approaches [15] by specifically developing LSA metrics to facilitate understanding of changes in individual mental models over time in the service of illuminating how the interaction structure of design teams influences the development of individual and shared mental models. Learning how mental models change over time with respect to the individual's own cognitions and to design solutions represents a new way of applying LSA to empirical data, and provides a tool to understand trends in individual and team cognition.

## Importance

Problems in the science, engineering, and business communities are increasingly interdisciplinary and collaborative [3]. Much is known about basic cognitive processes of problem solving in domains like engineering design where problems often do not have algorithmic solutions and creative solutions are desirable [16, 17], but there is a paucity of data and theory on the relationship between cognitive processes and team behavior [9].

### *Shared Mental Models and Design*

Shared mental models research is one area where the design literature has contributed to better understanding the contribution of individuals' cognition to team performance. A shared mental model [2] of a problem is the combination of individual team members' knowledge about the task being performed, tools available to execute that task, other team members' skills and abilities, and the procedures for interacting with other team members. This is an extension of the mental model concept developed in cognitive science, representing the structure and content of an individual's knowledge that is used when solving everyday problems. An individual mental model of a problem is often a scaled down, simplified version of the task or problem context that the individual uses to reason about possible approaches and solutions, sometimes using these models to run simulations and forecast consequences [1, 18, 19]. Design researchers have refined the shared mental model concept to highlight the uncertainties often associated with design problem solving, including knowledge of different processes available for solving problems, the type of team (e.g., creative vs. tactical) [20], and organizational context [21], consistent with the tasks that tend to be the focus of design teams. Team cognition research shows that team performance in problem solving tasks is often associated with improvements in the shared mental model of the problem as a result of aligning individual team members' problem conceptualizations [8].

Specific to design, Dong [22] found that *team semantic coherence*, or the contextual similarity of individual team members' ideations, tends to increase over time as team members work together on a project. Individual ideation was measured by collecting emails and other correspondence between team members throughout a project, and analyzing the similarity of these documents to each other using LSA [13, 14]. Importantly, increasing team semantic coherence occurred irrespective of whether written or oral (e.g., transcripts of conversations) communications constituted the corpus of text being analyzed [22]. Dong et al. [6] also found a positive correlation between the semantic similarity of team members' individual mental models of the design process and final design quality, whereby teams with greater semantic coherence also produced higher quality final project designs across a wide range of design domains.

Fu et al. [15] found that design example quality provides one way to influence design team coherence. During the course of an hour-long experiment, four-person student teams worked together to design a peanut sheller for use in developing countries [4]. Along with a statement of the design problem and requirements, teams were given a good example solution that had a few simple parts and was human-powered [23], a poor example solution that had many intricate parts and was gasoline-powered [4], or no example solution. Design teams had 30 min to work on the task, plus an additional 5 min to write up their final designs. Team members were interrupted every 10 min to provide individual reports of the team's best current approach or solution to the problem in order to assess team semantic coherence. Similar to [22], Fu et al. found that team semantic coherence increased over time for teams in all conditions. In addition, teams who were introduced to the problem with a poor example showed less coherence compared to the good example condition from the beginning, and this difference perseverated throughout the course of the experiment. In addition, good example teams produced designs that were significantly higher in quality than the poor design teams. This research demonstrates that (1) design teams, like individuals, become fixated to poor examples [24–26], and (2) that poor examples produce disagreement among team members about the best way to proceed.

## *Ideation and Problem Spaces*

While knowledge about the relationship between individual and team cognition in design and problem solving research in general is underdeveloped (however [21]), a much more robust understanding of the individual and team cognition relationship is available in the brainstorming literature. First developed as a component of Osborn's proscriptive Creative Problem Solving model [5], brainstorming has become a ubiquitous part of the problem solving process for teams and organizations [27]. In the model, brainstorming is associated with a series of rules that explicitly exclude criticism and critical thinking from this stage of the process. These include (1) focusing on quantity (2) ruling out criticism, (3) encouraging unusual ideas, and (4) combining and improving suggested ideas. Once a set of ideas has been generated, a subset is selected for further refinement, combination, and eventual implementation.

Research in design has improved upon these basic tenets of idea generation to produce a variety of additional methodologies. Many of these include graphical representation of ideas in addition to brainstorming's traditional focus on textual representation. For instance, C-sketch [28] asks a team of five designers to sketch a solution, and then pass that sketch to another team member to either modify or use as inspiration. Ideas are passed four times, so that all team members have an opportunity to refine or add to the work of other members. This method was inspired by the textual method 6–3–5 [29], where six-member teams write three ideas and pass them five times as inspiration or for further refinement. These rotational methods are contrasted with gallery approaches, where design teams share concepts with all other members between episodes of ideation, and traditional brainstorming approaches where ideas are shared as they are being produced [4]. Linsey et al. [30] found that rotational methods that include both words and sketches tend to produce the most ideas, but that sketch-only methods tend to produce more high-quality ideas. Linsey et al. also note that, in general, high-quality ideas tend to be the result of development of earlier design concepts.

The semi-directed search process promoted by team ideation models like those described above is analogous to what individuals already tend to do when solving new problems. Cagan and Kotovsky [31] show that individuals solve puzzle problems like the Tower of Hanoi [32] and Balls and Boxes (a Chinese Ring Puzzle isomorph) [33] by a process similar to the machine learning algorithm of simulated annealing [34]. Individuals solve these problems first by randomly selecting among alternatives in an exploratory phase, and then rapidly accelerating toward the goal in a final path to the solution. In much the same way, machine learning algorithms like simulated annealing search a state-space of solutions with the goal of maximizing an objective function. They do so first by taking large steps around the problem space, followed by incrementally smaller ones as they find locations in the state space with higher objective function values. Cagan and Kotovsky used simulated annealing to emulate human problem solving performance on both Tower of Hanoi and Balls and Boxes by assigning random objective function values to all states of the problem but the goal state, and adjusting them over successive runs by a temporal difference schedule. This method enables learning of the objective function within the simulated annealing algorithm and emulates the same final path lengthening behavior exhibited by human subjects over multiple runs.

The team ideation literature has provided some insights on the effectiveness of both *nominal* teams (members work independently) and *interacting* teams in searching a problem space and in selecting areas of the space to pursue for further refinement. Through both empirical work [11, 35] and simulation [36, 37], research indicates that nominal teams generate more ideas than interacting teams in the context of a brainstorming task, a phenomenon referred to as *productivity loss* [38]. Structured ideation models used in engineering design are aimed at overcoming this productivity loss problem. However, literature on both brainstorming and other ideation techniques focus on properties of the resulting pool of ideas (e.g., quantity, quality, novelty) [3, 11, 39] and often do not consider process variables that describe the activity occurring to produce ideas.

Encouraging team members to work independently to explore the problem space, followed by team interaction to select parts of the space to pursue and refine for a solution, may provide the best formula for problem solving success. Just as simulated annealing assures that the system will identify a solution that is a local optimum (potentially the global optimum) by exploring ever smaller areas of the problem space, placing independently-working individuals into interacting teams should encourage solvers to develop more coherent problem representations, on a smaller area of the problem space. As a consequence, search becomes directed toward a smaller region of the space, akin to lowering the step size of the simulated annealing search algorithm. One implication for the outcome is that the final team design solution should identify a local optimum in the search space, and perhaps even a global optimum. A key assumption, however, is that teams are able to assess and agree upon the best ideas from the earlier brainstorming search process.

Recent research suggests that both nominal and interacting teams are unable to select the highest quality ideas from a previously generated pool of candidates. This result has been shown across a wide range of quality metrics, including originality, feasibility, impact, and others. Rietzschel et al. [40], and also [35, 41] find that both nominal and interacting teams performed at chance when attempting to select high-quality ideas from an earlier brainstorming session. This is despite the fact that nominal teams tended to produce more high-quality ideas, though the proportion of high-quality ideas to total ideas generated was similar across nominal and interacting teams. That said, the above studies implemented a process partially representative of the problem solving process in general. The focus was only on the generation and identification of alternatives, and subsequent selection of a subset of them based on their quality. Problem solving teams in the real world frequently combine and modify ideas while searching the problem space for alternatives. In addition, the design of generation then selection of ideas artificially bifurcates the process into two phases, whereas problem solving teams outside of laboratory settings are likely to iterate through idea generation and selection in the course of problem solving.

One reason that interacting problem solving teams may have difficulty in selecting ideas based on quality may be because the team fixates on one category of problem solution, and is then unable to solve the problem effectively because members cannot help themselves but to think about the fixated category [25, 26]. Team coherence in this case would be high, but the focus on a poor solution candidate is detrimental for later team performance. It may then be difficult to compare solutions from the same category to each other on the basis of quality because of their relative similarity compared to the set of solutions generated. Splitting interacting teams into nominal ones may be a way to break category fixation and explore a greater variety of ideas. Kohn [42] finds that an unrelated interruption task is an effective way to reduce team fixation, at least in the course of idea generation. This may be because the interrupted team members are not able to ruminate together on the fixated elements of the problem. In addition [27] and [43] both suggest that allowing team members to interact before splitting them up on their own tends to produce more total ideas than keeping interacting teams together for the second part of the brainstorming session. In this case, and in order to prevent team fixation, it may be best to start with interacting teams and then split them into their nominal components.

## *Hypotheses*

In order to better understand the relationship between team interaction structure and shared mental models development, teams of mechanical engineering students worked collaboratively and independently on the Peanut Sheller design problem. This task that has been used in past engineering design research to better understand team solving processes [4, 15]. The goal of this problem is to design a mechanical system for deployment in resource-poor developing nations that shells peanuts without damaging the fruit. The experiment was conducted with 12 three-member design teams from senior-level and first-year graduate engineering design methods classes as part of a larger study on design cognition and team problem-solving processes. Three-member teams were utilized because this is the minimum number of members required to constitute a group in the psychological literature [44]. The use of three-member teams also decreases the likelihood of social loafing effects, a reduction of individual member effort associated with larger teams [45]. On a more pragmatic note, the primary analysis unit of interest here is the group, and three-member teams help us to maximize the number of available analysis units with limited participant availability.

Teams were instructed to work together (interacting team structure) or independently (nominal team structure) throughout the experiment according to a two-by-two factorial design. The two factors of this design were the team's interaction structure during the 1st half and 2nd half of problem solving (interacting or nominal). This produced four distinct conditions; two that had the same interaction structure (nominal–nominal and interacting–interacting; N–N and I–I, respectively), and two that changed midway through the experiment; nominal-interacting (N–I) and interacting-nominal (I–N), Fig. 1. All teams provided a final design solution or solutions depending on their interaction structure in the 2nd half of the experiment. The following hypotheses were proposed:

**Hypothesis 1** Team interaction structure will influence design team coherence. Consistent with [22], the structure of design team interaction will influence design team coherence. Specifically, interacting teams should produce self-reports of the design process that are more similar to each other than nominal teams after a problem solving episode. This is because interacting teams have had the opportunity to develop a shared mental model of the design process, while nominal teams have not.

**Hypothesis 2** Team interaction structure will influence the *self-consistency* of individual mental models over time, defined here as the semantic coherence of temporally contiguous individual mental models. Specifically, I–I team members should show the greatest self-consistency in their self-reports of the design process

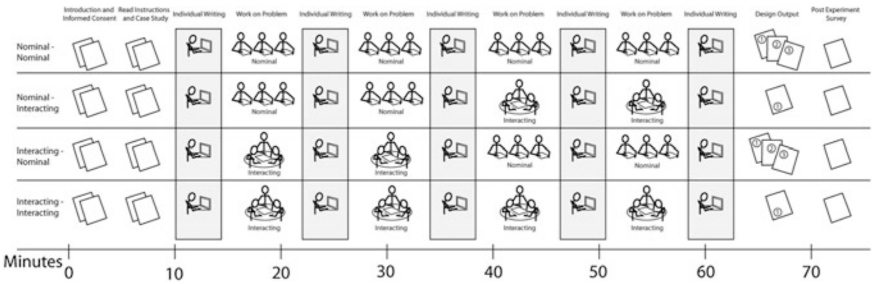


**Fig. 1** Experiment procedure. Row represents condition

compared to other conditions, for three reasons. First, group decision making research shows that teams tend to discuss facts about a problem that all team members know about (shared information) rather than facts of which only one or a few members have knowledge (unshared information) [11, 46]. As a result, teams will tend to discuss problem approaches and solutions for which they share familiarity. Second, simulations of brainstorming show that teams tend to focus on a small number of idea categories because the memories of their members are primed by earlier suggestions [37]. Combined with the tendency to focus on symmetric information, priming should cause teams to gravitate toward a small set of related solution ideas, and this focus should persist over time. Finally, members of interacting teams require time to communicate in the service of sharing and refining ideas. In contrast, members of nominal teams need no time to communicate ideas to their members, and therefore have more time available to explore different aspects of the problem. As a result, they may develop a series of distinct solution concepts that are unrelated to each other.

**Hypothesis 3** Team interaction structure will influence the relationship between individual mental models of the design process and the final design solution. Specifically, N–N teams should show the greatest semantic coherence between self-reports during the design process and their final solution outcomes, because they will be less likely (relative to other teams) to develop alternative ways of framing the problem, i.e., developing a coherent representation of the problems initial conditions and constraints, early in the problem-solving process. Research on the problem solving process with individuals demonstrates that problem framing has important consequences for the quality of later problem solving activities. A common way to facilitate development of good problem frames is to ask individuals to consider as many different ways of restating the problem as possible. Not only do problem framing heuristics like restatement lead to improved solving outcomes, but experts in a wide variety of domains have been shown to spend more time than novices on problem framing [16, 47], and creativity training programs which include a problem framing component are associated with better solution outcomes [48]. Team members who interact have the opportunity to share the differences in how they interpreted the problem early on in problem solving, and are thus exposed to different problem frames. As a result, final design solutions should have little in common with individual team members' early conceptions of the problem. In contrast, individuals who work alone are not given assistance in framing the problem in different ways, and are therefore more likely to fixate on a particular problem frame and neglect alternative approaches as they develop their initial conceptions toward a solution [24–26].

## Method

Three-member teams from senior-level and graduate engineering design courses participated in the experiment for course credit or entry into a raffle for one of four $25 gift cards. They worked on the problem for 28 min in four seven-minute solving sessions. In between sessions, team members wrote independently on laptop computers describing their team's current best approach or solution to the Peanut Sheller problem. Teams were randomly assigned to a team communication condition in a 2 × 2 factorial design, where members either worked independently or collaboratively for the first two problem solving sessions and also for the second two problem solving sessions; Fig. 1. For each of the four problem solving sessions, participants were unaware whether they would be working together or independently until the beginning of that session. Teams had a final 7-min session to write up their solution(s) to the design problem. Teams who worked collectively during the 2nd half of the experiment produced one design solution during this final write-up session. Teams who worked independently during the 2nd half of the experiment produced three distinct solutions, one for each team member. All teams and individuals were instructed to include as part of their solution description: a sketch of the design, labels of major elements, and a few sentences describing how the solution works.

### *Self-Report LSA Analysis*

Team members provided self-reports of their team's current best approach or solution to the Peanut Sheller problem after reading the problem, between problem solving sessions, and before providing their final design solution [15]. LSA [13, 14] was used to compare these self-reports and transcripts of the final design to each other, consistent with past research [15]. LSA produces pairwise comparisons of semantic similarity between documents with values ranging from $-1$ to $+1$. This measure is interpreted like a correlation coefficient, with positive values indicating synonymy, negative values indicating antonymy, and the magnitude reflecting the strength of the relationship [13]. Though the corpus here is relatively small compared to typical LSA applications, similarly sized corpora have been used with satisfactorily [15, 49].

### Metrics

Team *semantic coherence* was computed by taking the mean of all pairwise cosine similarities of self-reports in the final LSA matrix for the members of each team and for each point in time. The result is five semantic coherence values for each team, corresponding to each time during the procedure that a self-report was elicited.

*Self-consistency* was assessed by comparing with cosine similarity, for each individual, the self-report produced at the current time $t$ with the immediately preceding self-report at time $t - 1$. The result is four self-consistency values for each individual, corresponding to pairwise comparisons of contiguous self-reports.

*Similarity to solution* was computed by comparing each self-report for each individual to the transcript of the final design solution for the team or that particular individual, depending on the condition the individual in question was assigned to for the 2nd half of the experiment. The result is five similarity values for each individual.

## Results

### *Hypothesis 1*

Hypothesis 1 was partially supported. Team interaction structure was associated with changes in team coherence, especially during the 2nd half of the experiment. To test this hypothesis, team semantic coherence was compared across conditions using repeated-measures ANOVA with the two between-subjects factors of team interaction structure during the 1st (interacting or nominal) and 2nd (interacting or nominal) halves of the experiment. A marginally significant 2nd half structure by time interaction was identified, $F(4, 32) = 2.44$, $p = 0.07$. Within-subjects contrasts show that changes in team coherence between the individual writing episode after the 1st half (time_2) and the individual writing episode midway through the 2nd half (time_3) depend on 2nd half structure, with interacting teams showing greater coherence than nominal teams, $F(4, 32) = 9.53$, $p = 0.02$; Fig. 2. This influence of interaction structure is also seen in the 1st half of the experiment, especially from time_0 to time_1, although the time by 1st half interaction was not significant. Thus, Hypothesis 1 is partially supported, suggesting interacting teams in this study had the opportunity to develop a shared mental model of the design process over the 2nd half of the experiment.

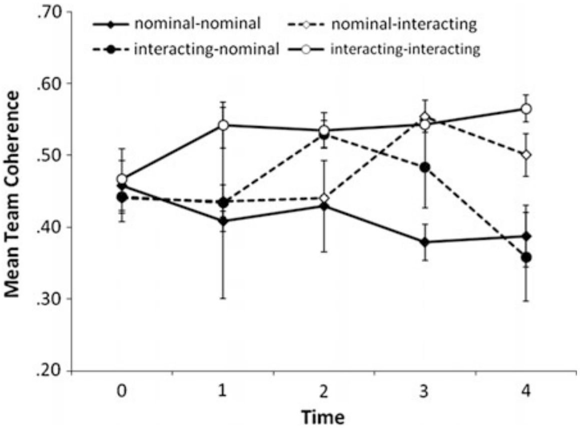


**Fig. 2** Team coherence as a function of time and condition. Error bars ±1SE

## Hypothesis 2

Hypothesis 2 was supported; self-consistency changed as a function of time and interaction structure. To test the hypothesis, self-consistency scores were compared across conditions using repeated-measures ANOVA with the two team interaction structure between-subjects factors (1st half, 2nd half). A main effect of time was identified, $F(3, 96) = 11.96$, $p < 0.01$. Within-subjects contrasts show a significant difference between self-consistency at time_0–1 and time_1–2, $F(1, 32) = 12.04$, $p = 0.01$, and a marginally significant difference between time_1–2 and time_2–3, $F(1, 32) = 3.02$, $p = 0.09$. These results suggest that, in general, self-consistency increases less over time, asymptoting after time_2–3; Fig. 3. This analysis also identified a main effect of 2nd half structure, $F(1, 32) = 6.57$, $p = 0.02$, that was qualified by a marginally significant 1st half by 2nd half structure interaction, $F(1, 32) = 3.17$, $p = 0.08$. Means of self-consistency across time for each condition demonstrate that self-consistency is moderated by team structure in the 1st half of the experiment, with teams that work together in the 1st half of the experiment showing larger differences in self-consistency across the 2nd half of the experiment Fig. 4.

Support is therefore found for Hypothesis 2. I–I teams tended to show the greatest self-consistency. Furthermore, the interaction suggests that working together early in the course of team problem solving has an important impact on subsequent solving activity, with I–I teams showing greater self-consistency on average and I–N teams showing less self-consistency as compared to teams who worked alone early in solving (N–I and N–N).

**Fig. 3** Self-consistency as a function of time and condition. Error bars ±1SE

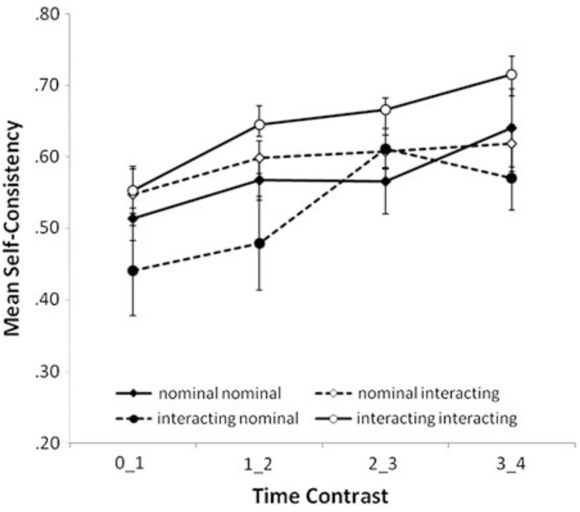
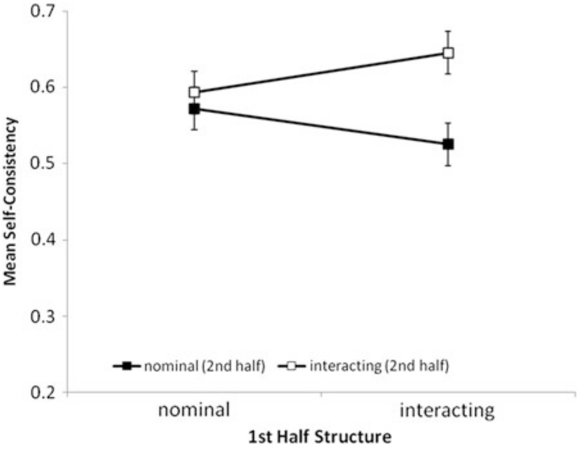


**Fig. 4** Self-consistency as a function of condition. Error bars ±1SE



## *Hypothesis 3*

Support was found for Hypothesis 3. Team interaction structure influenced the relationship between individual mental models of the design process and the final design solution. In order to test this hypothesis, similarity to solution scores were compared across conditions using repeated-measures ANOVA with the two team interaction structure between-subjects factors (1st half, 2nd half). This analysis identified a main effect of time, $F(4, 128) = 18.67$, $p < 0.01$; Fig. 5. Post hoc contrasts show a significant difference between time_0 and time_1, $F(1, 32) = 8.86$, $p < 0.01$, a marginally significant difference between time_1 and time_2, $F(1, 32) = 3.01$, $p = 0.09$, and a significant difference between time_2

**Fig. 5** Similarity to solution as a function of time and condition. Error bars ±1SE

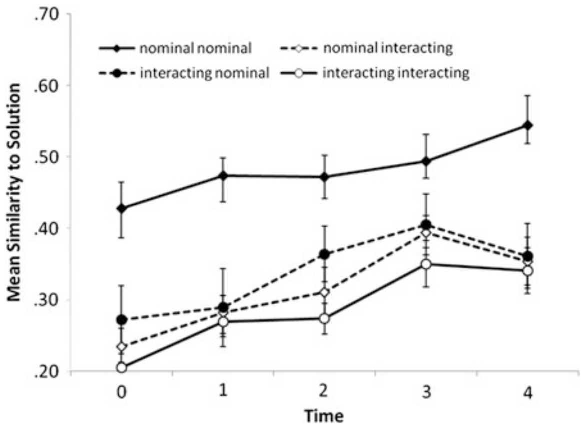
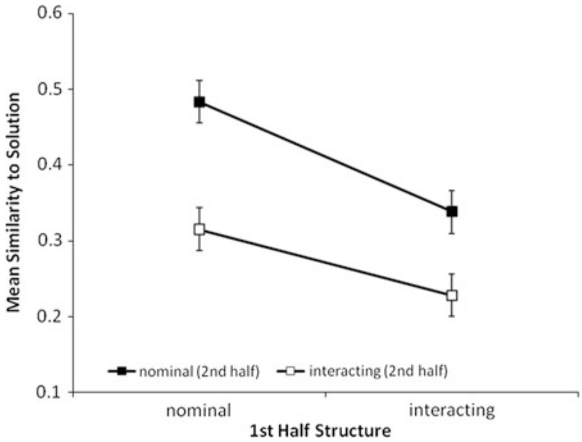


**Fig. 6** Similarity to solution as a function of condition. Error bars ±1SE



and time_3, $F(1, 32) = 11.53$, $p < 0.01$. As with self-consistency scores (Fig. 3), similarity to solution increased over time, but leveled off toward the end of the experiment. In addition, main effects were identified for both 1st half structure, $F(1, 32) = 9.96$, $p < 0.01$, and 2nd half structure factors, $F(1, 32) = 16.07$, $p < 0.01$. These main effects were qualified by a 1st half structure by 2nd half structure interaction, $F(1, 32) = 4.63$, $p = 0.04$. Teams who spent time working alone had greater similarity to solution scores on average than interacting teams. In addition, teams who interacted during the 1st half of the experiment had a smaller difference in similarity to solution scores than nominal 1st half teams as a function of 2nd half team structure; Figs. 5 and 6. These results are consistent with Hypothesis 3. N–N teams gave self-reports of the design process that were most similar to their final design solutions, evidence consistent with problem frame fixation. Interacting during either the 1st or 2nd half of problem solving appears to prevent fixation on one particular way to frame the problem, as teams who spent any time interacting gave self-reports with much less in common with their final design solutions. Indeed, I–I teams gave self-reports least similar to their final design solutions of all four conditions.

## Conclusions

In sum, this preliminary study provides clear evidence of a relationship between team interaction structure and changes in individual and shared mental model development, and team design quality. Interacting teams had similar individual mental models of the problem solving process during interaction, while nominal teams had less similar individual mental models. Individual mental models of the design process appeared to become more self-consistent, and presumably led to more stable problem representations as a function of work over time on the design task. In addition, interacting with other team members appears to help individuals to avoid fixation on sub-optimal problem frames. Interaction during the 1st or 2nd half of problem solving has a roughly additive effect on reducing problem frame fixation, while working entirely alone is associated with a greater focus on the same concepts these solvers used when initially framing the problem.

### *Limitations*

A few limitations of this study should be noted. First, these data represent preliminary results as it relates to developing a better understanding the relationship between shared mental models and team interaction structure. A larger sample should provide better estimates of change in mental model representations as a function of team interaction structure by reducing measurement error. In addition, while participants in these studies were representative of populations similar to those used in past research on the relationship between shared mental models and team design quality [6, 15, 49], student design teams in controlled settings like these only serve as proxy for the real-world design teams that researchers hope to inform with work like that presented here.

### *Contributions*

This research provides several contributions to our knowledge of mental model development in design teams. First, changes in team interaction structure were shown to have an effect on shared mental model development. When interacting with other team members, a period of as little as 7 min provided a substantial increase in the similarity of individual team members' self-reports of the best way to proceed on a design problem. Work alone for a similar period of time was shown to decrease the similarity of these self-reports; Fig. 2. While such changes may be indicative of agreement between team members on common terminology and ways of expressing ideas [50, 51], the fact that these trends are seen using a relatively coarse sampling technique suggests that team members are agreeing on more than merely what words to use for describing particular problem concepts. These trends were identified with just a few sentences of written description, while most studies of verbal economy, or the convergence of speakers toward similar vocabulary and grammatical structures over time, require much denser sampling over shorter timespans. Furthermore, the LSA metrics used to measure changes in mental models in this study are insensitive to differences in grammar and syntax consistent with a verbal economy explanation.

One interesting question raised by this research is the extent to which team structure effects seen over relatively short intervals here may scale to greater durations of problem solving effort (e.g., days). References [6] and [22] find team semantic coherence tends to increase over time for efforts spanning the weeks to months range, at least for interacting teams. It is unclear what the influence of longer durations might be on nominal teams, or teams who work under a variety of team interaction structure conditions over longer durations.

In addition, this research represents one of the first attempts to better understand how team interaction structure changes the mental models of individuals as it relates to a particular problem. One interesting finding is that N–N teams had the highest similarity between their individual mental models of the problem and the final design solution, and at the same time the lowest team coherence. These results are suggestive of individuals who are exploring quite different areas of the problem space, or at least are describing their exploration of overlapping problem spaces very differently. It should be noted that self-reports of nominal 2nd-half teams were compared to their own distinct design solutions, while self-reports of interacting 2nd-half teams were compared to a single final design solution. An interesting area of future inquiry would be to better understand the effect of independent work on individual mental models of the same problem, as well as the language that individuals use to describe these mental models.

Preliminary results here suggest that team interaction structure has other important implications for mental model development, including shared mental models, consistency across individual mental model representations, and consistency between individual mental model representations and final design solution outcomes. In particular, both results related to self-consistency and similarity between self-reports and solutions have important implications for our knowledge of fixation in team design.

On the matter of individual mental model self-consistency, it should be noted that, while the term *fixation* is typically associated with detrimental outcomes [24, 26], fixation only refers to focusing on a particular set of problem features or elements. When these features are important for solving the problem, fixation on such features may facilitate design problem solving. Self-consistency provides an assessment of problem representation stability that is independent of problem solving outcome.

Self-consistency should increase as an individual focuses on more and more specific problem elements or solutions, and should decrease when that individual considers different problem elements and solutions. The results of this experiment suggest that teams that begin work by interacting with each other may tend to fixate on particular problem aspects and solutions if allowed to continue working together, as evidenced by the relatively high self-consistency scores of these team members. In contrast, an opportunity to break out of team interaction may also help members to avoid fixating on particular design elements.

As it relates to similarity between individual mental models and final design solutions, this research shows that one benefit of working with other team members during the early phases of team design is that team members are able to alert each other to other ways of framing the design problem than the first one or two that come to mind. This may lead to discussion of problem framing alternatives, and serve as a pre-emptive strike against fixation on sub-optimal problem frames. Future work should focus on a more detailed analysis of problem frames proposed early in the team design process, and the process by which some problem frames are selected over others.

## References

1. Craik KJW (1943) The nature of explanation. University Press, Cambridge
2. Cannon-Bowers JA, Salas E, Converse S (1993) Shared mental models in expert team decision making. In: Castellan NJ Jr (ed) Individual and group decision making: current issues. Lawrence Earlbaum Associates Inc., Hillsdale, pp 221–246
3. Paulus PB, Dzindolet M, Kohn NW (2011) Collaborative creativity—group creativity and team innovation. In: Mumford MD (ed) Handbook of organizational creativity. Elsevier, New York, pp 327–357
4. Linsey JS, Green MG, Murphy JT, Wood KJ, Markman AB (2005) "Collaborating to success": an experimental study of group idea generation techniques. In: ASME 2005 international design engineering technical conferences and computers and information in engineering conference. Long Beach, CA
5. Osborn AF (1957/1993) Applied imagination, 3rd revised edn. Creative Education Foundation, Buffalo NY
6. Dong A, Hill AW, Agogino AM (2004) A document analysis method for characterizing design team performance. J Mech Des 126:378–385
7. Fiore SM, Rosen MA, Smith-Jentsch KA, Salas E, Letsky M, Warner N (2010) Toward an understanding of macrocognition in teams: predicting processes in complex collaborative contexts. Hum Factors 52:203–224
8. Salas E, Cannon-Bowers JA (1997) Methods, tools, and strategies for team training. In: Quinones M, Ehrenstein A (eds) Training for a rapidly changing workplace. American Psychological Association, Washington DC, pp 249–279
9. Reiter-Palmon R, Herman AE, Yammarino FJ (2008) Creativity and cognitive processes: multi-level linkages between individual and team cognition. In: Mumford MD, Hunter ST, Bedell-Avers KE (eds) Multi-level issues in creativity and innovation, vol 7. Emerald, Bingley, pp 203–267

10. Reiter-Palmon R, Wigert B, de Vreede T (2012) Team creativity and innovation: the effect of group composition, social processes, and cognition. In: Mumford MD (ed) Handbook of organizational creativity. Academic Press, New York
11. Kerr NL, Tindale RS (2004) Group performance and decision making. Annu Rev Psychol 55:623–655
12. Mumford MD, Feldman JM, Hein MB, Nagao DJ (2001) Tradeoffs between ideas and structure: individual versus group performance in creative problem solving. J Creative Behav 35:1–23
13. Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41:391–407
14. Landauer TK, Foltz PW, Laham D (1998) An introduction to latent semantic analysis. Discourse Process 25:259–284
15. Fu K, Cagan J, Kotovsky K (2010) Design team convergence: the influence of example solution quality. J Mech Des 132:1–10
16. Mumford MD, Reiter-Palmon R, Redmond MR (1994) Problem construction and cognition: applying problem representations in ill-defined domains. In: Runco MA (ed) Problem finding, problem solving, and creativity. Ablex Publishing, Westport CT, pp 3–39
17. Reiter-Palmon R, Mumford MD, Threlfall KV (1998) Solving everyday problems creatively: the role of problem construction and personality type. Creativity Res J 11:187–197
18. Bartlett FC (1932) A theory of remembering. In remembering: a study in experimental and social psychology. Cambridge University Press, London, pp 197–214
19. Gentner D, Stevens AL (1983) Mental models. Lawrence Erlbaum Associates, Hillsdale
20. Larson CE, LaFasto FMJ (1989) Teamwork: what must go right, what can go wrong. SAGE Publications, Newbury Park
21. Badke-Schaub P, Neumann A, Lauche K, Mohammed S (2007) Mental models in design teams: a valid approach to performance in design collaboration? CoDesign 3(1):5–20
22. Dong A (2005) The latent semantic approach to studying design team communication. Des Stud 26:445–461
23. Full Belly Project (2010) Full belly project—nut sheller. http://www.thefullbellyproject.org/Products/UniversalNutSheller.aspx
24. Jansson DG, Smith SM (1991) Design fixation. Des Stud 12:3–11
25. Smith SM, Blankenship SE (1989) Incubation effects. Bull Psychon Soc 27:311–314
26. Smith SM, Blankenship SE (1991) Incubation and the persistence of fixation in problem solving. Am J Psychol 104:61–87
27. Paulus PB, Brown VR (2002) Making group brainstorming more effective: recommendations from an associative memory perspective. Curr Dir Psychol Sci 11:208–212
28. Shah JJ, Vargas-Hernandez N, Summers JD, Kulkarni S (2001) Collaborative sketching (c-sketch)-an idea in generation technique for engineering design. J Creative Behav 35(3):168–198
29. Otto KN, Wood KL (2001) Product design: techniques in reverse engineering and new product development. Prentice Hall, Upper Saddle River
30. Linsey JS, Clauss EF, Kurtoglu T, Murphy JT, Wood KL, Markman AB (2011) An experimental study of group idea generation techniques: understanding the roles of idea representation and viewing methods. J Mech Design 133(3):031008
31. Cagan J, Kotovsky K (1997) Simulated annealing and the generation of the objective function: a model of learning during problem solving. Comput Intell 13:534–581
32. Kotovsky K, Hayes JR, Simon HA (1985) Why are some problems hard? Evidence from Tower of Hanoi. Cogn Psychol 17:248–294
33. Reber PJ, Kotovsky K (1997) Implicit learning in problem solving: the role of working memory capacity. J Exp Psychol Gen 126:178–203
34. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–680
35. Putman VL, Paulus PB (2009) Brainstorming, brainstorming rules and decision making. J Creative Behav 43:23–39

36. Brown VR, Paulus PB (1996) A simple dynamic model of social factors in group brainstorming. Small Group Res 27:91–114
37. Brown VR, Tumeo M, Larey TS, Paulus PB (1998) Modeling cognitive interactions during group brainstorming. Small Group Res 29:495–526
38. Diehl M, Stroebe W (1987) Productivity loss in brainstorming groups: toward the solution of a riddle. J Pers Soc Psychol 53(3):497–509
39. Hernandez NV, Shah JJ, Smith SM (2010) Understanding design ideation mechanisms through multilevel aligned empirical studies. Des Stud 31(4):382–410
40. Rietzschel EF, Nijstad BA, Stroebe W (2006) Productivity is not enough: a comparison of interactive and nominal brainstorming groups on idea generation and selection. J Exp Soc Psychol 42:244–251
41. Faure C (2004) Beyond brainstorming: effects of different group procedures on selection of ideas and satisfaction with the process. J Creative Behav 38:13–34
42. Kohn NW (2008) An examination of fixation in brainstorming. Texas A&M University, College Station
43. Baruah J, Paulus PB (2008) Effects of training on idea generation in groups. Small Group Res 39:523–541
44. Laughlin PR (2011) Group problem solving. Princeton University Press, Princeton
45. Latane B, Williams K, Harkins S (1979) Many hands make light the work: causes and consequences of social loafing. J Pers Soc Psychol 37(6):822–832
46. Stasser G, Titus W (1985) Pooling of unshared information in group decision making: biased information sampling during discussion. J Pers Soc Psychol 48(6):1467–1478
47. Reiter-Palmon R, Robinson EJ (2009) Problem identification and construction: what do we know, what is the future? Psychol Aesthetics Creativity Arts 3(1):43–47
48. Scott G, Leritz LE, Mumford MD (2004) The effectiveness of creativity training: a quantitative review. Creativity Res J 16(4):361–388
49. Strait MJ, Haynes JA, Foltz PW (2000) Applications of latent semantic analysis to lessons learned systems. In: Aha DW, Weber R (eds) Intelligent lessons learned systems: papers from the 2000 workshop. AAAI Press, Menlo Park
50. Matessa M (2000) Simulating adaptive communication. Carnegie Mellon University, Pittsburgh
51. Pickering MJ, Garrod S (2004) Toward a mechanistic psychology of dialogue. Behav Brain Sci 27:169–190

# An Empirical Study of the Effectiveness of Selected Cognitive Aids on Multiple Design Tasks

**Noe Vargas Hernandez, Linda C. Schmidt, Gul Okudan Kremer and Chun-Yu Lin**

**Abstract** The objective of this work is to study the concept generation effectiveness of three cognitive design aids: TRIZ—an ideation method, Sketching—a representation format, and use of the Smartpen—a journaling technology. The hypothesis is that TRIZ, Sketching and Smartpen, each improve the effectiveness of the concept generation process. The participating subjects belong to Penn State's Introduction to Work Design (IE 327) course. The course focuses on concepts of work design and measurement applied to manufacturing and service industries with a focus on improving worker performance, health and safety analyses. In the paper, we report on two sequentially completed design case studies, which allowed us to study the same group of subjects under two conditions. The first case study involved redesigns of a wire-cutter and a screw driver to improve work productivity. The second case consisted of analyzing an ultrasound operation for which students suggested improvements to the workplace and a redesign of the ultrasound transducer taking into account ergonomics and human factors principles. Our results indicate that indeed the tested design aids improved the ideation effectiveness; Smartpen has done the best in terms of increasing quantity of ideas generated, and TRIZ was the best in enhancing novelty.

---

N. V. Hernandez (✉)
University of Texas at El Paso, El Paso, USA
e-mail: nvargas@utep.edu

L. C. Schmidt
University of Maryland, College Park, MD, USA

G. O. Kremer · C.-Y. Lin
Pennsylvania State University, University Park, PA, USA

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_13,

## Introduction

Creativity in engineering design contexts is an important element of innovation. Accordingly, practitioners in industry [1] and academicians are striving to develop tools and methods to improve creativity [2–4]; and appropriately, researchers have been investigating the impact of these tools and methods. In this paper, we investigate the impact on ideation effectiveness of three design tools and methods: sketching, sketching with Smartpen and TRIZ. Although dissimilar (sketching is a representation method, Smartpen is a technology tool, and TRIZ is an ideation method), these three design aids provide a cross-cutting view of creativity in engineering design, particularly in design—related courses. Since these design aids contribute to the creativity process, the authors refer to them as cognitive aids in the context of this paper.

Sketching has been recognized as a tool to improve ideation. McCormick [5] summarizes the importance of sketching as follows: “Sketching is the tool for innovation, and is so vital to the engineering process that it should be taught and used as an essential part of engineering education and professional practice.” Shah et al. [6] also showed that sketching has advantages for collaborative design. Despite this fact, however, prior work on the impact of sketching on ideation is limited in many ways. For example, Bilda et al. [7] conducted a think-aloud experiment with experienced architects to see if sketching is necessary in conceptual design. Two separate design processes were employed: one with sketching and the other prohibiting sketching and using a blindfold. The data was analyzed on the design outcome, cognitive activity and idea links. They report no significant difference between sketching and not sketching during conceptual design. However, they included only three subjects in the study, and hence, the result is not widely generalizable. In addition, while designing across different domains may have commonalities, transference of the results to engineering design domain may not be expected.

The Smartpen technology (www.livescribe.com) brings potentially additional cognitive benefits to sketching while designing: a Smartpen is a writing device that creates a visual recording of everything written or drawn with the pen tip via an infrared camera. The Smartpen can simultaneously create an audio recording. The device is designed to be used with special paper (provided in notebooks) so that the visual and audio records can be uploaded to a website in digital form for storage and playback via Livescribe software. Livescribe software also allows the sharing of the files by way of “pencasts” through email, Google Docs, facebook and similar sites. The simultaneous recording of the audio is useful in two ways: (1) it links what the user is hearing to what he is writing or sketching; and (2) the digitized pencasts can be transmitted to others through email or networking sites.

TRIZ is a systematic approach to the generation of innovative designs to seemingly intractable problems. It was first developed in Russia by Genrich Altshuller [8] in the early sixties and seventies and has been used for many years in

Europe and Asia. It is based on the analysis of thousands of patents. These original analyses articulated numerous solution patterns from diverse disciplines. The patterns and the tools are continually being updated by researchers worldwide.

TRIZ provides steps that allow design teams to avoid the “psychological inertia” that tends to draw them to common, comfortable solutions when better, non-traditional ones may exist. Despite the anecdotal evidence that TRIZ helps designers to be more creative, no comprehensive study showing its effectiveness has been done. For example, Ogot and Okudan [9] discussed the suitability of TRIZ for its introduction within the engineering curricula; Shirwaiker and Okudan [10] reviewed the design for manufacturing, manufacturing processes, and systems related TRIZ applications; and Shirwaiker and Okudan [11] proposed an ideation approach, which combines TRIZ and Axiomatic Design; and Okudan Kremer et al. [12] showed its effectiveness when used with mathematical programming. However, none of these studies address TRIZ effectiveness as an ideation tool relative to other ideation tools or cognitive aids.

Given the highlighted need for understanding effectiveness of sketching, sketching with Smartpen technology and TRIZ, we have undertaken empirical experimentations; here, we report our findings from one of our designed experiments. We assert that the ideation effectiveness comparisons of sketching and sketching with Smartpen with TRIZ are especially important in that while sketching can be considered more natural, intuitive for most engineers, TRIZ may not be. Below, we first review the recent works with a similar focus before we discuss our experimental design and results.

## Literature Review

Stavridou and Furnham [13] state that during any creativity experiment, a researcher can focus on four major aspects: (1) the creative process, (2) the creative subject, (3) the creative outcome, and (4) the creative environment, where the task is performed by the subject following a process that is either predetermined (i.e., following a formal method) or not in order (i.e., freestyle creativity) to generate an outcome. The research questions could involve an intervention to the subject (e.g., designer) or to the process (e.g., ideation methods) or to improvements in the outcome. Most recent studies in this domain, however, focused either on studying the process and the individual’s designing (process-oriented studies), or the outcomes of the process (outcome-oriented studies).

Process-oriented studies of design creativity usually have been done through protocol studies. Such a study consists of a non-unique decomposition of the process and evaluation of each step using different metrics. In the 90s, there was an increase in the number of protocol studies of design constructed as studies of design activity. Protocol studies suggest that creativity is related to the discovery process and it can be measured in this stage. For example, Dorst and Cross [14] used protocol studies to identify creative aspects in the design related to formulation of the design problem. Protocol analyses are labor-intensive; however, they provide the best way to explain the influence of the experimental technique, context and cognitive aspects of creativity within the process framework in an explicit way.

The outcome–oriented approach hypothesizes that any intervention influencing the subject or the process will be reflected in the output [15]. If the outcome is creative, then it will be assumed that the intervention had a positive effect. Table 1 below summarizes a sample of most recent output-oriented studies. As can be viewed in the table, in these studies data collection was done either in individual [16] or team level [17, 18] and using tasks that are decidedly easy to understand in a short amount of time. Duration of the idea generation in class ranged between 20 min and 1 h; in some cases, in class idea generation was complemented with incubation periods outside of the class [3].

Most studies (see Table 1) included metrics with specific definitions in order to more objectively analyze the data. Direct quantity measures dominate the studies, as is to be expected in any experimental undertaking. Quantity metrics are usually objective to implement and provide data that is analysis ready. Quantity of ideas is particularly important in creativity studies as it is a measure of fluency in creativity terms. Variations of quantity metrics have been used as well (e.g., number of unique ideas, number of analogous ideas, etc.). For evaluation of design quality, criteria-based judgment has been used [19, 20].

Novelty is also among the most frequently used metric in these studies. Novelty is a measure of how unusual an idea is as compared to other ideas. It relates to expanding the solution space, and is calculated by collecting and categorizing all ideas generated per design function, and counting the number of instances of a particular one given the whole idea set [15]. Novelty measurement is important in understanding how unique the generated ideas are.

An important observation relevant to the sample studies is that one single intervention has been tested per treatment group and personal qualities of the subjects have been assumed to be equivalent (e.g., personality, creativity levels, gender, etc.); one exception to this is that of White et al. [21] where the authors used the self-assessment of creativity on Gough’s descriptors.

In an effort to understand how different cognitive aids might impact subjects differently, this study analyzes the novelty and quantity of generated ideas by student teams across two design case studies where subjects were given the benefit of two different cognitive aids across the cases.

Selection of the cognitive aids deserves explanation: we opted to experiment with cognitive aids that might be perceived as (1) more intuitive versus not, (2) requiring additional training versus not, (3) has potential to prompt distant analogies versus not. For example, while sketching might be perceived to be more natural to an engineering student, TRIZ may not be; the training amount to grasp TRIZ will be longer than that of sketching for many individuals; finally, TRIZ has the potential to retrieve design principles that are not immediately thought of by the designer.

**Table 1** A sample of outcome-oriented studies

| Authors (year) | Subjects | Task | Duration | Design representations | Outcome measures |
|---|---|---|---|---|---|
| Yang [18] | Three courses approx 24 ME students each, subjects work in teams of 2–5 | Pop a helium balloon suspended over a water pond | 10 weeks | Paper based logbooks | Final grade; team ranking in competition |
| Okudan et al. [3] | 121 engineering students, and 27 non-engineering students working in 4 person teams | Design of an affordable biomass cooker for rural communities | 30-min in class idea generation period, followed by a 5 day incubation | Sketch and written explanation | Novelty; variety; quantity; number of unique ideas per experimental group |
| White et al. [21] | 8 senior level design teams (5 or 6 per team) and 7 freshmen level teams (2 per team); pre-and post- test: self-assessment of creativity on Gough's descriptors | Design task differed by team | For each ideation method 30–60 min of training on methods;<br>60–90 min for idea generation | No information given | Quantity |
| Linsey et al. [19] | 60 ME seniors in design methods course participants; extra credit | Quick peanut shelling | 40-min idea generation period | Gallery View and rotational view of text only, sketches only, and text with sketches | Quantity; quality (rated by two judges); novelty; variety—measured per person |
| Vargas Hernandez et al. [17] | 350 engineering undergrads, 14 groups of 25, 256 undergrad students from intro to psychology | Engineering: transport ping pong ball; psychology: tools for alien race | In class ideation sessions 20 min each, one week incubation | Sketches with labels, except for flexible representation (sketch vs. text vs. free J) | Quantity, quality, novelty, variety |

(continued)

**Table 1** (continued)

| Authors (year) | Subjects | Task | Duration | Design representations | Outcome measures |
|---|---|---|---|---|---|
| Lopez et al. [16] | 17 Undergraduate ME students and 1 graduate student | Quick peanut shelling | 40-min idea generation period, followed by subject rating of ideas and surveys | No information given | Quantity of : ideas, analogous (and non) ideas, and emergent features; semantic distance; similarity rating |
| Chan et al. [20] | 153 senior engineering students (95 % ME) three factor- two levels each factorial experiment | Low-cost, portable device to harvest energy from human motion | 30-min idea generation period examples were provided at specific time intervals | Sketches | Degree of solution transfer; quantity; breadth of search; Novelty; quality rated by two students on 6 performance dimensions |
| Cardosa and Badke Schaub [28] | 60 industrial design engineering students, fourth year master students. | Device to retrieve a book from an out-of-reach shelf | 1 h idea generation period | Sketch and written explanation | Quantities: ideas, key example attributes, solution categories; originality; ease of use and manufacture; book damage |

Indeed, intuitiveness of sketching for many designers is clear in its widely used description: sketching is a designer's conversation with themselves. Further, researchers in engineering, architecture, art, education, and psychology used protocol analyses to ascertain cognitive aspects of sketching in design; the reader is referred to analysis of work on that topic by Purcell and Gero [22]. More recently, Cardella et al. [23] conducted a protocol study on engineering students, where the results reinforced that sketching supports communication and that sketching is a large part of the problem scoping stage. This study also correlated the representation activities (like sketching) to higher quality solutions. This finding is consistent with various other researchers whose work either showed that sketching aids designers/engineers work [24, 25] or there is a link between sketching and design thinking [7, 26].

We have used Smartpen-based sketching in our experimentation, along with regular (i.e., paper and pencil-based) sketching, which provides audio support as the designer reviews and progresses their design through sketching.

TRIZ has been recognized as a concept generation process that can develop clever solutions to problems by using the condensed knowledge of thousands of past inventors. The power of TRIZ, therefore, is its inherent ability to bring solutions from diverse and seemingly unrelated fields to bear on a particular design problem, yielding breakthrough solutions. Overall, TRIZ invites the designer to use a ready pool of knowledge for inspiration, retrievable through a systematic procedure. This systematic procedure affords the designer the benefit of a set of design principles that have worked before; in many cases these design principles can be considered to act as analogies that may not be native to the designer.

In the next section we explain the experimental set-up

## Methods

The experiment included students ($N = 79$) from an Introduction to Work Design (IE 327) course, where all the students are junior-level industrial engineering majors. The course focuses on concepts of work design and measurement applied to manufacturing and service industries to improve worker performance through health and safety analyses. Throughout the semester students participate in eight lab sessions where they experiment with certain products or work settings that are to be redesigned. Two case studies were selected from these eight lab sessions that were conducted four weeks apart. The first took place during the 4th week, and the second during the 8th week of the semester.

*Case Study* 1 The first case study involved redesigning a wire-cutter and a screwdriver in order to reduce the Cumulative Trauma Disorder (CTD) related injuries in an assembly plant. The lab session started with observing a video clip and then performing a CTD Risk Analysis on assembler's right hand. Then, in order to estimate how much force is required to do the job, students were asked to cut several wires, and then squeeze the grip dynamometer equally hard. After this hands-on experience, students were asked to redesign the wire-cutter and the screwdriver to reduce the CTD risk and make the assembly possible with less force.

*Case Study* 2 The second case study focused on sonography, a diagnostic medical procedure that uses high frequency sound waves (ultrasound) to produce dynamic visual images of organs, tissues, or blood flow inside the body. The process involves placing the ultrasound transducer against the patient's skin near the body area to be imaged. Musculoskeletal disorders (MSDs) are common amongst sonographers. Students are provided with a set of survey results indicating anatomical sites of discomfort, percentage of sonographers affected, and types of activities leading to discomfort and pain, and are then asked to design a better ultrasound sonography process addressing both the physical, musculoskeletal issues stemming from using the physical device itself.

The course (IE 327) lab has six sections from which four were used in the experiment (for three treatments and one control group). Each of these four sections contains five groups of three or four people. Three different cognitive aids for concept generation were tested: TRIZ, sketching, and sketching using a Smartpen. The cognitive aids were randomly assigned to each section. In addition, one of the sections was used as the control group and no specific cognitive aid was assigned to the students. The lab instructor explained the case study and presented the information to each section. The lab instructor trained each section in their respective cognitive aid (i.e., treatment). The students were given a full week to come up with their redesigns. There were no constraints on the time allowed to come up with the ideas or the number of ideas. The overall case grade took into consideration all the aspects of the case study, including the tool redesign as well as the workstation, process, etc. It also took into consideration the report format, and grammar. Note that even though all the students agreed on participating in the study and signed the consent form, there were some groups that did not follow the method assigned to them; hence, relevant data points were excluded from the analysis.

## Results

Upon return of the student work, lab reports were graded for correctness of the technical content by the lab instructor; Table 2 below displays these results for each treatment group.

Submitted designs from each group were tabulated describing each idea provided (Case 1 broken down into functions: screwdriver and wirecutter). The tabulated data were then used to calculate the quantity (total ideas generated by each team) and novelty (indicating how unique each provided idea is). Quantity and novelty data are shown in Tables 3 and 4. Sample designs are also provided in the Appendix.

**Table 2** Ideation methods assigned to groups and grades

| Section # | Group # | Gender | # Students per group | Method for CS#1 | Used the method? | Case study #1 grade | Method for CS#2 | Used the method? | Case study #2 grade |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1F3M | 4 | Sketching | Y | 85.5 | TRIZ | Y | 99 |
| 1 | 2 | 4M | 4 | Sketching | Y | 68.5 | TRIZ | Y | 76 |
| 1 | 3 | 4M | 4 | Sketching | Y | 80 | TRIZ | Y | 94 |
| 1 | 4 | 2F2M | 4 | Sketching | Y | 88.5 | TRIZ | Y | 77.5 |
| 1 | 5 | 1F3M | 4 | Sketching | Y | 89 | TRIZ | Y | 93 |
| 2 | 1 | 4M | 4 | Control | Y | 74.5 | Sketching | Y | 93.5 |
| 2 | 2 | 1F2M | 3 | Control | Y | 85 | Sketching | Y | 97 |
| 2 | 3 | 3M | 3 | Control | N | 75.5 | Sketching | Y | 78.5 |
| 2 | 4 | 4M | 4 | Control | Y | 85 | Sketching | Y | 87.5 |
| 2 | 5 | 4M | 4 | Control | Y | 77.5 | Sketching | Y | 97.5 |
| 3 | 1 | 3F2M | 5 | TRIZ | Y | 91 | Smart pen | Y | 90 |
| 3 | 2 | 1F3M | 4 | TRIZ | N | 84 | Smart pen | Y | 83 |
| 3 | 3 | 1F3M | 4 | TRIZ | N | 83 | Smart pen | N | 84.5 |
| 3 | 4 | 4M | 4 | TRIZ | Y | 96.5 | Smart pen | Y | 92 |
| 3 | 5 | 4M | 4 | TRIZ | N | 80 | Smart pen | N | 89.5 |
| 4 | 1 | 2F2M | 4 | Smart pen | Y | 77.5 | Control | Y | 89 |
| 4 | 2 | 1F3M | 4 | Smart pen | Y | 94 | Control | Y | 87.5 |
| 4 | 3 | 1F3M | 4 | Smart pen | Y | 90 | Control | Y | 94 |
| 4 | 4 | 4M | 4 | Smart pen | Y | 83.5 | Control | Y | 83.5 |
| 4 | 5 | 1F3M | 4 | Smart pen | Y | 83.5 | Control | Y | 90.5 |

**Table 3** Quantity results (quantity values for case study #1 covers both designs)

| Section # | Case study #1 | Quantity screwdriver | Quantity wirecutter | Case study #2 | Quantity |
|---|---|---|---|---|---|
| 1 | Sketching | 3.00 | 3.00 | TRIZ | 2.20 |
| 2 | Control | 1.50 | 3.50 | Sketching | 2.80 |
| 3 | TRIZ | 5.00 | 2.50 | Smart pen | 3.67 |
| 4 | Smart pen | 4.6 | 4.60 | Control | 2.60 |

Results, presented below in Tables 3 and 4, can be summarized as follows: we provide a rank order of cognitive aids as well the course section (in parentheses) in descending values of the performance metric of interest. For example, a quick review of Table 3 verifies that TRIZ intervention (undertaken by section

**Table 4** Novelty results (as summation of ideas with different novelty points)

| Section # | Case study #1 | Novelty screwdriver | Novelty wirecutter | Case study #2 | Novelty sonography |
|---|---|---|---|---|---|
| 1 | Sketching | 0.82 | 1.70 | TRIZ | 0.95 |
| 2 | Control | 0.58 | 0.57 | Sketching | 0.99 |
| 3 | TRIZ | 1.80 | 2.0 | Smart pen | 1.02 |
| 4 | Smart pen | 1.25 | 1.27 | Control | 0.75 |

"Methods") resulted in the highest quantity (5). These rankings show that across all interventions and sections, TRIZ and Smartpen interventions and section "Methods" and section "Results" seem to be better in comparison to others.

The following rankings compare each section (i.e., treatment) for each of the case studies.

**Quantity Value Result Ranking Comparison by Treatments:**
Case Study #1—Screwdriver
TRIZ (sec3) > Smartpen (sec4) > Sketching (sec1) > Control (sec2)
Case Study #1—Wire-cutter
Smartpen (sec4) > Control (sec2) > Sketching (sec1) > TRIZ (sec3)
Case Study #2
Smartpen > (sec3) Sketching (sec2) > Control (sec4) > TRIZ (sec1)
**Novelty Value Result Ranking Comparison by Treatments:**
Case Study #1—Screwdriver
TRIZ (sec3) > Smartpen (sec4) > Sketching (sec1) > Control (sec2)
Case Study #1—Wire-cutter
TRIZ (sec3) > Smartpen (sec4) > Sketching (sec1) > Control (sec2)
Case Study #2
Smartpen (sec3) > Sketching (sec2) > TRIZ (sec1) > Control (sec4)

Although these rankings provide easy to understand comparisons, convergence on both the interventions as well as the course sections makes it difficult to draw conclusions. In other words, it is not clear if the success observed in high performance of quantity and novelty is a result of the interventions (TRIZ and Smartpen) or the characteristics of the students in course sections "Methods" and "Results". Our relevant research questions can be expressed more explicitly as follows:

1. For all three conditions, which section performed the best in terms of quantity?
2. For all three conditions, which section performed the best in terms of novelty?
3. For all three conditions, which treatment (ideation method) performed best in terms of quantity?
4. For all three conditions, which treatment performed the best in terms of novelty?

To select the best performing course section and cognitive aid from the results on the three different design problems we use the Borda count selection process [27]. In Table 5, the first three columns from left show the quantity and novelty values

**Table 5** Borda Counts for sections and treatments

| Section # | Quality borda count | Novelty borda count | Treatment | Quality borda count | Novelty borda count |
|---|---|---|---|---|---|
| 1 | 1 | 0 | Sketching | 4 | 4 |
| 2 | 4 | 2 | Control | 3 | 0 |
| 3 | 6 | 9 | TRIZ | 3 | 7 |
| 4 | 6 | 4 | Smart pen | 8 | 7 |

for each section, and the rest of the columns to the right show the quantity and novelty values per treatment. The Borda count process helps select the best option out of a ranked ordered set of options by giving ascending weights (starting with $n - 1$ to 0) across cases. For example, for the screwdriver case study ranking of the treatments under the quantity metric, TRIZ (also sec3) gets a weight of 3, Smartpen gets 2, Sketching gets 1, and Control gets 0 as a weight (or a multiplier). Instances of the same treatment (e.g., TRIZ) are then summated across rankings with the appropriate weights.

A review of Table 5 shows that sections "Methods" and "Results" fared better compared the other two in terms of quantity, and TRIZ and Smartpen treatments were better in terms of novelty.

Borda counts, however, cannot explain if either section grouping or treatments are statistically significant in their effect on the performance measures (quantity and novelty). Accordingly, we proceeded with further statistical analysis of the data.

We have investigated the significance of these main effects using general linear models (GLM) where we have taken the course grade and the student count per team as co-variates. It was considered that the overall course grade might reflect students' ability, experience, and overall motivation in completing these design tasks, and we also wanted to ensure that our results were not confounded due to number of students in teams. Although most teams were 4 person teams, we had a few 3-person teams, and one 5-person team. Two GLMs, solved once for novelty and once for quality are shown below.

As it can be observed below, indeed, treatment is found to be a significant ($p = 0.048 < 0.05$) factor for its impact on novelty, and among the treatment options TRIZ seems to induce the highest values for novelty (see Fig. 1). A similar analysis was done for quantity (see Fig. 2); in this case, however, none of the main effects were significant. Among the treatment options, Smartpen seemed to produce the highest quantity in the generated ideas.

**Fig. 1** Mean novelty values for treatment alternatives

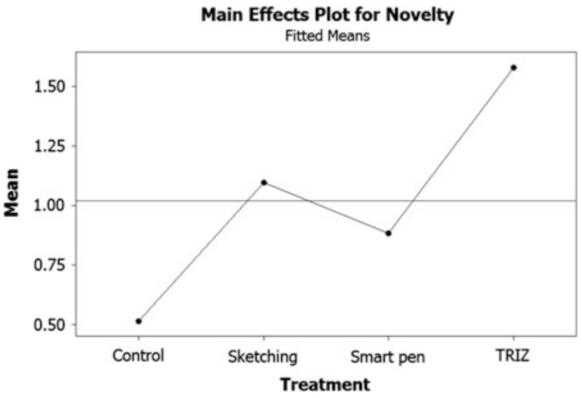
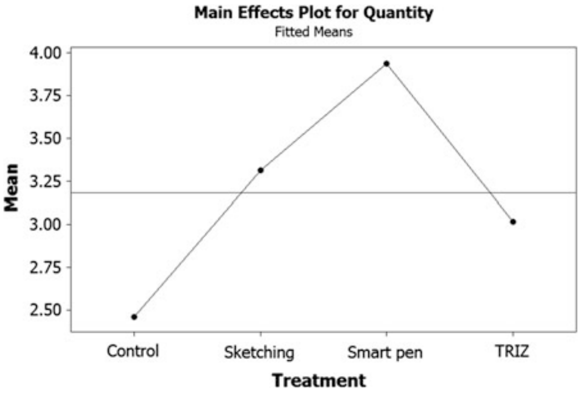


**Fig. 2** Mean quantity values for treatment alternatives



**General linear model: novelty versus treatment**

```
Factor          Type    Levels     Values
Treatment fixed 4 Control, Sketching, Smart pen, TRIZ
Analysis of Variance for Novelty, using Adjusted SS for Tests
Source          DF      Seq SS     Adj SS     Adj MS    F       P
Section         1       1.2508     1.6380     1.6380    3.04    0.089
Student/team    1       0.0186     0.4126     0.4126    0.77    0.387
Course grade    1       0.0401     0.0696     0.0696    0.13    0.721
Treatment       3       4.6370     4.6370     1.5457    2.87    0.048
Error           43      23.1879    23.1879    0.5393
Total           49      29.1344
```

Based on these presented, we assert that indeed ideation treatments have been found to impact design creativity outcomes, more specifically for our case: quantity and novelty.

**General linear model: quantity versus treatment**

```
Factor           Type      Levels     Values
Treatment fixed 4 Control, Sketching, Smart pen, TRIZ
Analysis of Variance for Quantity, using Adjusted SS for Tests
Source           DF        Seq SS     Adj SS     Adj MS     F         P
Section          1         14.695     2.727      2.727      1.32      0.257
Student/team     11.030    0.254      0.254      0.12       0.728
Course grade     1         0.224      0.020      0.020      0.01      0.922
Treatment        3         13.212     13.212     4.404      2.13      0.110
Error            43        88.839     88.839     2.066
Total            49        118.000
```

## Discussion

Various researchers study the impact on creativity in engineering design as measured by assessing the number and novelty of the output given to a specific design problem. Empirical studies are done most often using students and factorial designs that attempt to isolate the impact of a single treatment on the designing task results. When sufficient numbers of students are available a control group will be included in the experimental design. Having a control group allows the results of different treatment interventions on students with the underlying assumption that the student groups all have the same base-level experience, aptitude and motivation to perform the designing tasks. This is not automatically the case.

This work is different from the other creativity studies cited in that the reported results came from introducing the same interventions on each group of students in a series of treatments (e.g., section "Methods" used TRIZ to solve cases 1 and 2). The reported results for each intervention are from sections of the same students working in the same groups. Thus, the cumulative impact of personality and motivation are eliminated as sources of variation within each group and each ± ion and among treatment results for the same groups. There may still be differences between sections, although as the number of students per section increases, the differences will tend to diminish.

The Borda count was used as a simple indicator to select the treatments and sections that performed the best in terms of quantity and novelty of design results. The Borda count was able to identify the best (highest ranking) sections and treatments for each performance measure. The advantage of the Borda count is that it is simple to use, not requiring any statistical calculations. Naturally, then, no statistically significant conclusions can be drawn using the Borda count.

The results found by the Borda count were verified and supplemented by the GLM analysis. This analysis identified the treatment to be statistically significant in producing different results than the other factors. The GLM analysis showed that the section and group effects were not statistically significant in the presence of the interventions. Furthermore, it was shown that grades were not significant in describing the difference in results. Interestingly, the section effect was significant at a p-value of 10 % on novelty.

One final observation is that the results for the sketching and Smartpen groups were very similar on novelty but the Smartpen produced higher quantity (numbers of ideas) in the results than the other treatments. The authors hypothesize that this reflects a positive bias on the part of students toward playing with new technology.

## Conclusion

The experiments reported here indicate that the use of TRIZ aids student groups in the design tasks by improving the quantity and novelty of the ideas generated in two case studies over the control groups. The use of sketching, with and without the Smartpen technology improved the number (quantity) of ideas generated. These conclusions are limited to the experimental scope; nevertheless, these are relevant to improve our understanding of key cognitive aids for creativity. The importance of accounting for variation in results due to subject personality and motivation factors is discussed.

## Appendix A: Samples for Ideas Generated

(Fig. 3)
(Fig. 4)
(Fig. 5)
(Fig. 6)
(Fig. 7)

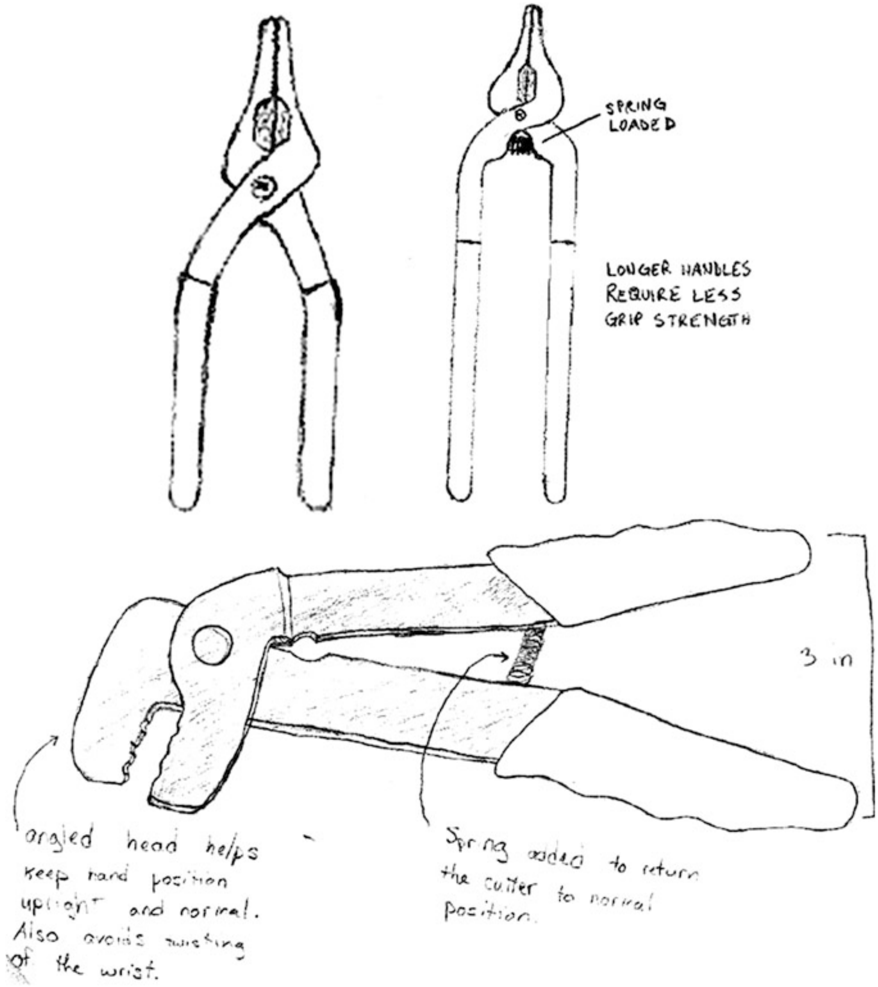


**Fig. 3** Set 1 old pliers (*Less Torque*) new pliers (*More Torque*) and wire cutter redesign

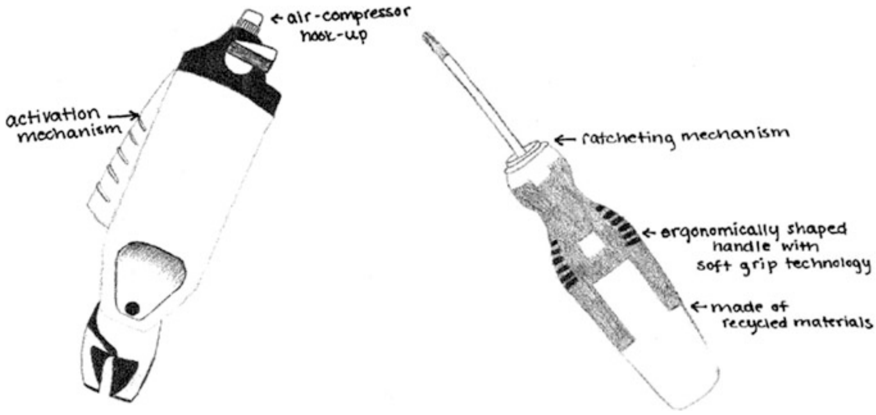


**Fig. 4** Set 2 compressed activated pliers and ratcheting screwdriver

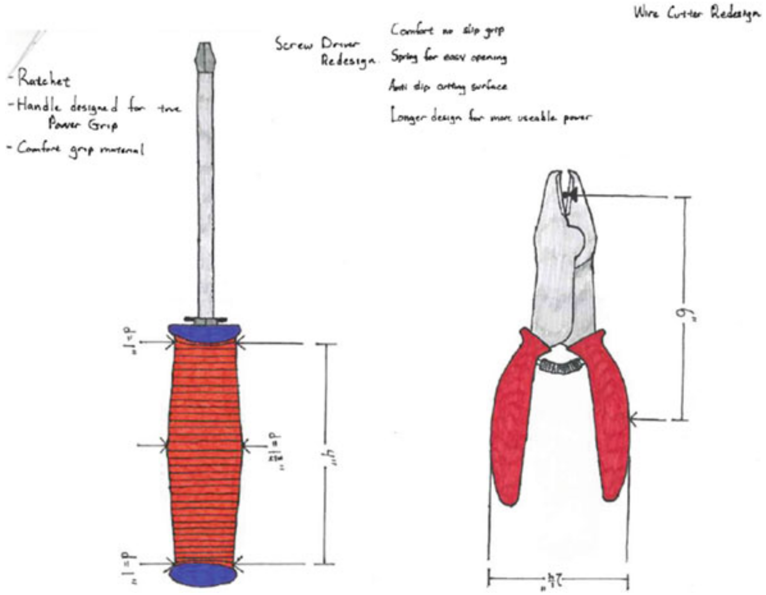


**Fig. 5** Set 3 screw driver redesign and wire cutter redesign

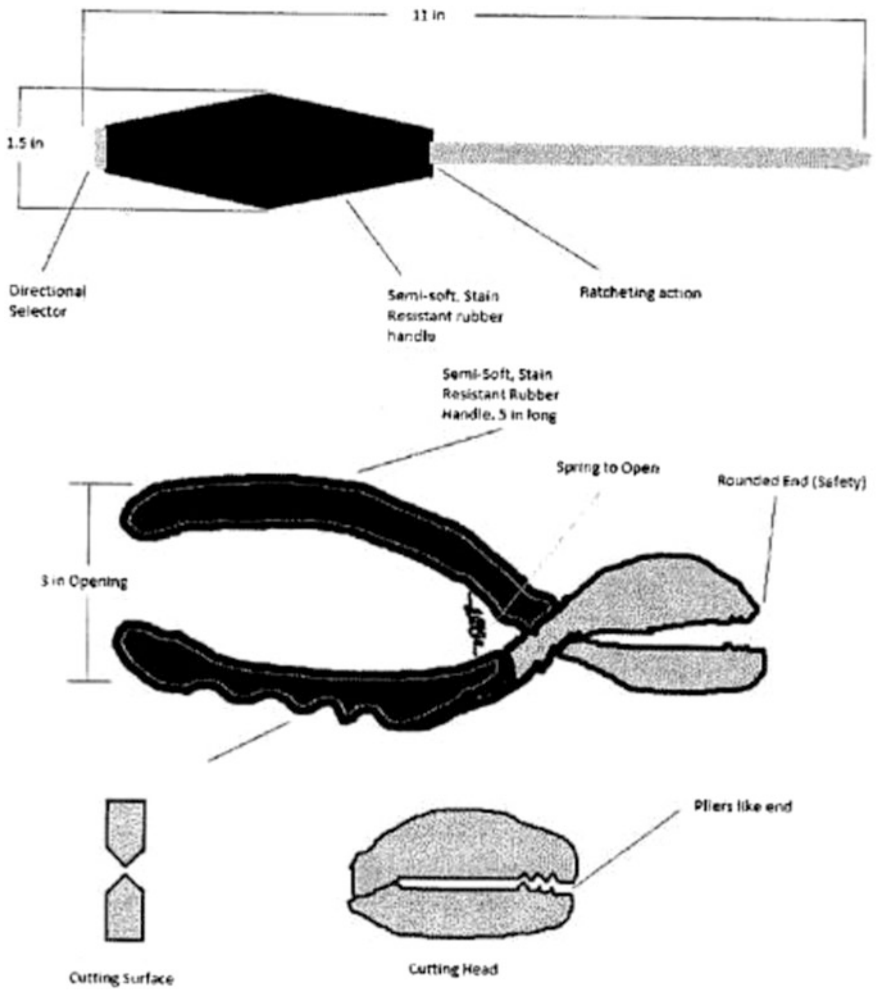


**Fig. 6** Set 4 screwdriver redesign and wire cutter redesign

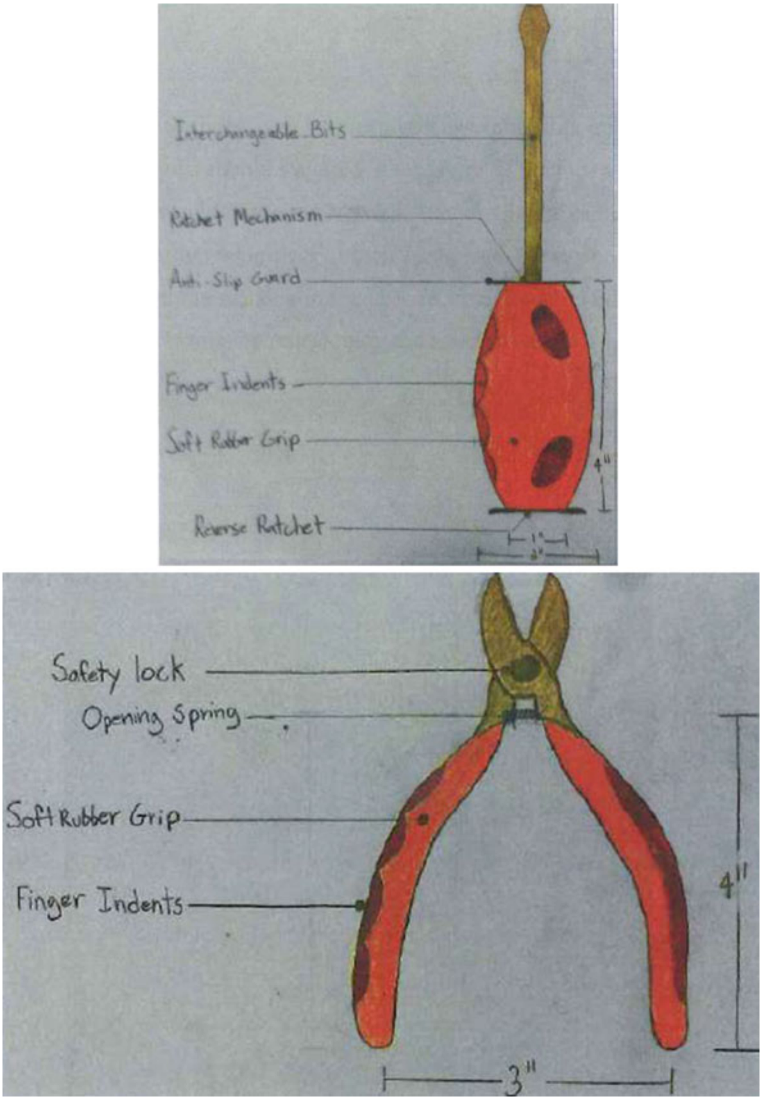


**Fig. 7** Set 5 screw driver redesign wire cutter redesign

## References

1. Goller I, Kobe C (2008) Developing a team creativity training for engineers. In: Proceedings of 9th CINet conference 2008, Valencia, 7–9 Sept 2008
2. Court AW (1998) Improving creativity in engineering design education. Eur J Eng Educ 23(2):141–154
3. Okudan GE, Vargas Hernandez N, Schmidt L, Linsey J, Chiu M-C, Lin C-Y (2010) A pilot exploration of systematic ideation methods and tools on design learning. In: ITHET'10

proceedings of the 9th international conference on information technology based higher education and training

4. Rodriguez SM, Gonzalez Macias R, Mendoza M, Vargas Hernandez N, Okudan GE, Schmidt L (2011) Guidelines for engineering design creativity: design of experiments. In: Proceedings of the international design engineering technical conferences paper no. DETC2011-48402, Washington, DC, 28–31 Aug, 2011
5. McCormick D (2007) Seeing mechanical. Mechanical engineering. American Society of Mechanical Engineers, New York, NY, pp 35–36
6. Shah JJ, Vargas-Hernandez N, Summers JD, Kulkarni S (2001) Collaborative sketching (C-Sketch)—an idea generation technique for engineering design. J Creative Behav 35(3):168–198
7. Bilda Z, Gero JS, Purcell T (2006) To sketch or not to sketch? That is the question. Des Stud 27(5):587–613
8. Altshuller G, Shulyak L (2002) 40 principles: TRIZ keys to technical innovation. Technical Innovation Center Inc, Worcester, MA
9. Ogot M, Okudan GE (2006) Integrating systematic creativity into first-year engineering design curriculum. Int J Eng Educ 22(1):109–115
10. Shirwaiker RA, Okudan GE (2008) TRIZ and axiomatic design: a review of case-studies and a proposed synergistic use. J Intell Manuf 19(1):33–47
11. Shirwaiker RA, Okudan GE (2007) Using TRIZ and axiomatic design synergistically: a case study. Int J Innov Technol Manage (IJITM) 4(2):155–170
12. Okudan Kremer GE, Chiu MC, Lin C-Y, Gupta S, Claudio D, Thevenot H (2012) Application of axiomatic design, TRIZ and mixed integer programming to develop innovative designs: a locomotive ballast arrangement case study. Int J Adv Manuf Technol (in press, DOI 10.1007/soo170-011-3752-1)
13. Stavridou A, Furnham A (1996) The relationship between psychoticism, trait-creativity and the attentional mechanism of cognitive inhibition. Pers Individ Differ 21:143–153
14. Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem-solution. Des Stud 22(5):425–437
15. Shah J, Vargas-Hernandez N, Smith S (2003) Metrics for measuring ideation effectiveness. Des Stud 24(2):111–134
16. Lopez R, Linsey JS, Smith SM (2011) Characterizing the effect of domain distance in design-by-analogy. In: Proceedings of the ASEM 2011 international
17. Vargas Hernandez N, Kremer G, Linsey J, Schmidt L (2010) Systematic ideation curriculum effectiveness investigation & deployment to enhance design learning. In: Proceedings of the 2010 American society for engineering education annual conference and exposition, Louisville, Kentucky, 20–23 June, 2010
18. Yang MC (2009) Observations on concept generation and sketching in engineering design. Res Eng Des 20(1):1–11
19. Linsey JS, Clauss EF, Kurtoglu T, Murphy JT, Wood KL, Markham AB, (2010) An experimental study of group ideation generation techniques: understanding the roles of idea representation and viewing methods. J Mech Des, vol 133, Mar 2011
20. Chan J, Fu K, Schunn C, Cagan J, Wood K, Kotovsky K (2011) On the benefits and pitfalls of analogies for innovative design: ideation performance based on analogical distance, commonness, and modality of examples. In: Chen W, Lewis K, Schmidt LC (eds) Decision making in engineering design, vol 133. Aug 2011, ASME Press, Two Park Avenue 2006 (Journal of Mechanical Design)
21. White C, Talley A, Jensen D, Wood K, Szmerekovsky A, Crawford R (2010) From brainstorming to C-sketch to principles of historical innovators: ideation techniques to enhance student creativity. In: Proceedings of the 2010 American society for engineering education annual conference and exposition, Louisville, Kentucky, 20–23 June, 2010
22. Purcell AT, Gero JS (1998) Drawings and the design process: a review of protocol studies in design and other disciplines and related research in cognitive psychology. Des Stud 19(4):389–430

23. Cardella ME, Atman CJ, Adams RS (2006) Mapping between design activities and external representations for engineering student designers. Des Stud 27(1):5–24
24. Westmoreland SN, Grenier A, Schmidt L (2008) Analysis of capstone design reports: visual representations. In: Proceedings of the ASME 2008 international design engineering technical conferences & computers and information in engineering conference, IDETC/CIE 2008, New York, 3–6 Aug, 2008
25. Bilda Z, Gero JS (2007) The impact of working memory limitations on the design process during conceptualization. Des Stud 28:343–367
26. Goel V (1995) Sketches of thought. MIT Press, Cambridge, Massachusetts
27. Chen W, Lewis K, Schmidt LC (eds) (2006) Decision making in engineering design, ASME Press, Two Park Avenue
28. Cardoso C, Badke-Schaub P (2011) The influence of different pictorial representations during idea generation. J Creative Behav 45(2):130–146

# A Pilot Protocol Study on How Designers Construct Function Structures in Novel Design

**Chiradeep Sen and Joshua D. Summers**

**Abstract** This paper reports a pilot protocol study that examines how designers construct function models as they develop and explore solution architectures for novel design problems. The purpose of this pilot project is to establish the experiment method and analysis protocol so that a repeatable and statistically large pool of participants can be used to draw significant conclusions about function model construction. In the study, voluntary participants with varied levels of experience in product design and function modeling are given a novel design problem and asked to develop functional architectures as part of concept development, using function structures as the modeling tool/language. The modeling actions are videotaped and the designers are interviewed using a predesigned questionnaire after the experiment. The data is analyzed using a predefined protocol that encodes the addition, deletion, and modification of model elements such as functions, flows, and text, and also actions such as reading the problem statement and pausing. The protocol analysis reveals patterns of modeling activities, such as forward chaining (expanding the model by adding functions to the head of flows and flows outgoing of functions), backward chaining (adding functions to the tail of flows and adding flows incoming to functions), and nucleation (starting with a few disconnected functions and flows, and gradually connecting them to complete the model). In aggregate, these observations provide insight into designers' thinking patterns while exploring solutions to unseen problems using function structures. The protocol is demonstrated to be complete within the scope of the study. The preliminary findings based on the two participants indicate that various parameters of solution exploration may largely vary between designers. The overall approach of model expansion also varies between forward chaining and

---

C. Sen
Oregon State University, Corvallis, OR, USA

J. D. Summers (✉)
Clemson University, Clemson, SC, USA
e-mail: jsummer@clemson.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_14,

nucleation. However, at a finer resolution of observing modeling actions, designers generally prefer nucleation or forward chaining of functions and forward or backward chaining of flows.

## Introduction

The purpose of the reported experiment is to establish an experiment method and analysis protocol to examine how mechanical product designers use function structure graphs to develop and explore solution architectures to novel design problems. This understanding can be useful in developing new design enablers, new approaches to early stage reasoning and analysis systems, and new methods for teaching students how to more effectively execute the design process. Function models are a popularly discussed product model used in early design to decompose the problem [1, 2], explore the solutions space [2, 3], search solutions [4], generate concepts [5, 6], and to archive design knowledge [7]. Many representations of mechanical functions are proposed in previous research, both in artificial intelligence [8–12] and in engineering design [1, 2]. Typically, these models are designed to support specific reasoning tasks. Specifically, function structure graphs and its extensions have been used to develop tools for concept generation [4–6], product similarity analysis [13], and failure modeling [14, 15] and failure propagation [16]. These graphs have also been used extensively to model the functionality of reverse engineered products [7] that are archived in a web-based data base named the Design Repository.[1]

Objective observation of function modeling by designers could help identify modeling action patterns, such as forward chaining, backward chaining, or nucleation, the type of stimuli and their use by the designers, and the type of information they prefer to capture for exploring solutions to new problems. This could in turn help identify important cognitive and behavioral trends about solution exploration. It could also provide insight to the design of formal representations of functions to enable early computer-aided design tools. For example, if designers tend to identify functions of a device before identifying the flows, a function-focused representation may be suitable, while if designers think of the device mainly in terms of its flows, then a flow-based representation may be more useful. Recent research proposed a formal representation of functions and showed that it could support early automated reasoning using the balance laws of mass and energy [17]. This experiment could reveal patterns of user-model interactions that could identify requirements for the user interface of such computer tools. The protocol study, described next, is an exploratory experiment designed with these anticipations.

---

[1] However, how designers interact with these models to develop and explore solutions to new, previously unaddressed problems has not been objectively examined http://repository.designengineeringlab.org/, accessed on Feb 23, 2011.

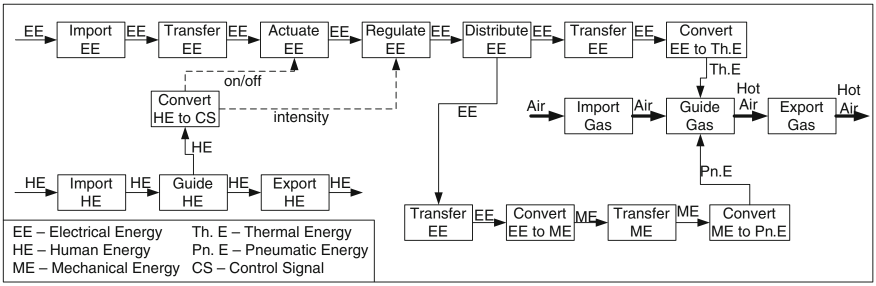


**Fig. 1** Function structure graph for a hairdryer product stored in the design repository

## Function Structures as an Early Design Representation

Functionality of mechanical devices is modeled in many different representations. Notable representations include The Function-Behavior-Structure model [18], the situated FBS model [19], the Function-Behavior-State model [20], a design pattern-based view [21], the Structure-Behavior-Function model [9, 22, 23], and the Functional Representation [10, 24]. This paper specifically focuses on user interaction with the graph-based Function Structure representation and this model is discussed in further detail. The use of function structures to model to support early design tasks is popularly recommended in design texts [1–3] and also widely explored in design research [4, 6, 13, 15, 25]. A function structure is a graph-based model of a product, where the nodes are transformative actions and the edges are the flows of material, energy, and signal undergoing transformations. Figure 1 shows the function structure for a hairdryer obtained from a web-based archive of reverse engineered product design information, called the Design Repository[1] [7, 26]. The flow name abbreviations are clarified in the legend.

This representation is described as a means to decompose a design problem [1, 2], search solutions [4], generate concepts [5, 6], and to archive design knowledge [7]. In order to enforce consistency of terms used in these models, many controlled vocabularies of function verbs and flow nouns are developed [27–30]. A notable example is the Functional Basis, developed by examining consumer products through systematic reverse engineering [2] and cataloging their functions and flows in a three-level hierarchical vocabulary. The Design Repository is an archive of these reverse engineered products and is currently expanding to support other research needs. The information stored in the repository, especially those related to function and failure of a product are used to develop early design tools for concept generation [4–6], product similarity analysis [13], failure modeling [14, 15], and failure propagation [16]. While function structures are widely used, especially to model reverse engineered products, the actual process of how these models are constructed and can be used by a designer to solve novel design problems (forward design) has not been objectively and systematically studied through experiments. The experiment described next is only a first step in this direction and describes results for only two participants. In recent future, this approach will be applied to run more experiments of this type through collaborative research.

## Design of the Experiment

In the experiment, voluntary participants with a variety of experiences in product design and function modeling are given a new product design problem and asked to develop and explore solution architectures using the function structure graphs. Each modeling session is videotaped and the designer interviewed subsequently. The video data is encoded using a predefined protocol that captures the addition, deletion, and modification of model elements such as functions, flows, and texts, and also actions such as reading the problem statement and pausing.

Three basic types of model expansion sequences are examined in this experiment: nucleation, forward chaining, and backward chaining. Nucleation of a function is an action when a function is inserted into the model without attaching it to an existing flow, indicating that the designers has identified a need for a function by its purpose or its name, but has not yet determined its transformative actions. Similarly, nucleation of a flow indicates that the designer identified the need for the flow's presence in the model, but has not indicated how it is going to be used.

Forward chaining is a sequence when the model is expanded by adding outgoing flows to functions and adding recipient functions to the head of those flows, indicating that the designer has already identified the functional transformation process steps leading to a means (eventually modeled at the head of the chain).

Backward chaining is a sequence, where the model is expanded by adding input flows to a function and producer functions to the tail of those flows, indicating that the designer has first determined the outcome (the head function or flow of the chain) and is exploring how that outcome can be obtained through transformations.

A given modeling session is not expected to be governed by one strategy alone, rather, an appropriate mix and variation between these three sequences is expected. Typically, nucleated sub-functions can be joined by both forward and backward chaining, until neighboring nuclei are connected through flows to complete the function structure.

## Participant Selection

The participants are volunteer graduate students of a design research lab with comparable educational background and training in design theory and methodology. Prior to selection, they are given a survey to collect data about their familiarity with function modeling and product design. For the first, the number of years of using and constructing function models in forward design and reverse engineering are collected, along with the number of unique products modeled or designed and the number of design projects done in different capacities (e.g., designer, leader, advisor). For the second, the volunteer is asked to list products under five categories—consumer appliances, shop tools, machinery and mechanisms, electronics, and toys—that he or she has used, studied (taken apart), or designed. Based on this survey, two participants are chosen for the study. The first participant, P1, has approximately 10 years of experience with mechanical product design as both designer and team leader and 4 years of experience in function modeling, while the second participant, P2, has approximately 2 years of experience in both areas. As a next step, this experiment will be repeated using more participants from collaborating research labs, in order to increase the replication and credibility of the data. However, this extension is out of scope for this current paper.

## Design Problem and Task

In order to ensure that the designers actually face a novel design problem and could still develop models for the solution, the design product must be new to the designer, yet one whose principles are familiar in real life. Accordingly, the following problem is developed.

> Design an automatic clothes-ironing machine for use in hotels. The purpose of the device is to press wrinkled clothes as obtained from clothes dryers and fold them suitably for the garment type. You are free to choose the degree of automation. At this stage of the project, there is no restriction on the types and quantity of resources consumed or emitted. However, an estimated 5 min per garment is desirable.

While an ironing machine is unfamiliar to the participants, ironing is a fairly well-understood activity, which enables the participants to intelligently ideate the functions. The design instructions ask the participant to develop and explore functional architectures for the design concept using function structure graphs as the medium of expression. The participants are given a marked area of a dry-erase board in the research lab—their familiar work environment—on which to construct their models.

## The Protocol Coding Scheme

Before collecting data through the experiments, a scheme to encode the collected data is developed. This coding scheme ensures that (1) all personal information such as handwriting are not accessible during data analysis, (2) the methods and algorithms for analyzing the data are committed to before data collection, so that the analysis is not biased by observations made at run-time of the experiments, and (3) all information used in analysis, such that the model itself and its construction process, could be completely re-enacted entirely based on the encoded information,

**Table 1** List of element types and their codes

| Element | Code | Definition |
|---|---|---|
| Block | B | A rectangle typically used to represent a mechanical function in the model. Incomplete definitions such as rounded edges or open corners will be included |
| Block text | BT | Text written within a block, indicating the name or description of the block (typically mechanical function) |
| Edge | E | An arrow, including its stem and its head, attached to a block or not, typically use to represent flows in the model |
| Edge text | ET | Text written above, below, or beside an edge, indicating a name or description of the flow |
| Source | SC | A circle or other shape, indicating the source of a flow or flows that are not originating from a rectangle (function) |
| Sink | SK | A circle or other shape, indicating the terminus of a flow or flows that are not terminating to a rectangle (function) |
| Note | N | A textual or symbolic expression that is not an ET or a BT |
| Symbol | S | A graphical expression (such as an arrow, a highlighting on existing text (e.g., underlines, encircling, or a punctuation mark) |
| Symbol text | ST | Text used to annotate a symbol, such as text written beside an arrow that is not an E |
| Diamond | D | Diamond-shaped boxes in the graph-based part of the model, typically used to represent a decision point |
| Diamond text | DT | Text written inside a diamond |
| Edge head | EH | The head of an edge, drawn at least one pregnant pause or more time lapse after drawing the stem |

without referring to the raw data. These requirements are imposed on the coding scheme in order to ensure objective, unbiased analysis. As the purpose of the experiment is to study designers' modeling actions, a list of atomic model elements and modeling actions are identified for monitoring through the code. Objective instructions for coding the raw data are also developed, as discussed next.

## *Atomic Model Elements*

A pilot study was conducted to identify a set of modeling elements a designer may use when modeling the ironing machine (see Table 1). For objective identification of these elements, their definitions are also developed. The code column indicates abbreviations to be entered during encoding the data.

## *Atomic Modeling Actions*

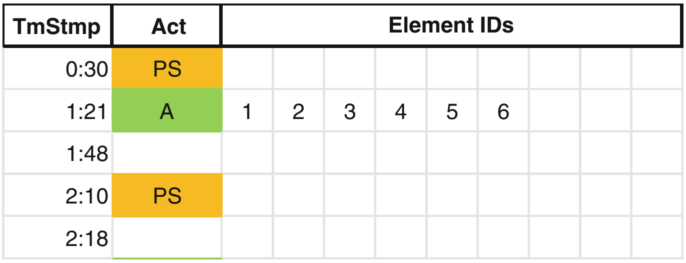
Based on the pilot study, five different activities and their definitions are identified (see Table 2).

**Table 2** List of activities and their codes

| Activity | Code | Definition |
|---|---|---|
| Add element | A | To insert an element in the model<br>Incomplete construction will be counted in when other elements are added to the model subsequent to inserting the partial element, since it constitutes "using" the partial element in the model. If the partial element is deleted after construction without any other activity other than a pregnant pause, its insertion will not be counted |
| Delete element | D | To remove an element from the model by erasing its graphics<br>Partial deletion will be considered a full deletion when the remainder of the element is subsequently not used in the model<br>Deletion of multiple elements without a pregnant pause (see below) is considered one deletion activity |
| Edit element | E | To make changes to an existing element<br>For blocks, sources and sinks it includes resizing. For arrows, it includes changing origin or destination, rerouting, and resizing. Edit operations performed by first erasing and then adding a redefined geometry will be counted as an edit only when there is no pregnant pause in between the two actions. Otherwise, count them as D and A. If a source is converted into a sink, or vice versa, by redirecting the arrows, the operation will be counted as the deletion of the source followed by the insertion of the sink. Overwriting and strikethrough count as editing<br>Editing multiple elements without a pregnant pause (see below) is considered one editing activity |
| Read problem stmt. | PS | To interact with the given design problem and instructions, while not doing A, D, or E. All actions such as pointing, emphasizing (underlining, highlighting), writing on the document, erasing from the document, or soliloquies during reading the document count as PS |
| Pregnant pause | Blank | To take a pause in between A, D, E, or PS for one second or longer |

## *Step-by-Step Instruction for Encoding of Video Data*

Encoding is completed in three steps: activity encoding, element encoding, and topology encoding. The first step is a time study of the sequence of different activities through the modelling session. The second is a cataloging of the model elements created through the sessions, while the last is an accounting of the modeling process, which accounts for how each element was connected to the surrounding elements at creation-time. Collectively, these three steps encode the entire modeling process, such that both the model and the modeling process can be re-enacted purely from the encoded information. These three steps are explained next.



| TmStmp | Act | Element IDs | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0:30 | PS | | | | | | | | | |
| 1:21 | A | 1 | 2 | 3 | 4 | 5 | 6 | | | |
| 1:48 | | | | | | | | | | |
| 2:10 | PS | | | | | | | | | |
| 2:18 | | | | | | | | | | |

**Fig. 2** Activity encoding for participant P1

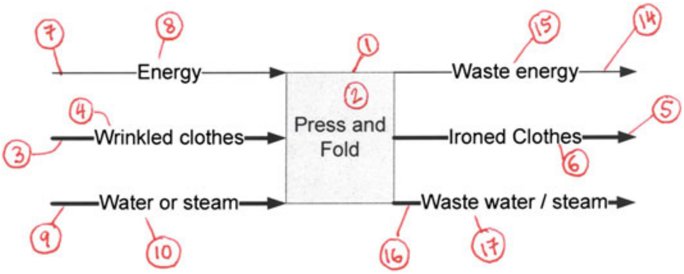


**Fig. 3** Assigning IDs to elements for P1

## Step 1: Activity Encoding

*Needed*:

1. The video file and a video-player software
2. The “Activity Encoding” spreadsheet (see Fig. 2)
3. The final resulting concept model from the experiment, redrawn using software (see Fig. 3).

**Instructions**: Start the video. Look for the starting instant of each activity. Pause at the start of each activity, and enter the following data in the spreadsheet.

1. **TmStmp**: the start time stamp (mm:ss) from the video
2. **Act**: the activity code performed from Table 2.

**Element IDs**: a unique numeric ID (serial number) for the element operated on in that activity. In the printed photo, assign this unique ID to the element in the same order of their addition, deletion, or edition, using bubbles. An example data entry operation is shown in Fig. 2. The element IDs represent the order of adding the elements in the model.

In this case, the designer started reading the problem statement 30 s into the experiment, as indicated by 0:30 in the first column. At 1:21, he started adding elements and added six elements (1 through 6). This took him up to 1:48 on the clock, at which point he started a pregnant pause. At 2:10, he started reading the

**Fig. 4** Element encoding for participant P1

| Elem ID | Elem Typ |
|---|---|
| 1 | B |
| 2 | BT |
| 3 | E |
| 4 | ET |
| 5 | E |
| 6 | ET |
| 7 | E |

**Fig. 5** Topology encoding for participant P1

| Elem ID | Elem Typ | Topology | |
|---|---|---|---|
| 1 | B | 0 | 0 |
| 2 | BT | 1 | |
| 3 | E | 0 | 1 |
| 4 | ET | 3 | |
| 5 | E | 2 | 0 |
| 6 | ET | 5 | |
| 7 | E | 0 | 1 |

problem statement again. At 2:18, he stopped reading the problem statement, yet was not doing A, D, or E, and therefore, the activity starting at 2:18 is recorded as a pregnant pause. After entering the activity codes 1–6 into the columns, the redrawn image of the final model is annotated by assigning these IDs to the elements by bubbles, in the same order of their addition, as seen in **Fig.** 3.

### Step 2 **Element Encoding**

*Needed*:

1. Completion of activity encoding
2. The Element Encoding spreadsheet (see Fig. 4)
3. The redrawn final model, with element numbers.

**Instructions**: For each element, enter the following information as shown in Fig. 4.

1. **Elem ID**: Write the unique numeric ID of the elements from the photo.
2. **Elem Typ**: Write the element type code from Table 1.

### Step 3 **Topology Encoding**

*Needed*:

1. Completion of element encoding
2. The video from the experiment
3. The Element Encoding spreadsheet.

**Instruction**: For each element in the spreadsheet, enter the following information under Topology, as shown in Fig. 5.

1. **Left column**: Enter the ID for the origin or input element. For blocks, enter the ID of the already existing flow to whose head the block is added (zero if no input flow already existed). For edges, enter the ID of the already existing block from where the edge originates (zero if the tail is not associated to an already existing block). For BT and ET, enter the ID of the block or the edge to which the texts belong.
2. **Right column**: Enter the ID for the destination or output element. For blocks, enter the ID of the already existing flow to whose tail the block is added (zero if no output flow existed). For edges, enter the ID of the already existing block to where the edge terminates (zero if the head is not associated to an already existing block).

In Fig. 5, block 1 is the first element added to the model and hence has zeros for both input and output. Block text 2 is added to block 1. Edge 3 has a dangling tail, but terminates on to block 1, which already existed. Edge text 4 is added to edge 3. Edge 5 originates from existing block 1, but has a dangling head. Edge text 6 is added to edge 5. Finally, edge 7 has a dangling tail, but terminates on to block 1.

The pre-defined set of model elements, modeling actions, and the steps of activity-, element-, and topology-encoding constitute the coding scheme. For illustration, the results from an actual experiment run are used in Figs. 2, 3, 4 and 5. As noted earlier, the coding scheme is developed without using specific data points, before conducting the experiment.

Since element encoding captures each element's type and topology encoding captures the connectivity between them, the model can be entirely reconstructed from the information captured in the Element Encoding spreadsheet, except that the exact literal strings in the texts (BT, ET, DT), notes, and symbols cannot be reproduced. Further, since activity encoding captures the sequence of adding, editing, and deleting those elements, the clustering of those activities, and the pauses separating those clusters, the **model construction process** can also be fully re-enacted in due time from the information in the spreadsheet. This completeness gives the confidence that all information produced by the designer during the experiment, except the exact texts, notes, and symbols, are encoded in this coding scheme. Notably, since the analysis focuses only on how designers construct concept models rather than what those concepts specifically are, this missing information is not necessary for the analysis. Thus, all information necessary for analysis this study are captured in this coding scheme.

## Sample Raw Data for Illustration

Figure 6 shows the final model produced by participant P2 during the experiment. For legibility, the smaller of the two models (eleven functions) is shown. The model produced by P1 has seventeen functions. Both models include blocks (B), block texts (BT), edges (E), edge texts (ET), and notes (N), although P1 made

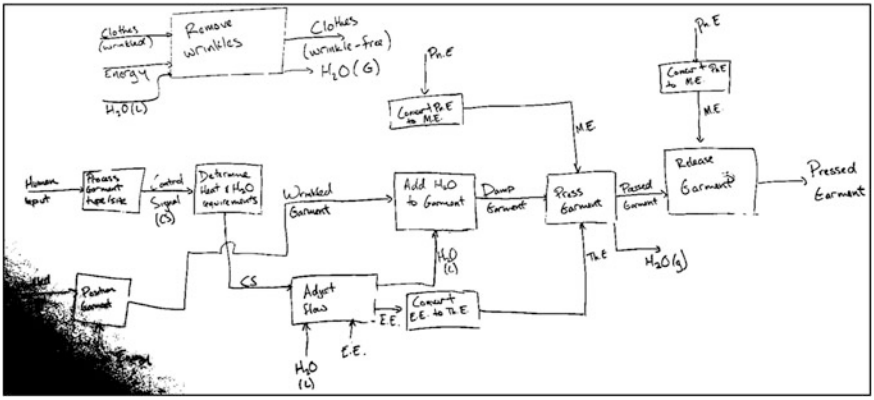


**Fig. 6** Sample final model produced by participant P2 (photo recolored for better legibility in print)

more use of notes than P1. Only P1 used sources (SC), sinks (SK), diamonds (D), and diamond texts (DT), while P2 left the edge ends coming from or going to the environment dangling, as seen in Fig. 6. The total number of nodes (function, sources, sinks, and diamonds) in P2's model was 58.

## Post-experiment Interview

After each experiment, both participants are interviewed using a questionnaire. It asks about the participants' familiarity with ironing and folding configurations of different garments, and also whether they were able to ideate the design purely in terms of its functions, or if they needed to think about the form of the device before translating it into a functional description. Both designers expressed that in the first few minutes they had difficulties separating form from function, although later they could think purely in terms of the device's required functions and flows. The duration of the modeling sessions between the participants were comparable (P1 = 20:20, P2 = 23:57 mm:ss).

## Observation, Analysis, and Implications

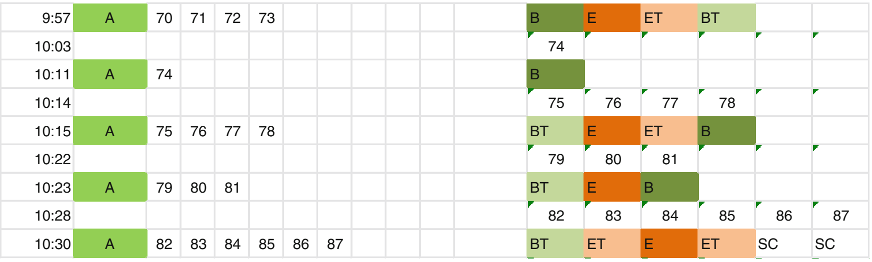
Once the video data is encoded using the coding scheme, it is presented in two views for analysis: activity encoding sheet and activity graph, which are discussed next. The pattern of modeling actions emerging from the experimental data indicates that the various parameters of solution exploration, such as modeling rate, number of edits, distribution of add, delete, and edit operations, and the duration and distribution of pauses between modeling actions may largely vary between designers. The overall approach of model expansion also varies between forward chaining and nucleation. However, at a finer resolution of observing modeling actions, designers generally show preference to nucleation or forward chaining of functions and forward or backward chaining of flows.

## Activity Encoding Sheet Analysis

A portion of the activity encoding sheet for P1 is shown in Fig. 7. The first column indicates the time stamp (mm:ss), the second column indicates actions (A = add, blank = pause), the next nine columns are element IDs, and finally, the last six columns show the element type for each element ID added. An inspection of this activity sheet reveals that for 46 of the 48 flows in the resulting model made by P1, the designer added the flow name (ET) immediately after adding the flow (E). Yet, for thirteen of the seventeen functions, the function name was added only after adding their attached flows, or at least after a pregnant pause. This trend is visible from the succession of the symbols “E” and “ET” in the shown portion, while the symbols “B” and “BT” are separated by the addition of other elements of pauses. This trend was also strongly visible in P2’s activity sheet, although for brevity, that diagram is not presented. Thus, the flow type or name was known to the designers as soon as a flow was conceived, indicating that these designers conceived the device in terms of the flows it would process, rather than in terms of functions it would perform. The function names were later retrofitted to describe the resulting transformative actions indicated by the flows.

In terms of modeling sequence, P1’s overall approach was nucleation, as he started the decomposed model with a few sub-functions on each end of the board, indicating the clearly identified sub-actions involved in ironing, and eventually finished the model by connecting those functions and others through edges in the middle of the board. P2, however, followed a generally forward chaining approach. The first function drawn was a subfunction on the left end of the board and the last function was the final action that produced folded pressed clothes.

However, when the modeling actions are observed at a finer time-resolution, booth designers seem to use nucleation and forward chaining when adding functions, and forward and backward chaining when adding flows. For example, P2 added six functions by nucleation, seven functions by forward chaining on the head of an existing flow, but only one function by backward chaining, on the tail of a flow. However, for the flows, P2 added sixteen and eleven flows by forward and backward chaining, and only two through nucleation, which were later appended with functions through forward chaining. These data are collected by analyzing the topology encoding. For example, in Fig. 5, edge 3 is added through backward chaining, as it originates in the environment (zero in left column under topology) and ends on a function (1). Edge 5 is added through forward chaining, as it



| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 9:57 | A | 70 | 71 | 72 | 73 | | | | | | | B | E | ET | BT | | |
| 10:03 | | | | | | | | | | | | 74 | | | | | |
| 10:11 | A | 74 | | | | | | | | | | B | | | | | |
| 10:14 | | | | | | | | | | | | 75 | 76 | 77 | 78 | | |
| 10:15 | A | 75 | 76 | 77 | 78 | | | | | | | BT | E | ET | B | | |
| 10:22 | | | | | | | | | | | | 79 | 80 | 81 | | | |
| 10:23 | A | 79 | 80 | 81 | | | | | | | | BT | E | B | | | |
| 10:28 | | | | | | | | | | | | 82 | 83 | 84 | 85 | 86 | 87 |
| 10:30 | A | 82 | 83 | 84 | 85 | 86 | 87 | | | | | BT | ET | E | ET | SC | SC |

**Fig. 7** A portion of the activity encoding sheet for P1

originates in block 2 and is not attached at head at creation-time. Block 1 is added through nucleation, as it has no existing edges at input or output at creation-time.

From the function structures produced, it was visible that both designers started with defining the overall functionality in a black box function, as seen in the top-left corner of Fig. 6. However, P1 finished drawing all the flows attached to the black box, including the energy and water input required for ironing and the emissions (see Fig. 3), before starting the decomposed model. By contrast, P2 repeatedly returned to add the two $H_2O$ flows to the black box (see Fig. 6) after starting to decompose the model. This could indicate that P2 was using the model to actually explore a solution more than P1, who committed an idea to the drawing only after developing it mentally. However, in the entire observed period, P1 read the problem statement only during black box construction and decomposed the model by referring back to the black box. By contrast, P2 repeatedly referred to both the problem statement and the black box while constructing the decomposed model. This indicates that P1 was more successful in translating the problem from natural language to the functional language before beginnings to explore solutions for it. However, since the quality of the model or the solution is not observed in this experiment, no conclusions should be drawn to whether such translation is conducive for better design. For the design of a future function-modeling tool, this observation shows that the software should allow saving and viewing other models such as the black box during constructing or decomposing a model.

## Activity Graph Analysis

One other view of the data is the activity graph, which tracks the rate of modeling activities against time. Figures 8 and 9 show the activity graphs for P1 and P2. The horizontal axis denotes time in hh:mm:ss format, while the vertical axis denotes the count of model elements added, deleted, and edited. The black bars above the axis are addition, the ones below the axis are deletion, and the grey bars above the axis denote edition of model elements. These graphs are used to make the following observations.

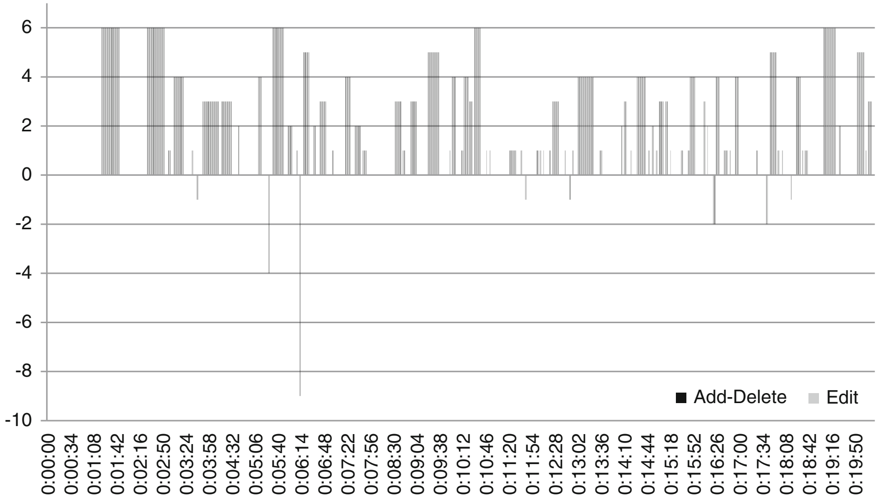


**Fig. 8** Activity graph for participant P1

P1 used function lists in form of notes (N) extensively at the beginning of decomposition (observed in the activity encoding sheet), immediately after developing the black-box model. During making these notes, P1 iterated thrice before settling with a decomposition approach, indicated by the frequent adding and deleting actions between the third and the sixth minutes on P1's activity graph (see Fig. 8). However, once he settled with an approach, his activity graph does not show any deletion until the later part of the process, which again indicates that this designer develops an idea mentally or using the list view, before including it to the model. For the software design, this may indicate that function listing can be useful for high-level architecture design, while for detailing each atomic action, a graph-based model is more suitable.

By contrast, P2 edited the elements more uniformly through the design session and frequently deleted large number of elements together, even only 3 min prior to finishing the design. The video, as well as the activity encoding sheet reveals that many of P2's edits were for renaming functions or rerouting flows to and from the functions. For example, the function "Add $H_2O$ to garment" was assigned four names, at steps no. 52, 61, 68, and 74, while four flows—wrinkled garment, $H_2O(L)$, ThE, and ME—were rerouted multiple times through the model. This indicates that the concept exploration approach may vary between designers largely and for some designers the software should allow simultaneous viewing and comparison of multiple model options.

The difference between the two designers' thinking patterns also appears through a few other observations. First, with one exception, P1 begins to edit the model only after significantly developing it up to the tenth minute. From then until the end of modeling, edits were frequent, and often lumped between the pauses. This indicates that P1 prefers to first model raw ideas without concern for their

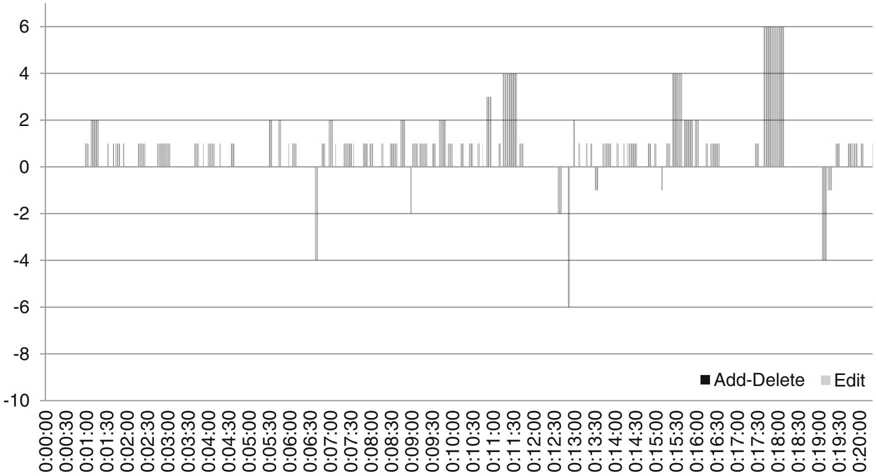


**Fig. 9** Activity graph for participant P2

compatibility, and considers that only later. By contrast, P2 edited the model more uniformly in time. Fourteen out of the 22 edit operations by P2 are done on elements immediately after adding them, separated only by a pause. This indicates that P2 considered model options as they were being included and revised. Also, the pauses, indicated by the blank spaces between the bars in Figs. 8 and 9, are more uniformly distributed between the actions in the case of P1, while P2 has displayed repeated long pauses.

Additionally, there is a significant difference between the modeling rates of the designers. During black-box modeling and the beginning of decomposition, P1's modeling rate was marked by large lumps of addition (up to six elements at a time without pause), large lumps of deletion (up to nine elements without pause), and long pregnant pauses between them (up to 27 s). These lumps and pauses became significantly smaller after P1 identified the list of sub-functions. Thus, for this designer, the listing indicates the point of commitment to design architecture. However, for P2, the heights of the bar-clusters that indicate the number of elements added together without pausing were shorter in the beginning and gradually increased. In fact, the tallest clusters for P1 are the first two, which were drawn during drawing the black box model, while for P2, the tallest clusters are close to the end of the design, in the eighteenth minute. This once again indicates that P2 used the model to explore and develop the ideas in the initial minutes and sped up when those ideas came together toward the end. P1, by contrast, first developed an idea and its possible decomposition mentally and using the lists, and then committed them to the model.

Finally, the maximum number of items added at a time (between pregnant pauses) is six for both designers. This indicates that the number of elements these designers could comfortably perceive is six and this may have a relation with the typical human cognitive chunk count of seven established in psychology research [31]. However, delete operations are done on chunk sizes of up to nine elements at a time. Between the two designers, P1 used more large chunks than P2. For example, P1 created five chunks of six elements, four chunks of five elements, and twelve chunks of four elements each, while P2 created one chunk of six elements and two chunks of four elements each. This observation may indicate a trend that more experienced designers can, perhaps, process larger cognitive chunks than novice designers (P1 has more design experience than P2), as discussed in previous research [3]. However, since only one participant of each experience level was used in this study, this could be a random personal trait that is not generalizable, yet suggests new avenues of investigation.

## Discussion and Path Forward

This paper reports an initial experiment that is a first step to study how designers construct and reason with function structure graphs when developing and exploring functional architectures for novel design problems. However, since only two participants are used in this pilot version, some observations cannot be generalized. A collaborative research effort is already underway between multiple design research labs to repeat this experiment with more participants, thus giving the required replication of data.

Nevertheless, this pilot experiment produced some interesting observations and hypotheses to be investigated in the future. First, the study shows that modeling strategies may widely vary between designers and identifies the experiment parameters, such as modeling rate, duration and distribution of actions such as addition, deletion, and edition of model elements can be used in the future studies to monitor them. It also produced a finite list of model elements and modeling actions that can be used in the future version of this experiment.

Further, the experiment shows that flows are more concrete concepts in function modeling and functions are more abstract. Designers identify flow names and types as soon as they add a flow, but function names are often identified much later than their addition, typically constructing all the flows attached to the block. This observation agrees with previous theoretical research that demonstrated that in function structure graphs, the topology (flow connectivity) of the model carries much more information than the functions [32] and may guide the development of the user interface features of future CAD tools fir function-based early design.

The experiment also demonstrates a means to monitor model expansion sequences, such as nucleation, forward chaining, and backward chaining. It shows that the 1 s resolution of time is adequately fine to capture these trends, which is a learning to be applied to the future experiments. The experiment also shows that while the overall modeling sequence varies between designers, at a finer resolution of observation, designers prefer to use only nucleation and forward chaining for functions and only forward and backward chaining for flows. In terms of a future software design for function-based automated reasoning, this observation may guide to the design of the internal data structure of formal function representations and also modeling options available through the graphic user interfaces.

Finally, this experiment shows many evidences of the behavioral differences between designers, in terms of modeling rate, decomposition strategies, and the use of notes and plans. It has also shown possible evidences that the modeling actions may be related to the cognitive chunk sizes of designers that may vary with designer experience. While these trends could not be verified due to small replication size, these trends can be used as hypotheses for extensions of this experiment in the future. As indicated earlier, more detailed version of this experiment are already underway in a larger collaborative effort, the results of which will be published in due time.

## References

1. Pahl G et al (2007) In: Wallace K, Blessing L (eds) Engineering design: a systematic approach, 3rd ed. Springer, London
2. Otto KN, Wood KL (2001) Product design techniques in reverse engineering and new product development. Prentice Hall, Upper Saddle River, NJ
3. Ullman DG (1992) The mechanical design process. McGraw-Hill, New York
4. Sridharan P, Campbell MI (2005) A study on the grammatical construction of function structures. Artif Intell Eng Des Anal Manuf 19(3):139–160
5. Kurtoglu T, Swantner A, Campbell MI (2010) Automating the conceptual design process: From black box to component selection. Artif Intell Eng Des Anal Manuf 24(1):49–62
6. Vucovich J et al (2006) Concept generation algorithms for repository-based early design. In: ASME 2006 international design engineering technical conferences and computers and information in engineering conference, Philadelphia, PA
7. Bohm MR, Stone RB, Szykman S (2005) Enhancing virtual product representations for advanced design repository systems. J Comput Inf Sci Eng 5(4):360–372
8. Gero JS, Kannengiesser U (2002) The situated function-behaviour-structure framework. In: Gero JS (ed) Artificial intelligence in design. Kluwer Academic Publishers, Norwell, pp 89–104
9. Goel AK, Bhatta SR (2004) Use of design patterns in analogy-based design. Adv Eng Inform 18:85–94
10. Chandrasekaran B, Josephson JR (2000) Function in device representation. Eng Comput 16(3–4):162–177
11. Umeda Y et al (1996) Supporting conceptual design based on the function-behavior-state modeler. Artif Intell Eng Des Anal Manuf 10(4):275–288
12. Vescovi M et al (1993) CFRL: a language for specifying the causal functionality of engineered devices. In: Eleventh national conference on artificial intelligence. American Association for Artificial Intelligence, Washington, DC
13. McAdams DA, Wood K (2002) A quantitative similarity metric for design-by-analogy. J Mech Des 124(2):173–182
14. Tumer IY, Stone RB (2001) Analytical methods to evaluate failure potential during high-risk component development. In: 2001 ASME design engineering technical conferences, Pittsburgh, PA
15. Arunajadai SG et al (2004) Failure mode identification through clustering analysis. Qual Reliab Eng Int 20:511–526

16. Kurtoglu T, Tumer IY (2008) A graph-based fault identification and propagation framework for functional design of complex systems. ASME J Mech Des 130:051401-1–051401-8
17. Summers JD, Mocko GM (2011) A protocol to formalize function verbs to support conservation-based model check reasoning. J Eng Des 22:765–788
18. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36
19. Bobrow DG (1984) Qualitative reasoning about physical systems: an introduction. Artif Intell 24(1–3):1–5
20. Umeda Y et al (1990) Function, behavior, and structure. In Gero JS (ed) Applications of artificial intelligence V, vol 1. Springer, Boston, pp 177–193
21. Bhatta SR, Goel AK (1997) A functional theory of design patterns. In: 15th International joint conference on artificial intelligence, vol 1, Morgan Kaufmann Publishers Inc., Nagoya
22. Bhatta S, Goel AK, Prabhakar S (1994) Innovation in analogical design: a model-based approach. In: Artificial intelligence in design. Kluwer Academic Publishers, Dordrecht
23. Goel A, Bhatta S, Stroulia E (1997) Kritik: an early case-based design system. In: Maher ML, Pu P (eds) Issues and applications of case-based reasoning in design. Erlbaum, Mahwah, pp 87–132
24. Chandrasekaran B (2005) Representing function: relating functional representation and functional modeling research streams. Artif Intell Eng Des Anal Manuf 19(2):65–74
25. Arunajadai SG, Stone RB, Tumer IY (2002) A framework for creating a function-based design tool for failure mode identification. In: ASME 2002 design engineering technical conferences and computers and information in engineering conference, Montreal, Canada
26. Bohm MR, Stone RB (2004) Representing functionality to support reuse: conceptual and supporting functions. In: ASME 2004 design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers, New York, Salt Lake City, pp 411–419
27. Collins JA, Busby HR, Staab GH (2010) Mechanical design of machine elements and machines, 2nd edn. Wiley, Hoboken, NJ
28. Szykman S, Racz JW, Sriram RD (1999) The representation of function in computer-based design. In: ASME design engineering technical conferences, Las Vegas, NV
29. Kirschman CF, Fadel GM (1998) Classifying functions for mechanical design. J Mech Des 120(3):475–482
30. Hirtz J et al (2002) A functional basis for engineering design: Reconciling and evolving previous efforts. Res Eng Design 13(2):65–82
31. Miller GA (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychol Rev 63:81–97
32. Sen C, Summers JD, Mocko GM (2010) Topological information content and expressiveness of function models in mechanical design. J Comput Inf Sci Eng 10(3):031003-1–031003-11

# Commonalities Across Designing: Empirical Results

**John S. Gero, Udo Kannengiesser and Morteza Pourmohamadi**

**Abstract** This paper presents empirical evidence of commonalities across designing that appear to be independent of the designers' geographical location, expertise, discipline, the specific design task, the size and composition of the design team, and the length of the design session. Our evidence is founded on thirteen highly heterogeneous design case studies that differ along these dimensions but exhibit some commonalities. We analysed the results from protocols of these case studies produced by a variety of researchers, using a method that is based on the FBS framework and is independent of any domain- or situation-specific parameter. We found commonalities across all thirteen case studies, related to the first occurrence of design issues in the design process, and to the continuity and the rate with which design issues are generated. Our findings provide preliminary support for the claim that designing can be studied as a distinct human activity that appears in different expressions but shares the same fundamental characteristics.

## Introduction

Designing is a complex activity that has attracted a significant amount of attention from different research domains, trying to demystify its manifold processes. One of the biggest challenges in this regard is to define designing as a unique activity

---

J. S. Gero · U. Kannengiesser (✉)
Krasnow Institute for Advanced Study, Fairfax, USA
e-mail: udo.kannengiesser@gmail.com

J. S. Gero
e-mail: john@johngero.com

M. Pourmohamadi
The University of Sydney, Sydney, Australia

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_15,

while it is used in a vast range of domains such as engineering, software, graphical interfaces, and electronics, to name a few. Understanding the commonalities amongst different expressions of designing is a prime step in developing a universal understanding of it [1–3]. Currently, interviews and protocol studies are two of the most credited and frequently used methods to study the behaviour of designers in solving different design problems. Despite their validity in obtaining insight into the thoughts of designers [4, 5], the ad-hoc dependency of these methods on the data has been a barrier for generalising the results of these studies across different designers, design situations and design researchers [6]. In addition, the complexity of designing per se makes aggregating the results of empirical studies in large, statistically significant scales a challenging task.

In this paper we use an approach to the analysis of multiple design protocols that allows studying their commonalities independently of any environmental parameter. It is based on the cumulative occurrence of design issues over the course of designing, coded according to the Function-Behaviour-Structure (FBS) design issue system. The results of applying this method indicate that there are significant commonalities across different instances of designing.

## Source Data: Thirteen Design Protocols

Our source data consists of thirteen segmented and coded design protocols produced by various research groups. These protocols differ from one another in multiple ways producing a highly heterogeneous data source. Table 1 presents the state space covered by the thirteen protocols, in terms of seven independent variables and their ranges of values.

As shown in Table 1, the protocols originate from four different continents and address a wide variety of design tasks. The participants include designers with different levels of expertise and with varying education and training in different disciplines. The team sizes vary from small teams of only two designers to larger teams of up to 9 designers. Some of the teams are homogeneous (consisting of designers with the same knowledge background), while others are heterogeneous (consisting of designers with different knowledge backgrounds). The lengths of the design sessions vary from 192 to 1,280 segments of the coded protocols.

Table 2 shows the specific characteristics of each of the thirteen design protocols.

The thirteen protocols were segmented and coded by nine different coder teams from various research groups. All coders used the same coding scheme based on the FBS framework [7, 8]. It consists of six design issues:

- *Requirements*: includes all requirements and constraints that were explicitly provided to the designers at the outset of the design task.

**Table 1** The state space covered by the thirteen design protocols

| Variable | Range of values |
|---|---|
| Source location of data | Australia, Singapore, Taiwan, UK, USA—five states: CA, IL, MN, UT, VA |
| Design task | Designing of:<br>• Assistive window raising device<br>• Assistive door opening device<br>• Novel thermal ink pen<br>• Software system to simulate road traffic controls<br>• Art gallery<br>• Teaching device<br>• Future personal entertainment system<br>• Coffee maker<br>• Pedometer to encourage running<br>• Commercial website |
| Participants' expertise | Professional designers, Undergraduate students, High school students |
| Participants' knowledge domain | Architecture, Business, Electronics, Ergonomics, Industrial design, Interface design, Mechanical Engineering, Mechatronics, Psychology, Software, Web design |
| Team size | From 2 to 9 designers |
| Team composition | Homogeneous, Heterogeneous |
| Length of design protocol (in number of segments) | From 192 to 1,280 segments |

- *Function*: includes teleological representations that can cover any expression related to potential purposes of the design. These representations may be flow-based or state-based [9].
- *Expected Behaviour*: includes attributes of the design used as assessment criteria or target values for potential design solutions. They may include technical, economic, ergonomic and other characteristics [10, 11].
- *Behaviour derived from structure* (or, shorthand, "structure behaviour"): includes attributes of the design that are measured, calculated or derived from observation of a specific design solution.
- *Structure*: includes the components of a design and their relationships. They can appear either as a set of general concept solutions or as detailed solutions. This is consistent with similar distinctions of solution structure in the design literature [2, 11].
- *Description*: includes any form of external representation produced by a designer, at any stage of the design process. Descriptions may come as sketches, (CAD) models, physical prototypes, calculations, textual expressions or other observable outcomes of designerly activity.

The coders arbitrated their coding using the Delphi method [12], discussing any differences until reaching agreement on the assigned codes.

The average agreement between coders across the thirteen protocols is 89.8 %.

**Table 2** The thirteen design protocols

| Design protocol | Variables | | | | | | |
|---|---|---|---|---|---|---|---|
| | Source location of data | Design task | Participants' expertise | Participants knowledge domain | Team size | Team composition | Length of design protocol (number of segments) |
| PI | Virginia, USA | Designing an assistive window raising device; design method: unstructured | Undergraduate students | Mechanical engineering | 2 | Homogeneous | 891 |
| P2 | Virginia, USA | Designing an assistive window raising device; design method: brainstorming | Undergraduate students | Mechanical engineering | 2 | Homogeneous | 614 |
| P3 | Virginia, USA | Designing an assistive door opening device; design method: morphological analysis | Undergraduate students | Mechanical engineering | 2 | Homogeneous | 500 |
| P4 | United Kingdom | Designing a novel thermal ink pen | Professional designers | Electronics, mechatronics ergonomics, business | 7 | Heterogeneous | 1,280 |
| P5 | California, USA | Designing a Software system to simulate road traffic Controls | Professional designers | Software | 2 | Homogeneous | 596 |
| P6 | Sydney, Australia | Designing an art gallery | Professional designers | Architecture | 2 | Homogeneous | 192 |
| P7 | Utah, USA | Designing an assistive window raising device | High school students | – | 2 | Homogeneous | 426 |
| P8 | Illinois, USA | Designing a teaching device using prototyping | Undergraduate students | Mechanical engineering, psychology | 3 | Heterogeneous | 328 |
| P9 | Illinois, USA | Designing a teaching device without using prototyping | Undergraduate students | Mechanical engineering, psychology | 3 | Heterogeneous | 424 |

(continued)

**Table 2** (continued)

| Design protocol | Variables | | | | | | |
|---|---|---|---|---|---|---|---|
| | Source location of data | Design task | Participants' expertise | Participants knowledge domain | Team size | Team composition | Length of design protocol (number of segments) |
| P10 | Singapore | Designing a future personal entertainment system | Undergraduate students | Industrial design | 2 | Homogeneous | 418 |
| P11 | Singapore | Designing a coffee maker | Undergraduate students | Industrial design | 2 | Homogeneous | 782 |
| P12 | Taipei, Taiwan | Pedometer to encourage running | Undergraduate students | Industrial design | 2 | Homogeneous | 304 |
| P13 | Minnesota, USA | Designing a commercial-level website | Professional designers | Interface design, web design, business analyst | 9 | Heterogeneous | 289 |

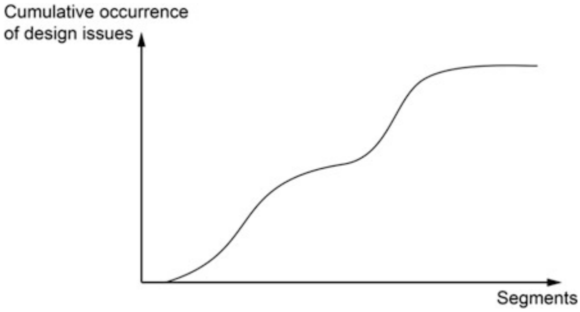


**Fig. 1** Graphical representation of the cumulative occurrence of design issues in design protocols

## Analysis Method: Cumulative Occurrence of Design Issues

The coded design protocols coded represent instances of designing as sequences of design issues. The commonalities we aim to identify are based on these sequences of design issues rather than the specific design methods used. Our approach for analysing and comparing the thirteen design protocols is to calculate the cumulative occurrence of each of the six design issues for every segment in a protocol. Specifically, the cumulative occurrence (c) of design issue (x) at segment (n) will be $c = \sum_{i=1}^{n} x_i$ where ($x_i$) equals 1 if segment (i) is coded as (x) and 0 if segment (i) is not coded as (x). Plotting the results of this equation on a graph with the segments (n) on the horizontal axis and the cumulative occurrence (c) on the vertical axis will visualise the occurrence of the design issues. Figure 1 shows a general representation of such a graph.

Based on the notion of cumulative occurrence of design issues, we determine the following qualitative measures for each of the six classes of design issues:

- *First occurrence at start*: Which design issues first occur near the start of designing, and which first occur later?
- *Continuity*: Which design issues occur throughout designing, and which occur only up to a certain point?
- *Shape of the graph*: For which design issues is the cumulative occurrence graph linear, and for which is it non-linear?

In addition, we will determine the following quantitative measures:

- *Slope*: This is a measure for the speed at which design issues are generated.
- *$R^2$ (coefficient of determination)*: This is a measure for the linearity of the graph. We will set a minimum value of 0.950 as a condition for linearity.

All of these measures are independent of the length of the design session. This allows comparing design protocols with different numbers of segments.

## Results

In this section we present the qualitative and quantitative measures we derived from analysing the design protocols. We describe the raw data as sets of graphs representing the cumulative occurrence of design issues of all thirteen protocols. These graphs are not presented for the purpose of measurement but for developing a qualitative understanding of the range and scale of the data. The graphs are of differing lengths, since each protocol has a different length.

### *Requirement Issues*

The cumulative occurrences of requirement issues are shown graphically in Fig. 2. Quantitative and qualitative measures for the requirement issue are provided in Table 3 for all but six design protocols, which are indicated by the asterisks. In these protocols the number of data points was too low (less than 10) to allow meaningful statements and statistical analyses of this issue. The remaining seven design protocols all exhibit the same qualities:

- Requirement issues in all protocols analysed occur from the start of the design session.
- Requirement issues in all protocols analysed occur discontinuously, as shown by the graphs tending to flatten out with increasing numbers of segments.
- The cumulative occurrence of requirement issues in all protocols analysed is non-linear. The mean $R^2$ value of 0.791 (standard deviation of 0.122) is below the threshold value of 0.950.

### *Function Issues*

The cumulative occurrences of function issues are shown in Fig. 3. The corresponding quantitative and qualitative measures are provided in Table 4, with the exception of five protocols that have too small datasets (as indicated by the asterisks in the table). From the protocols we analysed, we can make the following observations:

- Function issues in all protocols analysed occur from the start of the design session.
- Function issues in most protocols analysed occur discontinuously, as their graphs flatten out towards the end of the design session. There are continuous occurrences in protocols P1 and P6; however, the total number of data points in

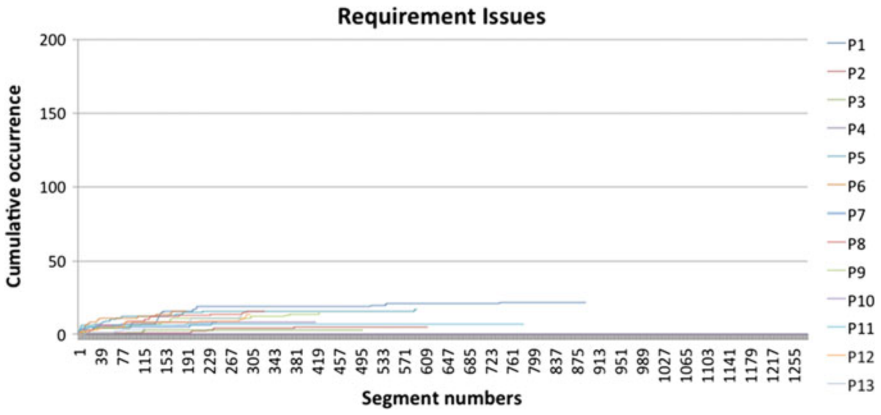


**Fig. 2** Cumulative occurrence of requirement issues

**Table 3** Quantitative and qualitative measures related to the cumulative occurrence of requirement issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.018 | 0.646 | Yes | No | Non-linear |
| P2* | – | – | – | – | – |
| P3* | – | – | – | – | – |
| P4* | – | – | – | – | – |
| P5 | 0.014 | 0.621 | Yes | No | Non-linear |
| P6 | 0.055 | 0.791 | Yes | No | Non-linear |
| P7* | – | – | – | – | – |
| P8 | 0.043 | 0.882 | Yes | No | Non-linear |
| P9 | 0.028 | 0.900 | Yes | No | Non-linear |
| P10* | – | – | – | – | – |
| P11* | – | – | – | – | – |
| P12 | 0.025 | 0.772 | Yes | No | Non-linear |
| P13 | 0.047 | 0.928 | Yes | No | Non-linear |
| Mean | 0.033 | 0.791 | | | |
| Stdev | 0.016 | 0.122 | | | |

*No statistical results produced due to small dataset (<10 data points)

these protocols (22 and 16, respectively) is fairly low, which makes their qualitative assessment less reliable.

- The cumulative occurrence of function issues in most protocols analysed is non-linear. The mean $R^2$ value is 0.888 (standard deviation of 0.071), which is below the threshold of 0.950. We found linearity only in protocol P3 ($R^2$ value of 0.960), yet based on a fairly small dataset (24 data points).

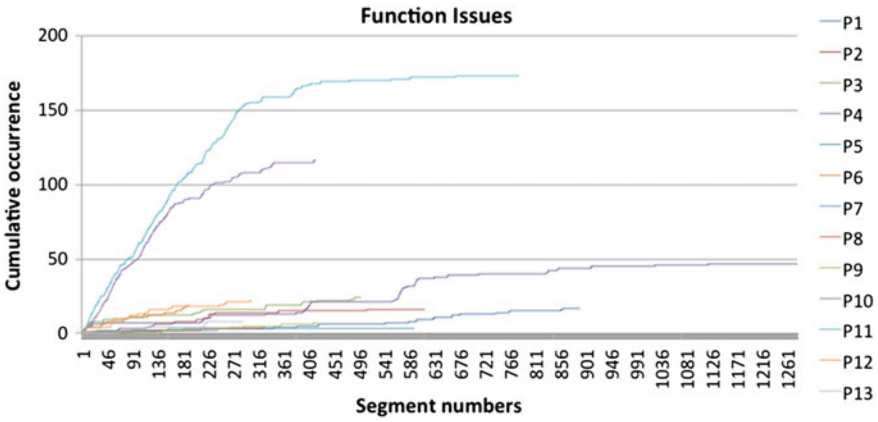


**Fig. 3** Cumulative occurrence of function issues

**Table 4** Quantitative and qualitative measures related to the cumulative occurrence of function issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.019 | 0.929 | Yes | Yes | Non-linear |
| P2 | 0.028 | 0.830 | Yes | No | Non-linear |
| P3 | 0.034 | 0.960 | Yes | No | Linear |
| P4 | 0.041 | 0.923 | Yes | No | Non-linear |
| P5* | – | – | – | – | – |
| P6 | 0.074 | 0.948 | Yes | Yes | Non-linear |
| P7* | – | – | – | – | – |
| P8* | – | – | – | – | – |
| P9* | – | – | – | – | – |
| P10 | 0.271 | 0.884 | Yes | No | Non-linear |
| P11 | 0.190 | 0.745 | Yes | No | Non-linear |
| P12 | 0.064 | 0.883 | Yes | No | Non-linear |
| P13* | – | – | – | – | – |
| Mean | 0.090 | 0.888 | | | |
| Stdev | 0.091 | 0.071 | | | |

*No statistical results produced due to small dataset (<10 data points)

## *Expected Behaviour Issues*

The cumulative occurrences of expected behaviour issues are shown graphically in Fig. 4. Quantitative and qualitative measures are summarised in Table 5.

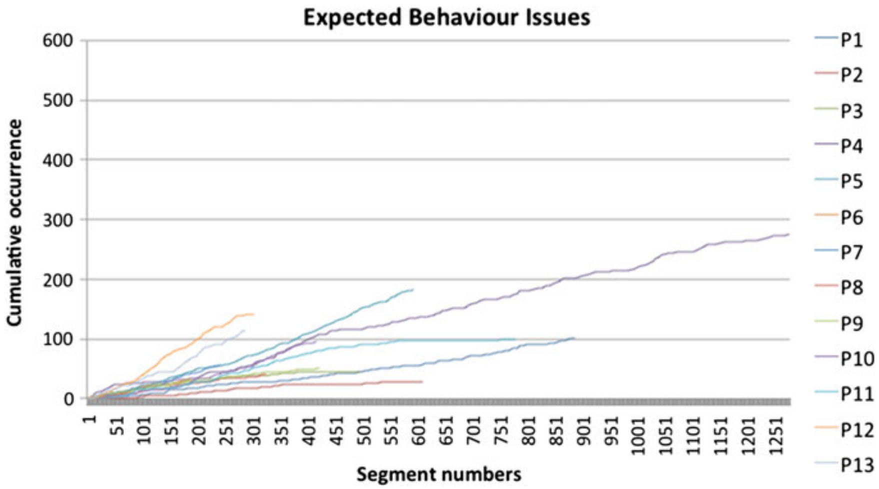


**Fig. 4** Cumulative occurrence of expected behaviour issues

**Table 5** Quantitative and qualitative measures related to the cumulative occurrence of expected behaviour issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.110 | 0.984 | Yes | Yes | Linear |
| P2 | 0.056 | 0.954 | No | No | Linear |
| P3 | 0.090 | 0.929 | Yes | No | Non-linear |
| P4 | 0.222 | 0.995 | Yes | Yes | Linear |
| P5 | 0.314 | 0.986 | Yes | Yes | Linear |
| P6 | 0.175 | 0.975 | Yes | Yes | Linear |
| P7 | 0.240 | 0.979 | Yes | Yes | Linear |
| P8 | 0.130 | 0.989 | Yes | Yes | Linear |
| P9 | 0.118 | 0.981 | Yes | Yes | Linear |
| P10 | 0.239 | 0.959 | Yes | Yes | Linear |
| P11 | 0.150 | 0.930 | Yes | No | Non-linear |
| P12 | 0.530 | 0.993 | Yes | Yes | Linear |
| P13 | 0.397 | 0.984 | Yes | Yes | Linear |
| Mean | 0.213 | 0.972 | | | |
| Stdev | 0.135 | 0.022 | | | |

These thirteen protocols exhibit similarities with some exceptions:

- Expected behaviour issues in all but one of the protocols occur from the start of the design session. The one exception is protocol P2; expected behaviour issues here occur with some delay.
- Expected behaviour issues in most protocols occur continuously. Exceptions are protocols P2, P3 and P11, where the occurrence of expected behaviour issues drops off towards the end of the design sessions.

- The cumulative occurrence of expected behaviour issues in most protocols is linear. The mean $R^2$ value is 0.972 (standard deviation of 0.022), which is above our threshold of 0.950. Only two protocols exhibit non-linearity in the occurrence of expected behaviour issues; these are protocols P3 and P11. This is probably related to the discontinuity observed in these protocols.

The mean slope of the graphs is 0.213, with a standard deviation of 0.135.

## *Structure Behaviour Issues*

The cumulative occurrences of structure behaviour issues are shown in Fig. 5, with Table 6 providing quantitative and qualitative measures, except for one protocol, P13 (as indicated by the asterisk in Table 6), that had too few occurrences of structure behaviour issues to be taken into account.

There are strong similarities across the protocols analysed:

- Structure behaviour issues in most protocols occur from the start of the design session. There are exceptions in protocols P3, P10 and P11, where structure behaviour issues occur with some delay.
- Structure behaviour issues in all protocols analysed occur continuously.
- The cumulative occurrence of structure behaviour issues in most protocols is linear. The mean $R^2$ value is 0.982 (standard deviation of 0.019), which is above the threshold of 0.950. Only one protocol, P12, exhibits non-linearity.

The mean slope of the graphs is 0.246, with a standard deviation of 0.092.

## *Structure Issues*

The cumulative occurrences of structure issues are shown in Fig. 6, with Table 7 providing quantitative and qualitative measures.

Commonalities across the thirteen protocols include:

- Structure issues in most protocols occur from the start of the design session. There are exceptions in protocols P10 and P11, where the designers did not generate structure issues until later in the design session.
- Structure issues in all protocols occur continuously.
- The cumulative occurrence of structure issues in all protocols is linear. The mean $R^2$ value is 0.994 (standard deviation of 0.004), which is above the threshold of 0.950. In this analysis we ignored the initial segments of P10 and P11 based on the late beginning of a clearly linear part of the graphs representing these protocols.

The mean slope of the graphs is 0.386, with a standard deviation of 0.088.

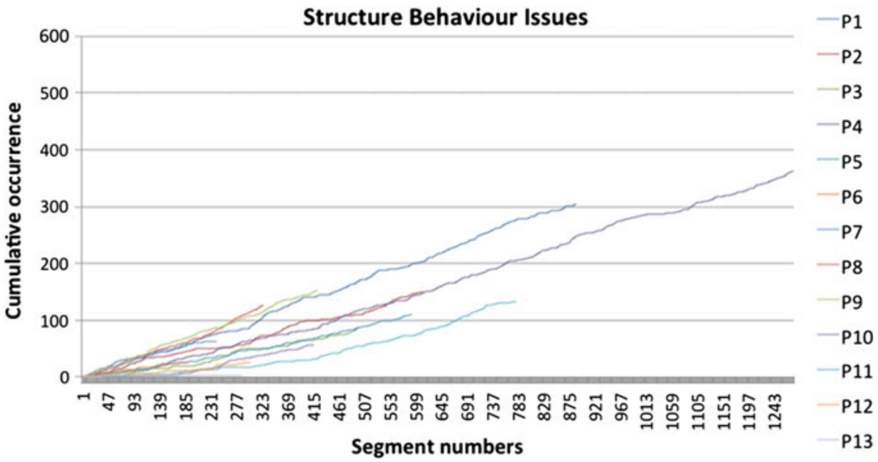


**Fig. 5** Cumulative occurrence of structure behaviour issues

**Table 6** Quantitative and qualitative measures related to the cumulative occurrence of structure behaviour issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.352 | 0.997 | Yes | Yes | Linear |
| P2 | 0.235 | 0.987 | Yes | Yes | Linear |
| P3 | 0.179 | 0.982 | No | Yes | Linear |
| P4 | 0.296 | 0.995 | Yes | Yes | Linear |
| P5 | 0.186 | 0.991 | Yes | Yes | Linear |
| P6 | 0.138 | 0.973 | Yes | Yes | Linear |
| P7 | 0.283 | 0.975 | Yes | Yes | Linear |
| P8 | 0.372 | 0.989 | Yes | Yes | Linear |
| P9 | 0.361 | 0.998 | Yes | Yes | Linear |
| P10 | 0.219 | 0.992 | No | Yes | Linear |
| P11 | 0.254 | 0.974 | No | Yes | Linear |
| P12 | 0.079 | 0.928 | Yes | Yes | Non-linear |
| P13* | – | – | – | – | – |
| Mean | 0.246 | 0.982 | | | |
| Stdev | 0.092 | 0.019 | | | |

*No statistical results produced due to small dataset (<10 data points)

## *Description Issues*

The cumulative occurrences of description issues are shown in Fig. 7. Quantitative and qualitative measures are summarised in Table 8. Protocols P8, P9, P12 and P13 were not taken into account in this analysis because of the small dataset they provide for description issues.

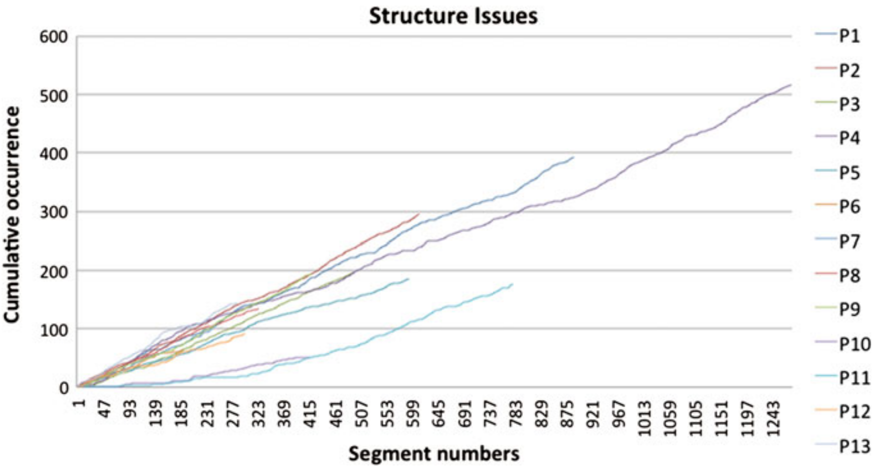


**Fig. 6** Cumulative occurrence of structure issues

**Table 7** Quantitative and qualitative measures related to the cumulative occurrence of structure issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.437 | 0.999 | Yes | Yes | Linear |
| P2 | 0.476 | 0.999 | Yes | Yes | Linear |
| P3 | 0.417 | 0.998 | Yes | Yes | Linear |
| P4 | 0.378 | 0.993 | Yes | Yes | Linear |
| P5 | 0.313 | 0.994 | Yes | Yes | Linear |
| P6 | 0.372 | 0.988 | Yes | Yes | Linear |
| P7 | 0.411 | 0.997 | Yes | Yes | Linear |
| P8 | 0.424 | 0.998 | Yes | Yes | Linear |
| P9 | 0.469 | 0.995 | Yes | Yes | Linear |
| P10* | 0.186 | 0.993 | No | Yes | Linear |
| P11** | 0.336 | 0.993 | No | Yes | Linear |
| P12 | 0.287 | 0.990 | Yes | Yes | Linear |
| P13 | 0.507 | 0.989 | Yes | Yes | Linear |
| Mean | 0.386 | 0.994 | | | |
| Stdev | 0.088 | 0.004 | | | |

*The first 160 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

**The first 300 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

We can observe the following commonalities:

- Description issues in most protocols do not occur from the start. Exceptions include protocols P2, P3 and P10.
- Description issues in most protocols occur continuously, except in P1 and P4.

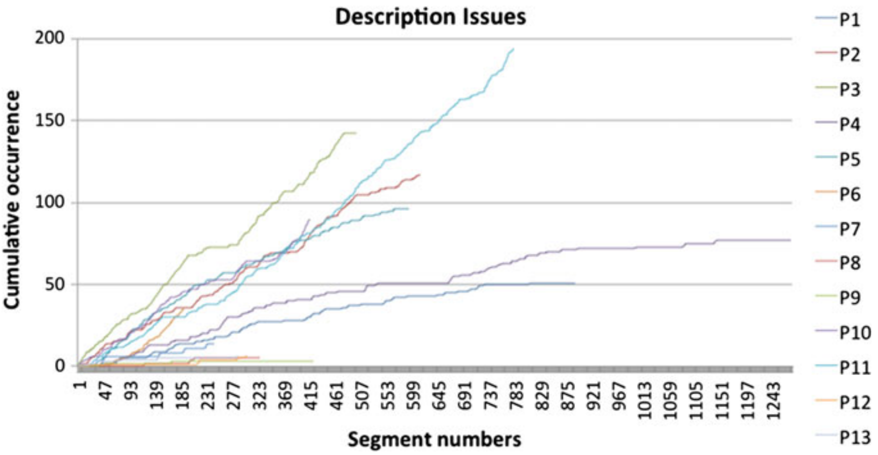


**Fig. 7** Cumulative occurrence of description issues

**Table 8** Quantitative and qualitative measures related to the cumulative occurrence of description issues

| Protocol | Slope | $R^2$ | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| P1 | 0.064 | 0.970 | No | No | Linear |
| P2 | 0.196 | 0.994 | Yes | Yes | Linear |
| P3 | 0.274 | 0.992 | Yes | Yes | Linear |
| P4 | 0.063 | 0.934 | No | No | Non-linear |
| P5 | 0.170 | 0.973 | No | Yes | Linear |
| P6* | 0.238 | 0.962 | No | Yes | Linear |
| P7 | 0.051 | 0.881 | No | Yes | Non-linear |
| P8** | – | – | – | – | – |
| P9** | – | – | – | – | – |
| P10 | 0.192 | 0.979 | Yes | Yes | Linear |
| P11 | 0.249 | 0.986 | No | Yes | Linear |
| P12** | – | – | – | – | – |
| P13** | – | – | – | – | – |
| Mean | 0.166 | 0.964 | | | |
| Stdev | 0.086 | 0.036 | | | |

*The first 46 segments of the protocol are ignored in slope and linearity calculation to take into account that the first occurrence is not at the start

**No statistical results produced due to small dataset (<10 data points)

- The cumulative occurrence of description issues in most protocols is linear, except in P4 and P7. The mean $R^2$ value is 0.964 (standard deviation of 0.036), which is above the threshold of 0.950. In this analysis we ignored the initial segments of P6 based on the late beginning of a clearly linear part of the graph representing this protocol.

**Table 9** Summary of commonalities

| Design issue | Mean slope (Stdev) | Mean $R^2$ (Stdev) | First occurrence at start | Continuity | Shape |
|---|---|---|---|---|---|
| Requirement | 0.033 (0.016) | 0.791 (0.122) | Yes | No | Non-linear |
| Function | 0.090 (0.091) | 0.888 (0.071) | Yes | No* | Non-linear* |
| Expected behaviour | 0.213 (0.135) | 0.972 (0.022) | Yes* | Yes* | Linear* |
| Structure behaviour | 0.246 (0.092) | 0.982 (0.019) | Yes* | Yes | Linear* |
| Structure | 0.386 (0.088) | 0.994 (0.004) | Yes* | Yes | Linear |
| Description | 0.166 (0.086) | 0.964 (0.036) | No** | Yes* | Linear* |

*For at least 75 % of the protocols analysed
**For 66 % of the protocols analysed

The mean slope of the graphs is 0.166, with a standard deviation of 0.086.

## Summary of Commonalities Found

Our analysis has uncovered a number of commonalities among the protocols. Table 9 summarises our findings.

Some of the commonalities are not surprising, given existing assumptions, observations and hypotheses about designing. For example, it is often assumed that the design process commences with clarifying a set of requirements and functions [10, 11]. This is confirmed by our empirical data that indicates that requirement issues and function issues occur from the start of a design session. Our graphs also show that these two issues occur discontinuously, which is consistent with many design theories that see a diminishing role of requirements and functions in the later stages of designing. Further, our finding that structure issues occur from the start of the design process confirms observations by other design researchers [2, 13], namely, that designers tend to commit to specific solutions early on.

There are other commonalities that have not been observed in previous studies. One observation is that expected behaviour issues, structure behaviour issues, structure issues and description issues occur continuously throughout design sessions. They also occur at a highly linear rate, with most $R^2$ values exceeding the threshold of 0.950. A comparison of the slopes in Table 9 indicates that the rate at which structure issues are generated is significantly higher than for any other design issue. There is very little variance in the slopes for structure issues and structure behaviour issues across different design protocols.

## Conclusion

Our empirical results indicate that there are regularities across designing that are independent of individual parameters including location, knowledge domain, expertise, team size, team composition, design task and length of the design session. Many of these regularities can be seen as significant, based on the heterogeneity of the data and on the statistical evidence. It supports the premise that designing can be studied as a distinct human activity that transcends disciplinary boundaries and specific design situations [14, 15].

The findings presented in this paper provide a starting point for two future research avenues. One avenue includes increasing the empirical basis of our findings by analysing a larger number of design protocols with additional parameters. Examples include studying design processes in collocated versus remotely located teams, in single versus multiple design sessions, using synchronous versus asynchronous modes of communication, and with designers of different gender. The results may explain some of the exceptions or “outliers” we found in our analysis of design protocols, such as the designers’ delayed focus on structure issues in protocols P10 and P11.

The other avenue includes investigating some of the unexpected results of our analysis. This includes the strong focus of designers on structure issues, in terms of the high rate at which they are generated, and the high continuity and linearity with which they accumulate. What research is needed to explain this phenomenon? Are there any implications for design theory or design education?

**Acknowledgements** This research is supported in part by grants from the US National Science Foundation grant nos. SBE-0750853, EEC-0934824, CMMI–0926908 and IIS-1002079. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We would like to thank the following for making their data available: Elizabeth Houle, University of Minnesota; Hao Jiang, National University of Singapore; Jeff Kan, Taylors University, Malaysia; Matthew Lammi, Utah State University; Marie Paretti, Virginia Tech; Hsien-Hui Tang, Taiwan University of Science and Technology; Christopher Williams, Virginia Tech; Robert Youmans, University of California at Northridge; Design Theory Research Symposium 7, London; and Studying Professional Software Designers Workshop, Irvine.

## References

1. Asimow M (1962) Introduction to design. Prentice-Hall, Englewood Cliffs
2. Lawson B (1980) How designers think: the design process demystified. Architectural Press, Amsterdam
3. Dym C (1994) Engineering design: a synthesis of views. Cambridge University Press, Cambridge
4. Ericsson KA, Simon HA (1993) Protocol analysis: verbal reports as data. MIT Press, Cambridge
5. Cross N, Christiaans H, Dorst K (1996) Analysing design activity. Wiley, Chichester
6. Gero JS (2010) Generalizing design cognition research. DTRS8: interpreting design thinking. DAB documents, Sydney, pp 187–198
7. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11:26–36
8. Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Des Stud 25:373–391

9. Chittaro L, Kumar AN (1998) Reasoning about function and its applications to engineering. Artif Intell Eng 12:331–336
10. Hubka V, Eder WE (1996) Design science: introduction to the needs, scope and organization of engineering design knowledge. Springer, London
11. Pahl G, Beitz W (2007) Engineering design: a systematic approach. Springer, Berlin
12. Gero JS, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19:21–61
13. Ullman DG, Dietterich TG, Stauffer LA (1988) A model of the mechanical design process based on empirical data. Artif Intell Eng Des Anal Manuf 2:33–52
14. Cross N (1982) Designerly ways of knowing. Des Stud 3:221–227
15. Visser W (2009) Design: one, but in different forms. Des Stud 30:187–223

# Part V
# Design Generation

# Integrated Generative Design Tools for the Mass Customization of Furniture

**Mário Barros, José Pinto Duarte and B. M. Chaparro**

**Abstract** This paper presents the optimisation step in a grammar-based generative design system for the mass customisation of furniture. The ongoing research assesses the use of integrated CAD-CAE tools in the development of a digital design process involving closer collaboration between design and engineering as a feasible motivation for optimum designs. The optimisation of structural behaviour is illustrated by a series of experiments using a simulated annealing algorithm to explore solutions for custom chairs generated by parametric models. Constraints are defined according to the aesthetic considerations established in the design language. The paper concludes with a discussion of the effective use of integrated performance tools in furniture design methodology in the age of mass customisation.

## Introduction

The use of digital design tools has progressed from the execution of final drawings towards a framework in which the design, analysis and manufacturing phases are integrated. In a design-to-production context—in both academic and practical terms—the convergence of CAD, FEM analysis and CAM provides new opportunities for collaboration in the fields of design and engineering. The assimilation of these tools may reverse the traditional process of product development so that it becomes "*material*, *structure*, *form*" [1].

---

M. Barros · J. P. Duarte (✉)
Technical University of Lisbon, Lisbon, Portugal
e-mail: jduarte@fa.utl.pt

M. Barros
e-mail: milbarros@gmail.com

B. M. Chaparro
Politechnic Institute of Tomar, Tomar, Portugal

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_16,

In the field of CAD, parametric design offers the possibility of exploring new versions of a design based on changes in the relationships between elements of a schema [2]. Defining and editing the relationships becomes an integral part of the design process.

Finite Element Method (FEM) is the numerical technique most widely used to analyse solid structures [3]. In a CAE environment, FEM provides the ability to test designs based on the calculation of a digital discrete model.

Combining the freedom to explore new versions using parametric design with the pragmatism of performance considerations enables research to be undertaken into optimum designs.

The potential offered by experimentation aimed at efficiency in a highly controlled digital environment is the main theme of this article. The use of design automation to guide the search for the optimum design will be assessed, in terms of aesthetics and structural requirements.

## Research Scope

This work is part of ongoing PhD research aimed at developing a digital design process for mass customisation in the furniture industry. Considering a dynamic design process composed by the designer, the manufacturer and the user, this research is focused on closing the gap between the first two, establishing a methodology that allows a future active participation of the end user in the furniture design process. The research examines technical issues of the design activity and is aimed at developing design methods that allows the designer to generate alternative design solutions in direct correlation with fabrication issues, through the use of computational design thinking and computational design tools.

The research consists of (1) the study of an existing design language in a way that accounts for the generation of new designs that maintain the original style features; and (2) the transformation of the grammar to create new design styles based on an assessment of the knowledge acquired in the first research phase. The first stage is intended to develop a computer implemented design system that allows the design and fabrication of mass customised chairs. In the second stage the design system will be used to demonstrate a range of creative possibilities that can occur under the paradigm of mass customisation. This paper describes research belonging to the first stage.

The conceptual process of the research is based on the customisation schema devised by Duarte [4], which considers the existence of a generative design system and a production system controlled by a computational system that supports both the exploration of solutions and generation of the data for CNC production. In the design system for the mass customisation of furniture, the generative design system comprises two phases—shape generation and shape evaluation, as outlined in Fig. 1.

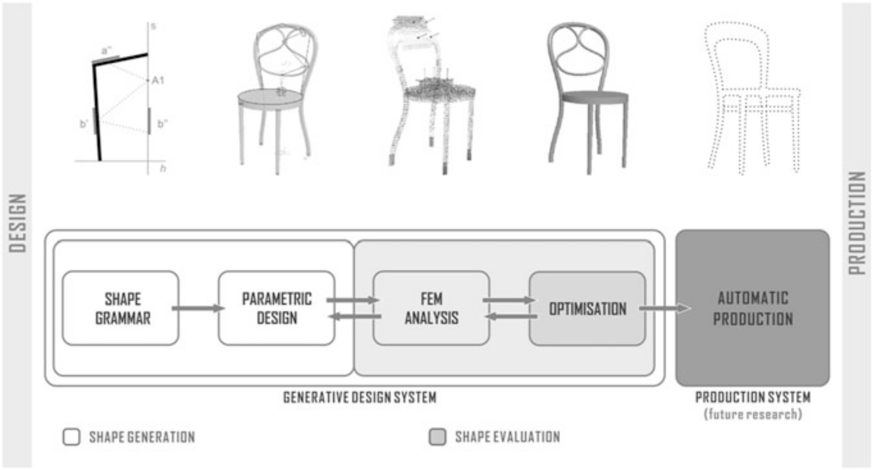


**Fig. 1** Design system for the mass customisation of furniture

The shape generation phase encompasses the use of a shape grammar [5] and parametric design. A shape grammar was developed to encode an existing design style, which can be used by the designer to create alternative design solutions by manipulating rule application. The shape grammar was then encoded into parametric design models in CAD software to assist the designer in further exploration of the solutions to meet individual user's needs, by assigning values to the parameters.

The digital data produced in the first phase was linked to the shape evaluation phase. The second phase of the generative design system acts upon previous information and is an iterative step in the continuous flow of information from conception to fabrication. It analyses structural behaviour of each custom design through the use of FEM and optimises the digital model considering a particular set of physical and material conditions. The present paper is focused on describing the optimisation step of the shape evaluation phase.

Future research will address the development and testing of the production system. The production system will transform the digital data from the design system into a set of instructions for CNC production of the furniture.

## The Thonet Custom Chair Case Study

In order to illustrate the potential application of the system envisaged for mass customisation in the furniture industry, an existing design style was selected. This approach aimed to provide means of comparing and validating the results, both in terms of aesthetics and production. The chair was the selected typology because its design requires direct inputs from the human body and allows the experimentation and demonstration of several aspects within a particular vision of design and manufacturing methods [6].

As Thonet chairs may be considered to represent the mass production paradigm, they were chosen to assess the degree of transformation that might occur when changing the paradigm to mass customisation. Strong material constraints and non-standard production techniques constituted layers of complexity and established firm boundaries for the research project.

## The Generative Design System

Shape grammars were used as the formalism to provide an algorithmic description of the Thonet design style and to generate new designs by a creative application of the rules. Thonet chair design grammar (for additional information see [7]) comprised a simplified representation for the curves, similar to the one developed by Knight for the Hepplewhite chairs [8]. The grammar rules are focused on the chair-back design—both frame and inner area—because its appearance is modified between versions while other structural elements remain similar. Thonet design grammar is the Cartesian product of different algebras; labels (points and segments) were used in order to regulate the position of the elements and weights to describe material proprieties. The grammar provides the ability to create several designs that can be classified under three main design families, each characterized by the style of the connection between the backrest inner element, the backrest frame and the seat. Figure 2 shows the derivation of a design following the shape grammar rules.

Each design provided by the grammar was encoded as a parametric design model using CATIA. The aim of using an existing CAD package was to assist the furniture designer in the generation of multiple design alternatives in a digital environment [9, 10]. The use of a twofold step in the generative design system reflected the preset condition of family identification and reproduction—shape grammar—and the reconfigurable smart models—parametric models. In short, the shape grammar acts as a genotype by providing a conceptual structure to guide the parametric models, which act as phenotypes, allowing for variations in different environment.

As the main purpose of the system was the production of customised chairs, the generative design system had to include an iterative shape evaluation step. This step estimates the degree of feasibility prior to real construction. Shape evaluation is a twofold step comprising FEM analysis and performance optimisation.

FEM analysis was used to test the structural behaviour of the parametric model products [10]. Tests were conducted using the CATIA Generative Structural Analysis (GSA) workbench. By using the same software the design process was streamlined, due to the time saved in terms of transferring digital data. It also made it possible for the reconfigurable logic of the parametric model to be retained and

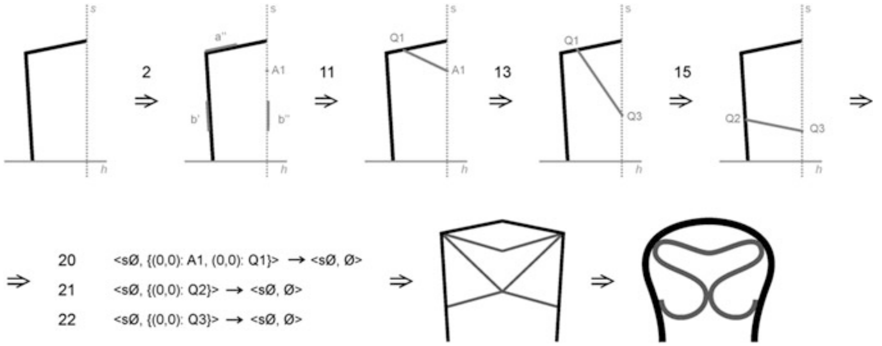


**Fig. 2** Derivation of a new chair by shape grammar rule application

updated in the GSA workbench, thus establishing interdependent relationships between the parametric logic and the structural analysis logic. Although there was interoperability between the workbenches, each version had to be analysed and manually refined in the parametric model if it failed to meet the structural requirements. The automation goal in this design process routine led to the formulation of the optimisation step, therefore creating an integrated generative design process.

The optimisation step formulated in this article aims to provide and automated support for product synthesis, by searching the best solution under predetermined conditions. The boundary conditions for the optimisation problem are both aesthetic and functional; reflecting the need for a design complied in the Thonet style that performs within ISO standards.

## Optimisation

Design problems are usually poorly defined [11]. The evaluation and refinement phases are the most time-consuming parts of the traditional design process due to the complexity of the existing features, some of which are interrelated. In an iterative-intuitive design process there is no optimal solution, but a range of acceptable solutions [12].

In a digital methodological framework, the computer is used to assist in the creation of algorithmic driven designs and can also be used to automate the search for alternatives to a preferred solution.

Optimisation is the ability to find the best solution within a given space. To optimise a system it is necessary to formulate a problem in mathematical terms with different constraints that can be measured and satisfied with performance criteria expressed as objective functions [13]. The search for optimum elements will be achieved through a simulation performing optimisation algorithms.

## *Optimisation Algorithms*

Optimisation algorithms can be divided into two groups: deterministic and probabilistic. Deterministic algorithms were the first to be used and their applications correspond to the beginnings of computation. They are often used as exploratory algorithms or for local optimisation [14] when there is a clear insight into the nature of the variables. Heuristics divide the problem into sub-problems and each iteration determines the criterion for finding the next state.

Probabilistic algorithms are used in problems when there are uncertainties in the elements, the search space or the path for solutions. In this class of algorithm the probabilistic component is included in the routine, so that the current state does not determine any other state. These algorithms do not guarantee the best global solution but the trade-off between time and the effectiveness of the solution makes them suitable for structural optimisation problems [13].

## *Simulated Annealing*

Simulated annealing is a stochastic optimisation method analogous to the annealing of metals in statistical mechanics [15]. Annealing is a process in which melted metal is slowly cooled to solidify in its minimum energy state. Temperature is the gradient variable that stabilises the system in each state. The search for the next stage is based on a degree of randomness that also involves comparison with the energy of the system in the previous state. If the next stage is suitable, it will be accepted, otherwise the previous step is retained and a new iteration begins in order to find another state. The iterations stop when the energy of the system reaches a local minimum configuration.

The ability to search for global optima makes this optimisation technique suitable for combinatorial and non-linear problems with several constraints.

## *Optimisation and Shape Grammars*

The motivation for designing for optimality within a design process specified by a shape grammar has been subject of previous research. Cagan and Mitchell [16] proposed the shape annealing concept, an optimisation technique that combined the simulated annealing algorithm with shape grammar formalism. The algorithm controls rule application, providing early-stage design exploration based on structural optimisation. This concept has been applied in the configuration of trusses [17, 18] and has evolved into structural topology and shape annealing (STSA) [18] with practical applications for the design of civil structures [19].

In the case presented in this paper, simulated annealing is applied to sizing and shape optimisation. Since the shape grammar solutions are encoded into equivalent parametric design models, there is no generation by rule application and shape emergence. The algorithm must handle parameters' constraints in order to optimise their size and shape to meet predefined intents. This is accomplished by using the available computational optimisation techniques embedded in CATIA. The existence of a graphic user interface provided a shorter learning curve for the pre-processing and post-processing of data, whilst meeting the initial goals for design automation.

## *Optimisation of a Custom Chair*

The reason for integrating an optimisation step into the generative design system for Thonet chairs is to guarantee the minimum thickness of the beech profiles in compliance with ISO standards. The aim of using this computational technique is to permit additional exploration of a given design and the possibility of automating shape refinement, thus aiding the designer in the search for an acceptable solution within a given set of conditions.

The optimisation process for custom chairs was performed in the CATIA Product Engineering Optimizer (PEO) workbench, using a standard tool that includes a graphic user interface to allow the user to encode parameters as design variables to be optimised to a minimum, maximum, or target given value. Different built-in algorithms can be selected to optimise the given design. Iterations that analyse the model across the design variables in order to meet the target condition are saved in a *.xls file.

In this case study, stochastic optimisation using a simulated annealing algorithm was iterated in the different experiments.

## *The Parametric Model and the FEM Model*

The data collected for the optimisation of a custom Thonet chair is derived from both the parametric and the FEM design models.

In the parametric model, the chair generation involved selecting the *.CATPart model and the respective *.xls file and then assigning values to the design parameters using the spreadsheet. In the *.CATPart model, the desired configuration was selected and the 3D model updated [7].

Prior to the definition of the structural model, the loading pads and material proprieties were set in the parametric model [10]. The FEM model was completed with information on applied loads, translation constraints, and mesh description (see Fig. 3). The goal was to reproduce in a digital environment the standard conditions of ISO test methods for domestic chairs [20].

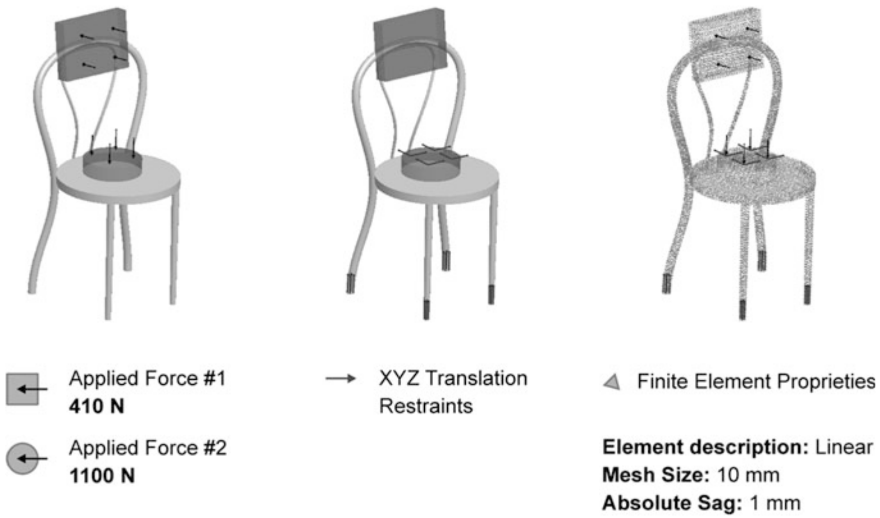


**Fig. 3** Key steps of the FEM model setup for structural analysis

The tests were examined considering the Von Mises yield criterion. In order to read the results, the beech wood design resistance was calculated using the Eurocode 5 function that takes into account the effect of load duration and a factor for material propriety in the ultimate limit state conditions [21]. The corresponding stress limit value to compression parallel to grain is 21.2 MPa.

## *Problem Formulation*

Two problem formulations were submitted for optimisation experiments: (1) minimisation of maximum equivalent stress and (2) a target value of 18 MPa, corresponding to a safety factor of 15 % below the resistance limit. The respective objective functions can be expressed as:

1. $\min\,(\max(\sigma_{eqi}))$
2. $\max(\sigma_{eqi}) \sim 18$ MPa

The variables that needed to be modified in order to achieve the goals were selected from the parametric model and their domains were constrained to a range of integer values, thus setting a discrete optimisation problem. The range of values was set as a domain in which the chairs generated would comply with the proportions of the Thonet design style (see Table 1, Fig. 4). The corresponding version for the inferior range does not meet the structural requirement. Its general proportions are retrieved from a standard Thonet chair, but thinner beech wood profiles for the backrest inner and outer frames are employed. The proportions of

**Table 1** Initial problem setup

| Parameter | | Inferior range | Superior range | Step |
|---|---|---|---|---|
| $p_1$ | Backrest outer frame diameter | 27 mm | 40 mm | 1 mm |
| $p_2$ | Backrest inner frame diameter | 10 mm | 30 mm | 1 mm |
| $p_3$ | Backrest angle | 8 deg | 13 deg | 1 deg |
| $p_4$ | Backrest width | 400 mm | 510 mm | 1 mm |
| $p_5$ | Backrest height | 420 mm | 530 mm | 1 mm |
| $p_6$ | Seat height | 450 mm | 480 mm | 1 mm |
| $p_7$ | Seat diameter | 420 mm | 510 mm | 1 mm |
| $p_8$ | Seat thickness | 30 mm | 60 mm | 1 mm |
| $p_9$ | Leg angle | 4 deg | 10 deg | 1 deg |

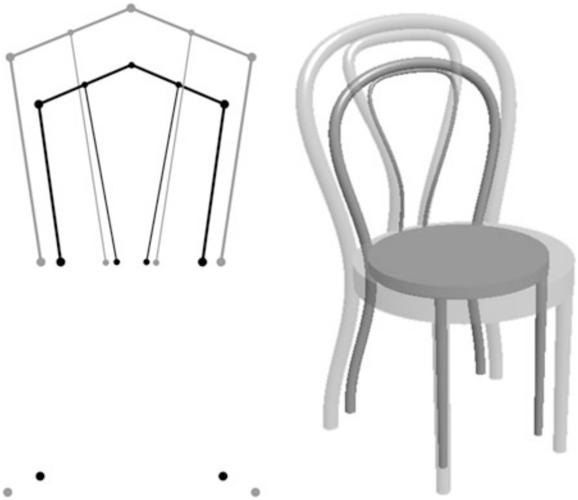


**Fig. 4** Initial chair and search space for optimization

the corresponding version for the superior range were set by analysing the dimensions of Thonet armchairs from the 1904 catalogue [22]. Backrest outer frame diameter parameters were set to use a thicker solid bendable beech wood profile and its angle variation set according to seating considerations recommended in anthropometric literature [23].

The compressive yield strength parallel to grain for the initial chair design was calculated as 39.4 MPa, 84 % above the yield limit. This approach was employed in order to maximise the algorithm's capacity to search in a larger space, rather than starting with a local minima.

A simulated annealing (SA) algorithm was selected to search for optimum designs. In the PEO workbench, the SA optimisation setup provides the user with the ability to select the speed of convergence, from *slow*, *medium*, *fast* and *infinite* (hill climbing). Conversion occurs when the algorithm reaches a solution or limits the search space to a smaller domain of the problem.

Termination criteria could be set as time, the maximum number of iterations or the number of iterations without improvement.

A custom chair was submitted to optimisation experiments with different convergence speeds and different times established as termination criteria.

The estimated size of the universe of solutions generated by each parametric model was calculated using CATIA Design of Experiments procedure. This tool allows the user to evaluate interaction between parameters and to predict the number of possible combinations, considering a step size for each parameter. Adjusting the number of levels for each parameter ($p_1 = 1$; $p_2 = 2$; $p_3 = 1$; $p_4 = 10$; $p_5 = 10$; $p_6 = 1$; $p_7 = 10$; $p_8 = 10$; $p_9 = 1$) the total number of combinations is 46,718.000.

## Analysis of Results

In the minimisation function problem, experiments were carried out using the same initial setup—dimensions and stress value—and time as termination criteria (20, 200 and 400 min) to compare the available convergence speeds. The results showed the effectiveness of the algorithm in optimising the chairs, decreasing the stress by 80.8 % (Table 2). A 400 min test with *slow* convergence speed yield the best result for this problem formulation, converging with less iterations than the *fast* and *medium* tests. In all these tests the optimisation of the backrest outer frame diameter was the key to finding the optimum design. The algorithm set this variable close to its maximum and then continued the search by combining other variables.

This type of problem formulation was useful to assess how different types of convergence speeds explore the range of the parameters. It also provided better understanding regarding the ratio between outer and inner backrest elements; initially set as 0.4 and optimised to an average of 0.75 in all the tests. These remarks can be useful to refine parameters relations in the parametric design models.

In the set of experiments carried out to minimise the function to a given target of 18 MPa, the boundary of 40 mm for the backrest outer frame diameter was not reached (see Table 3). As this problem formulation is not as linear as the minimisation problem, convergence with the objective function was achieved with additional iterations in the majority of the tests.

In the different experiments that were carried out, the *hill climbing* and *medium* convergence speeds proved best suited to meeting the initial goals of the optimisation step in the generative design system. This was accomplished with the generation of optimal solutions using the thinnest profiles and a ratio of 0.6 between outer and inner beech wood profiles. The *medium* convergence speed tests also provided a wider search space between design variables, generating chairs with a greater level of compliance with the aesthetic criteria of the Thonet design

**Table 2** Optimization results for the minimisation problem

| A | B | C | D | Value of optimised parameters (mm) | | | | | | | | | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | |
| Fast | 20 | 95 | 25 | 39 | 27 | 9 | 407 | 458 | 453 | 431 | 34 | 7 | 75.4 |
| | 200 | 1,039 | 656 | 39 | 29 | 11 | 405 | 420 | 450 | 420 | 46 | 6 | 80.2 |
| | 400 | 2,201 | 540 | 39 | 30 | 13 | 400 | 421 | 450 | 420 | 31 | 4 | 78.0 |
| Med. | 20 | 109 | 25 | 39 | 27 | 9 | 407 | 458 | 453 | 431 | 34 | 7 | 75.4 |
| | 200 | 460 | 25 | 40 | 31 | 13 | 407 | 421 | 452 | 420 | 49 | 4 | 79.2 |
| | 400 | 2,081 | 1,735 | 39 | 30 | 9 | 407 | 458 | 453 | 431 | 34 | 7 | 76.8 |
| Slow | 20 | 98 | 25 | 39 | 27 | 9 | 407 | 458 | 453 | 431 | 34 | 7 | 75.4 |
| | 200 | 1,175 | 805 | 39 | 30 | 9 | 407 | 458 | 453 | 431 | 34 | 7 | 76.0 |
| | 400 | 694 | 458 | 39 | 30 | 10 | 407 | 420 | 452 | 420 | 49 | 4 | 80.8 |
| Infinite | 20 | 28 | 23 | 39 | 29 | 11 | 405 | 420 | 450 | 420 | 34 | 8 | 79.6 |
| | 200 | 268 | 23 | 39 | 29 | 11 | 405 | 420 | 450 | 420 | 34 | 5 | 79.6 |
| | 400 | 538 | 508 | 39 | 31 | 11 | 405 | 420 | 450 | 420 | 34 | 8 | 80 |

*A* Type of convergence speed
*B* Time (min)
*C* Number of iterations
*D* Iteration of convergence
*E* Percentage of stress reduction

**Table 3** Optimization results for 18 MPa target problem

| A | B | C | D | Value of optimised parameters (mm) | | | | | | | | | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | |
| Fast | 20 | 91 | 68 | 32 | 17 | 9 | 411 | 420 | 451 | 428 | 33 | 4 | 17.9 |
| | 200 | 1,323 | 158 | 32 | 16 | 9 | 410 | 420 | 450 | 426 | 32 | 4 | 18.0 |
| | 400 | 2,663 | 2,631 | 32 | 16 | 9 | 410 | 420 | 450 | 426 | 32 | 4 | 17.9 |
| Med. | 20 | 68 | 93 | 32 | 17 | 9 | 411 | 420 | 451 | 428 | 33 | 4 | 17.9 |
| | 200 | 414 | 17 | 30 | 18 | 12 | 408 | 420 | 450 | 421 | 33 | 8 | 17.9 |
| | 400 | 855 | 638 | 30 | 18 | 12 | 408 | 420 | 450 | 421 | 33 | 8 | 17.9 |
| Slow | 20 | 102 | 68 | 32 | 17 | 9 | 411 | 420 | 451 | 428 | 33 | 4 | 17.9 |
| | 200 | 417 | 308 | 37 | 10 | 8 | 400 | 433 | 461 | 420 | 39 | 6 | 17.9 |
| | 400 | 857 | 638 | 30 | 18 | 12 | 408 | 420 | 450 | 421 | 33 | 8 | 17.9 |
| Infinite | 20 | 29 | 16 | 35 | 15 | 9 | 401 | 423 | 452 | 421 | 34 | 5 | 17.8 |
| | 200 | 1,190 | 122 | 31 | 17 | 11 | 400 | 421 | 450 | 420 | 30 | 4 | 18.0 |
| | 400 | 2,484 | 122 | 31 | 17 | 11 | 400 | 421 | 450 | 420 | 30 | 4 | 18.0 |

*A* Type of convergence speed
*B* Time (min)
*C* Number of iterations
*D* Iteration of convergence
*E* Stress value (MPa)

language. In the 20 and 400 min tests *hill climbing* was faster to reach the iteration of convergence, while in the 200 min test, *medium* convergence speed provided the optimum solution in 17 iterations of a total of 414.

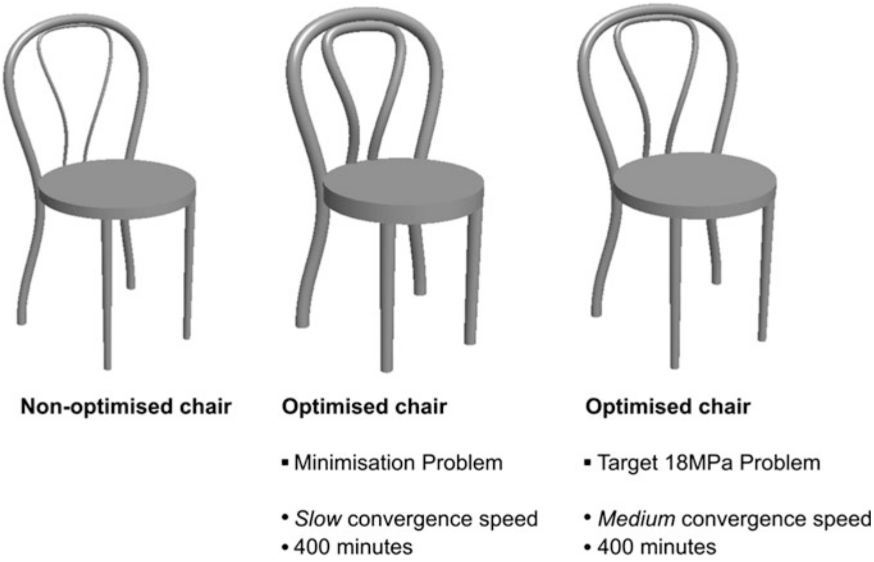


**Fig. 5** Visual comparison of non-optimised and optimised chairs

Although a stochastic algorithm does not necessarily find the same search space paths, the CATIA SA algorithm proved to have a predefined search routine. This was assessed by analysing experiment results with the same initial setup and different conversion speeds. For instance, in *fast* convergence speed tests with 20, 200 and 400 min setups and a total of 91, 1,323 and 2,663 iterations respectively, 90 iterations were equal in all three. When experiments of between 200 and 400 min were compared, a total of 1,322 iterations were equal.

Despite this fact, the SA algorithm proved useful in solving the optimisation problems formulated. The use of an automatic procedure to generate acceptable solutions within the range set by the design language streamlines the design process. It would not be possible for the designer to search in such solution space, or to generate and evaluate the same number of alternatives in the same time. Figure 5 shows the small difference between a non-optimised and optimised design of a chair.

The ability to analyse alternatives in a shorter time and retain the relevant information needed to select the best potential solutions to meet the formulated goals leads to an improvement in the performance of the design process.

As the results showed, there is a range of optimum designs for both formulated problems. In order to provide additional insights into the impact of each variable within the overall context of the structure, the results must be analysed further by the designer, rather than simply accepted. In analysing the optimisation results, the designer will also be reflecting on the nature of the parameterisation and the relationships between parameters. This knowledge can be helpful in relaxing parameters and model under-constrained sketches to support additional shape exploration.

Interpretation of the results led to additional experiments in fine-tuning the weighting of variables within the overall configuration by adjusting their step size. Tests with longer optimisation runs did not improved the results; in experimenting with different time setups, the 200 min tests provided the best trade-off between exploration of the variable space and the response to the formulated problems.

## Conclusions

The tools used in this research are effective as an integrative generative design tool for mass customisation. They can lead to better exploration of new design alternatives, stimulate multidisciplinary collaboration and find potential best solutions prior to the manufacturing phase. The prediction of better designs depends on how the problem formulation is set. In this sense, the search for optimum designs remains largely guided by the designer who specifies the system components. It must be an interactive process involving all the phases, namely the shape grammar, parametric model, FEM model and optimisation problem. The final result is a trade-off between interpretations of shape grammars as parameters, the weighting of the parameter constraints, the definition of the finite element mesh and the formulation of the optimisation problem and its constraints.

The use of optimisation automated the search process, thus streamlining the effectiveness of the generative design system. It would not be possible to produce and analyse the same amount of custom versions using a manual setup.

Shape and structural optimisation is a design formulation to be considered in both the concept and the detail phases. The use of the optimisation tool should be assessed in the configuration of the generative design system. Improvements to the amount and quality of information can lead to better decision-making and to the development of a robust design methodology using digital design tools.

Nevertheless, certain limitations in current CAD systems could be improved to make them more suitable for design activities using a mass customisation paradigm. In parametric design, new ways of generating, storing and visualising alternatives can be explored. In the optimisation phase, enhancements to the graphic user interface to allow for the possibility of customising tools to meet the specific requirements of the design problem could improve the designer's performance when using the system. In order for it to become a real computer-aided design synthesis tool with potential use in future situations, it would be beneficial for the tool to be able to record activities and learn from previous experiments [24].

There are two complementary and interactive computational design approaches in the generative design system presented here: (1) computational design thinking, encoded in the shape grammar and (2) computational design methodology, characterised by the use of digital tools. The first is used to explain a design language through vocabulary and rules. Creative, manual, semi-automatic rule application allows for design exploration in a style constrained design space. In the computational design methodology presented, shape grammar is the first input in the

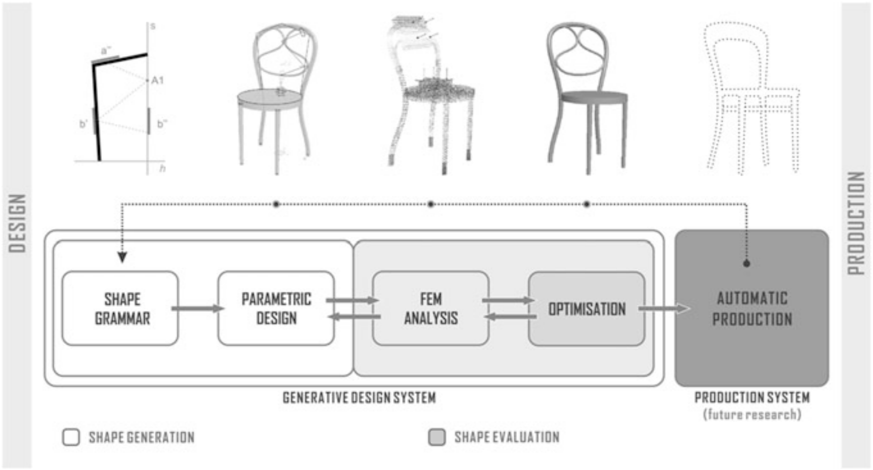


**Fig. 6** Incorporation of knowledge into the generative design system

system. Solutions are transformed into parametric models with equations describing topological relations. This step can be interpreted as limiting the creative span of the shape grammar, because it is not possible to explore solutions by rule application. The parameterisation process is a pragmatic approach that rationalises the formal variation, although it maximises variability taking account the subsequent steps of the digital design process. In a traditional design process, creativity is tapered throughout the phases until it reaches the solution that is optimised for fabrication. In this process, we aim to extend creativity to latter stages of the design process, by incorporating knowledge about physical behaviour and fabrication issues into the system itself. When the form-material relation is established there is space for including the user in the system, by having the possibility of exploring a custom solution that can perform in the physical world: under structural, manufacturing, economical, environmental and cultural conditions.

In the optimisation step, the range of variable setups is similar to the definition of the range of labels in the design of grammar rules. The analysis of the optimisation experiments can loop into the generative design system, since it extends the knowledge of chair behaviour in the simulated environment, as outlined in Fig. 6. The abstract knowledge provided by digital tools can be translated into empirical knowledge in order to refine or transform the previous steps—the shape grammar, parametric model, and FEM model.

For the particular case of this research, the loop can transform Thonet design grammar in two particular ways. Rules can be refined in order to accommodate further exploration. For instance, labels addition or subtraction can be explored to open up the grammar solution space, becoming less deterministic. The generation of new designs that can be incorporated in the corpus, initially composed by six chairs, will also lead to an augmented shape grammar. Beside the technological issues informed by the experiments conducted in this research, new designs can creatively embody socio-cultural narratives. The formalisation of these threads in Thonet design grammar will lead the system to another fluid state that will guide subsequent stages of the design system for the mass customisation of chairs. The transformation of the shape grammar in order to create new design styles will be addressed in future research.

## References

1. Oxman R, Oxman R (eds) (2010) Introduction. Architectural design: the new structuralism, vol 80(4), pp 15–23
2. Monedero J (1997) Parametric design: a review and some experiences. In: Proceedings of 15th CAADe conference, pp 17–20
3. Macneal R (1994) Finite element: their design and performance. Marcel Dekker, New York
4. Duarte (2001) Customizing mass housing: a discursive grammar for Siza's Malagueira Houses. PhD in Architecture. Massachusetts Institute of Technology
5. Gips J, Stiny G (1972) Shape grammars and the generative specification of painting and sculpture. In: Proceedings of information processing 71, pp 1460–1465
6. Nelson G (1953) Chairs. Whitney Publications, New York
7. Barros M et al (2011) Thonet chair design grammar: a step towards the mass customization of furniture. In: Proceedings of 14th CAAD futures conference, pp 181–200
8. Knight T (1980) The generation of Hepplewhite-style chair-back designs. Environ Plann B 7:227–238
9. Barros M et al (2011) A design system for the mass customization of Thonet chairs. In: Proceedings 1st sustainable intelligent manufacturing conference, pp 737–745
10. Barros M et al (2011) Digital Thonet: an automated system for the generation and analysis of custom-made chairs. In: Proceedings of 29th CAADe conference, pp 521–529
11. Cross N (2000) Engineering design methods: strategies for product design, 3rd edn. Wiley, West Sussex
12. Lawson B (2005) How designers think: the design process demystified, 4th edn. Architectural Press, Oxford
13. Arora J (2004) Introduction to optimum design, 2nd edn. Elsevier Academic Press, San Diego
14. Lewis RM, Torczon V, Trosset MW (2000) Direct search methods: then and now. J Comput Appl Math 124:191–207
15. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. Science 220:671–679
16. Cagan J, Mitchell WJ (1993) Optimally directed shape generation by shape annealing. Environ Plann B 20:5–12
17. Reddy GM, Cagan J (1995) Optimally directed truss topology generation using shape annealing. J Mech Des 117:206–209
18. Shea K et al (2005) Towards integrated performance-driven generative design tools. Autom Constr 14(2):253–264
19. Shea K, Smith I (2006) Improving full-scale transmission tower design through topology and shape optimization. J Struct Eng 132:781–790
20. ISO 7173 (1989) Ameublement—chaises et tabourets—détermination de la résistance et de la durabilité. ISO

21. Porteous J, Kermani A (2007) Structural timber design to Eurocode 5. Blackwell, Oxford
22. Wilk C (1980) Thonet: 150 years of furniture. Barron's, New York
23. Diffrient N et al (1985) Humanscale 1/2/3. MIT Press, Cambridge
24. Gero JS (1996) Design tools that learn: a possible CAD future. In: Kumar B (ed) Information processing in civil and structural design. Civil-Comp Press, Edinburgh, pp 17–22

# A Transformation Grammar-Based Methodology for Housing Rehabilitation

**Sara Eloy and Jose Pinto Duarte**

**Abstract** The goal of this research is to rehabilitate the existing housing stock to meet the new needs of dwellers in the current information society and the consequent need for the integration of Information, Communication and Automation Technologies (ICAT) in living areas. This article focuses on the use of both shape grammar and space syntax as tools to identify and encode the principles and rules behind the adaptation of existing houses to new requirements. The idea is to use such rules as part of a methodology for the rehabilitation of existing dwellings.

## Introduction

This research starts from the premise that the future of the real estate market in Portugal will require the rehabilitation of existing residential areas in order to respond to new life-styles and dwelling requirements that have emerged in an era in which information plays a structuring role in society.

The goal of this research is the definition of design guidelines and a rehabilitation methodology to support architects involved in the process of adapting existing dwellings, allowing them to balance sustainability requirements and economic feasibility with new dwelling trends such as the incorporation and updating of ICAT and the need to solve emerging conflicts affecting the use of space prompted by the introduction of new functions associated with such technologies.

---

S. Eloy · J. P. Duarte (✉)
ADETTI-IUL and TU Lisbon, Lisbon, Portugal
e-mail: jduarte@fa.utl.pt

S. Eloy
e-mail: sara.eloy@gmail.com

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_17,

In addition to defining a general methodology applicable to all the building types, the study focuses on a specific type, called “rabo-de-bacalhau,” built in Lisbon between 1945 and 1965 for which a specific methodology has been generated.

The typology known in professional jargon as “rabo-de-bacalhau” first appeared in the 1930s and became common in the 1940s and 50s. The configuration of the “rabo-de-bacalhau” buildings is characterised by a symmetrical plan consisting of a sequence of two or more rectangles, the smallest of which overlooks the open space in the rear. The buildings have a “right and left” arrangement, a predominance of reticulate concrete structures filled with masonry walls and vary in height with six floors being the average. In general the dwellings are essentially very similar and usually have a two or three bedroom layout and relatively small and very divided areas. According to Reis [1], the shape of the rear of the “rabo-de-bacalhau” is the result of the widening of the (side and interior) yard to the back of the building, meaning that the inner yard was completely merged with the open space in the rear. In various Lisbon blocks it is possible to observe the evolution of the inner yard from its beginnings, when it was enclosed in the centre of the building or adjacent to the side wall, to its complete opening out onto the open space in the rear (see left figures of Fig. 1). Figure 1 shows, on the right side, the basic type of “rabo-de-bacalhau” dwelling and four sub-types A, B, C and D. The base type includes features that are common to all sub-types.

This article focuses on the use of shape grammar in the rehabilitation methodology as a tool to identify and encode the principles and rules behind the adaptation of existing houses to new requirements and space syntax as a tool to evaluate the spatial properties of the proposed dwelling designs in the final layout and during the transformation.

## Shape Grammar and Space Syntax

Shape grammars were invented by Stiny and Gips [2] more than thirty years ago. They are “algorithmic systems for creating and understanding designs directly through computations with shapes, rather than indirectly through computations with text or symbols” [3]. Shape grammars are generative because they can be used to synthesize new designs in the language, descriptive because they provide for ways of explaining the formal structure of the designs that are generated and analytical because they can be used to tell whether a new design is in the same language. The process generated by shape grammars is not a deterministic one since it enables multiple designs to be generated, based on a single language but determined by different choices.

Space syntax was conceived by Bill Hillier and Julienne Hanson in the late 1970s as a tool to help architects understand the role of spatial configurations in shaping patterns of human behaviour and to estimate the social effects of their designs. In their theory, space is represented by its parts, which form a network of

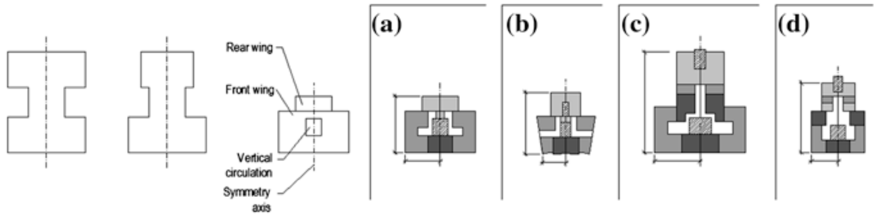


**Fig. 1** Subtypes of “rabo-de-bacalhau” dwellings (*right*) and the evolution of the building shape (*left*)

related components. The space syntax approach can be used both to understand how existing dwellings work and to simulate the likely effect of the proposed interventions. Both goals consist of a ‘diagnosis’ of the existing situations in order to identify a range of spatial factors that strongly influence the activities of the inhabitants as well as the social value of places.

In this research, shape grammar is used as a tool to define the principles and rules that compose the methodology for rehabilitating existing types and space syntax as a tool to evaluate the spatial properties of the existing and proposed dwelling designs. The combination of a shape grammar with an analysis tool such as space syntax offers the possibility of producing rehabilitation projects that conform to the requirements stipulated by the inhabitants and the specifications set by the architect. In this context, space syntax is used to determine the universe of valid solutions generated by the grammar and to validate them in terms of social properties.

## Rehabilitation Methodology

The proposed methodology provides for different rehabilitation solutions according to variable factors.

The work started with an analysis of contemporary demands for dwellings and the development of a knowledge base for the existing ICAT sets for homes, to be taken into account in the application of the rehabilitation methodology.

We then proposed a hypothesis for such a methodology, based on the conceptual schema for the design process proposed by Duarte [4] for the mass customization of housing, following March and Stiny’s “Design Machines” [5]. According to this conceptual schema, the design process consists of two subprocesses: a formulation process that takes user and site data and generates a description of an appropriate house, and a design process that takes this description and generates a matching solution within a given design language. Accordingly, it was hypothesized that a rehabilitation methodology should encompass four steps, as shown in Fig. 2. The figure shows the flow of information and the corresponding decision-making steps.

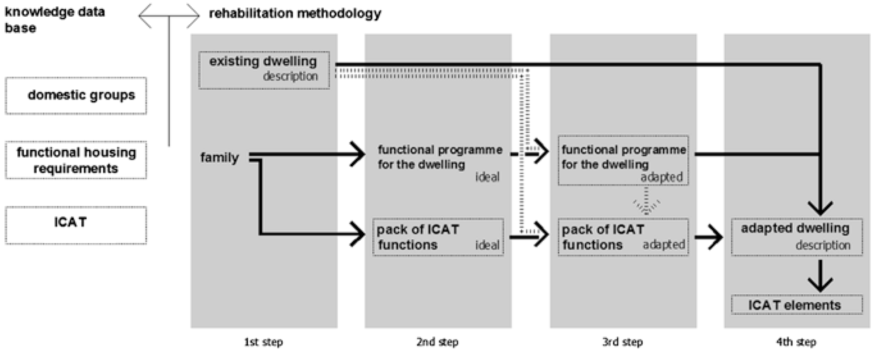


**Fig. 2** Basic steps in the planned rehabilitation methodology

Prior to the first step in the methodology a knowledge database was created, which plays a key problem-solving role [6]. It contains the following knowledge, which is required in order to perform the proposed rehabilitation: domestic groups—containing information on the various domestic groups and their characterisation; functional housing requirements—containing data on the functional and spatial requirements for the housing; ICAT—containing data on ICAT for housing.

This data constitutes the knowledge database for the methodology represented on Fig. 2. With this knowledge and a set of algorithms and rules which determine how to act on the information, a particular dwelling can be rehabilitated for a particular family.

The first step in the rehabilitation methodology consists of gathering the data needed for the rehabilitation process, namely a profile of the household and a description of the existing dwelling:

- Existing dwelling—a description (characterisation, analysis and diagnosis) of the existing dwelling. The existing dwelling represents the major constraint on the rehabilitation process;
- Family—a profile of the future dwellers. Characterisation according to kinship, number and age is a condition for the proposed solution. If the future inhabitants are unknown a typical family will be assumed, according to the profile defined by the client (property developer).

In the second step, the household profile is used to determine the ideal functional programme for the dwelling—following Pedro's [7] and Duarte's [4] work on the housing programme—as well as the ideal pack of ICAT functions:

- Ideal functional programme for the dwelling—this description comes from data obtained from the family and responds to their expectations and needs in terms of space. The functional programme in this case is a description of an ideal housing solution for the family that is not bound by any existing morphological structure or design language, since we are not yet considering the existing dwelling;

- Ideal pack of ICAT functions—this ideal ICAT description includes all the useful technologies for the family profile. Like the ideal functional programme, this ideal ICAT pack does not consider the existing dwelling and therefore does not have any functional or constructional limitations.

In the third step, the existing dwelling, the ideal functional programme, and the ideal ICAT pack are used to derive a description of a compromise or adapted solution based on the existing dwelling. Since the solution is influenced by the existing morphological structure, it is necessary to transform the description of the ideal solution obtained in step 2 into the description of the adapted solution. The phases are as follows:

- Adapted functional programme for the dwelling—in this phase, the data obtained from the ideal functional programme is combined with the characterisation of the existing dwelling in order to define the adapted functional programme. This adapted functional programme will need to accommodate the morphological and constructional restrictions on the existing dwelling whilst still meeting the family's requirements;
- Adapted pack of ICAT functions—in this phase, the data obtained from the ideal pack of ICAT functions is combined with the characterisation of the existing dwelling in order to define the adapted pack of ICAT functions. This adapted pack will consist of a set of functions which can feasibly be integrated into the existing dwelling, whilst still meeting the family's requirements.

Finally, the layout of a design solution for the particular family in the particular dwelling is obtained from the description of the adapted dwelling, including the ICAT components needed in the dwelling. The data obtained from this final step is as follows:

- Description of the adapted dwelling—this phase is a result of the previous steps and consists of the final solution chosen for the existing dwelling;
- Description of ICAT elements—when the adapted pack of ICAT functions and the description of the adapted dwelling are known, a full description can be provided of the ICAT elements that are to be included, such as the layout of the wiring, sensors and appliances.

## Transformation Grammar

The proposed grammar will be called a transformation grammar, as it aims to transform dwellings to adapt them to contemporary user needs.

Within this context, the concept of transformations in design explored by Terry Knight in her study on stylistic changes in different periods in the work of artists such as the De Stijl work of Georges Vantongerloo and Fritz Glarner or the work of Frank Lloyd Wright is used as a starting point [8, 9]. This work, however, proposes a different approach, in that it aims not to generate the design of the dwellings or their design after transformation, but the principles that enable them to be adapted to new lifestyles.

The transformation grammar enables one specific dwelling to be transformed into another by applying transformation rules and rather than generation rules as in a traditional shape grammar. In this grammar there is no predefined initial shape but there are countless possibilities since the initial shape is the floor plan of the existing dwelling, which can have many specific and complex shapes. This makes the proposed grammar a transformation grammar and not an original or analytical grammar.

The use of a shape grammar enables existing houses to be transformed in a very precise and systematic way. This process was used to manage shape transformation within dwellings to create a systematic and methodical process that could encompass all the valid transformation rules for a given dwelling. The transformations respond to functional and technical requirements as well as constructional requirements. This research explores a method which seeks to encode both the architect's knowledge and the knowledge acquired from other experiences of rehabilitation work in the form of rules.

## *Corpus of Design*

Although a Shape Grammar was not developed for the original "rabo-de-bacalhau" buildings as part of this study, a functional, constructional and social characterisation of the dwellings was produced. The functional and social characterisation provided an understanding of the logic of the spatial-functional organisation of the dwellings and the social reason for their existence, whilst the constructional characterisation allowed the constructional constraints affecting rehabilitation work on the buildings to be understood. Within the constructional context of the buildings and in the light of current concepts of rehabilitation work, the aim was for the proposed rehabilitation to follow the original language and not be too intrusive, essentially from a constructional point of view.

As our aim was not to produce a grammar of the original "rabo-de-bacalhau" buildings the original process of design was not considered in the grammar. It was also not our intention to develop a grammar for the rehabilitated ones. Instead our goal was to create a transformation grammar which encoded all the rules that allow a dwelling to be transformed. For that purpose the corpus used for the transformation grammar was the result of three sets of experiments done by several experimental subjects. The first and second set of experiments revealed different architectural approaches to the problem of rehabilitating "rabo-de-bacalhau" dwellings. These approaches, all done by architects, resulted in 40 processes of rehabilitated dwellings whose transformations formed the initial corpus of the grammar. The third set of results was performed by architectural students and provided 11 more transformation examples which were then incorporated in the final grammar.

## *Strategies of Rehabilitation*

The process of inferring transformation rules from the experiments carried out enabled two possible methods of transforming the dwellings to be identified. In addition, a third method of transforming the dwellings was explored in order to create smaller dwellings for households consisting of one or two people. Each of these three rehabilitation strategies has its own advantages and disadvantages in terms of functional and constructional aspects, and they can be combined within the same building to generate a wider market offer.

The buildings in the case study have 6–9 floors with a left-right symmetrical layout and 2 dwellings on each floor. This arrangement is repeated on all floors. The differences in the resulting transformations lie in the number of dwellings on each floor and the position of the kitchen in each dwelling.

The three strategies are as follows:

- Strategy 1: Maintaining two dwellings on each floor and moving the kitchen from its original position in the rear wing of the building to the front of the building. The aim is to strengthen the relationship between the social and service areas and to segregate the private area from the rest of the dwelling.
- Strategy 2: Maintaining two dwellings on each floor and the position of the kitchen. The aim is to keep construction transformations to a minimum without compromising the use requirements established in the functional programme. This strategy can be used to rehabilitate just one dwelling in the entire building.
- Strategy 3: Dividing one dwelling into two smaller ones and creating a kitchen in one of the new dwellings. The aim is to obtain smaller dwellings and a variety of dwelling types within the building.

Figure 3 shows the simplified derivation and decision tree for the three rehabilitation strategies described above.

For reasons of space and clarity of information, the decision tree shows only some of the transformation options for each strategy. The first transformation step shown is the choice of strategy, according to the resident/developer’s objectives, and consequently the definition of the kitchen’s place (for strategies 1 and 2). It then shows the option of defining the social area for strategies 1 and 2 and their basic variations. For strategy 3, the second step consists of choosing between the three methods of dividing up the dwelling to create different types of housing and the third step is the definition of the social area. The third step (strategies 1 and 2) or fourth step (strategy 3) shows the definition of private space, which may have various configurations. The tree aims only to simulate the derivation by considering the main functional areas and not the specific rooms, with the exception of the kitchen, which is the major defining space in the service area.

The graphs which accompany the tree in Fig. 3 are simplified and provide a general view of the situations illustrated.

Figure 4 shows examples of the transformation of an existing dwelling into new dwellings according to the three strategies of rehabilitation.

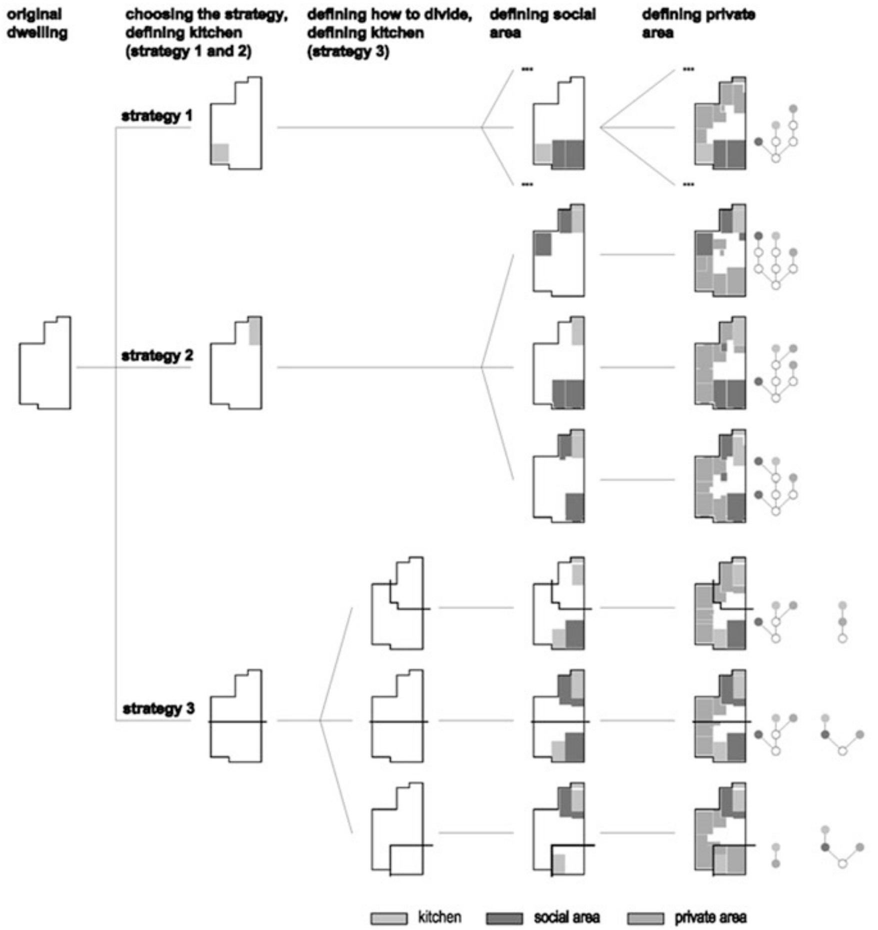


**Fig. 3** Simplified derivation tree for the different rehabilitation strategies

## Elements of the Grammar

The proposed grammar is a compound grammar defined in three algebras: (i) the combination of algebras $U02$ and $U12$ in which points and lines are combined on a plane supports the graph definition (see Fig. 5a); (ii) the algebra $U12$, in which lines are combined on a plane (see Fig. 5b) supports the plan definition, (iii) the algebra $U22$ in which planar surfaces are combined on a plane (see Fig. 5c) supports the spatial void definition. The algebra $U_{ij}$ contains shapes which are a set of basic elements. These elements are points, lines, planes or solids that are defined in dimension $i = 0, 1, 2$ or $3$, and combined in dimension $j \geq i$ [10].

Labels add information not provided by shapes. In the proposed transformation grammar, shapes (algebra $U$), in the algebra $U12$, are combined with labels

**Fig. 4** **a** Original dwelling: rehabilitated dwellings, floor plan and graph, according to the three strategies; **b** 1st strategy; **c** 2nd strategy; **d** 3rd strategy

(algebra *V*) in the algebra *V*02 (see Fig. 5e), in which label points are used to define dwelling functions. These labels are "where" labels because they specify which subshape or subshapes a rule may be applied to and because their location in relation to the associated shapes is essential to their function.

The labels used define the existing and proposed dwelling functions. When the grammar is first applied, a label is attributed to each existing space in the dwelling, according to the following criteria: *hs* for habitable spaces; *nhs* for non-habitable spaces; *Xba* for existing bathrooms; *Xki* for existing kitchens; *Xla* for existing laundries; and *Xbc* for existing closed balconies.

After assigning the first labels to the initial shape (original dwellings) derivation begins and the labels are replaced by others indicating the main function for which

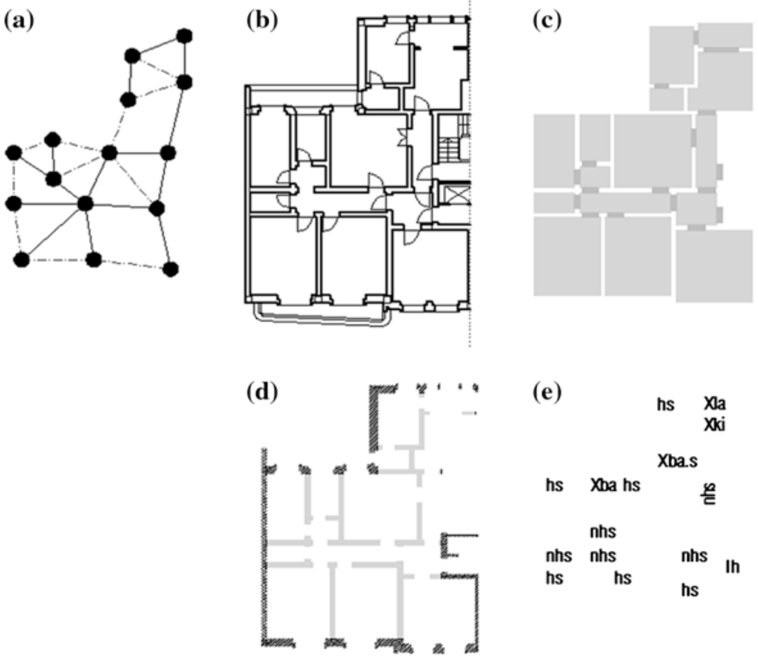


**Fig. 5** **a** Algebra *U02* and *U12* with dots and arcs; **b** algebra *U12* with the floor plan; **c** algebra *U22* with planar surfaces; **d** algebra *W22* for weights; **e** algebra *V02* for labels

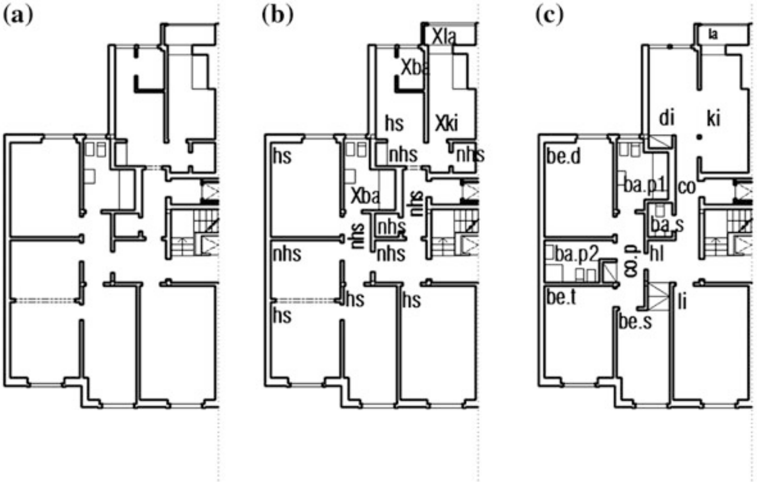


**Fig. 6** Floor plan representation: **a** without labels (initial shape); **b** with labels indicating whether rooms are habitable, non-habitable, kitchen, laundry or bathrooms; **c** with labels showing one final dwelling layout

**Fig. 7** Weights associated with the transformation grammar (*uba* and *uki* weights are differentiated with different colours in the original version of the transformation grammar [11])

the various spaces are destined (e.g. *be.d* for double bedroom or *li* for living room). At the end of the derivation, new labels should have been attributed to each of the spaces.

In order to define constructional constraints, weights [10] were used to incorporate shape properties.

The algebras $U12$ (for shapes) and $V02$ (for labels) are combined with algebra $W22$ (for weights) to obtain a new algebra used to execute the proposed dwelling transformation (see Fig. 5d). Weights are defined as planes with an interior filled with a pattern (see Figs. 6 and 7).

The attribution of weight to the shapes used in the proposed grammar is intended to characterise two different aspects: the existing and proposed construction systems for walls, and the acceptable position of new spaces within the dwelling.

For the wall construction systems four weights were used to represent four different types of wall. It was necessary to define these types of wall in order to establish rules that take construction and location requirements into account: existing brick walls (*ub*) these are the only walls that can be demolished; existing structural elements (*us*) these elements (front and back facade and inside columns) cannot be demolished under any circumstances; existing side walls (*usi*) these walls cannot be demolished; proposed light system (*ul*) new walls must be constructed as a light partition wall.

Two types of weights were used to identify the acceptable area for the allocation of new infrastructured spaces within the dwelling: appropriate areas for accommodating new bathrooms (*uba*), and appropriate areas for accommodating new kitchens (*uki*).

In addition to the floor plans, graphs were used to complement the description and evaluate spatial properties. In order to characterise the different forms of connections, descriptors were attributed in natural language (English), mathematical language, as graphs or as shapes. In this way, any of these forms of description can be used to indicate the topological properties of the rooms. Finally, graphs were also used to track the development of spatial relationships between rooms during the various transformation stages.

The conditions part, which is included in the shape rule, is essential in order to ensure that its use is limited to particular cases.

The conditions are divided into dimensional and functional conditions. The former includes minimum areas and linear dimensions and the latter an indication of the functions of the room referred to in the rule.

## *Descriptive Grammar*

A descriptive grammar consists of a description of shapes using symbols instead of shapes, as in a shape grammar.

Even though designs are technically shapes it is possible to use a recursive schema to describe them in terms of functionality, construction and meaning, among other factors which do not refer directly to their component spatial elements [12].

Stiny [12] states that design descriptions specify the relevant features and properties of designs in a finite way, determined according to certain relevant fixed criteria. Choosing these criteria is an important step in defining the description of the design, from the initial shape and throughout the derivation until the final design is completed.

In the present case the description must include building properties such as dwelling capacity, spaciousness and type, room function, area and dimensions, connections between different rooms and wall construction system and dimensions.

On the basis of these assumptions it can be said that the development of descriptive grammars involves two crucial questions: (i) the content of the description; (ii) the development of descriptive rules.

A transformation grammar allows for two types of description:

- The description of the design derivation, from the initial shape to the finished design, which includes all the criteria defined for the descriptions;
- The rules description, which may include some or all of the criteria selected for the descriptions (e.g. a rule that eliminates a wall may only include criteria for the construction system of the said wall, in addition to its dimensions and a definition of the spaces in which demolition takes place).

At the beginning of the definition of the descriptive grammar synthetic descriptions were developed in an abbreviated form. This abbreviated description proved to be insufficient in terms of implementing the grammar in computer software since it does not have all the information required. Thus, in a more advanced stage of the research, a detailed description was defined despite the computer software implementation were not a goal at the moment.

It was considered necessary to establish two parallel descriptions for the transformation grammar that could be combined to define the final design description:

- A description of the functional programme and ICAT pack, identified by the symbol β;
- A description of the dwelling, identified by the symbol α.

The proposed transformation grammar is a discursive grammar since it is able to generate syntactically and semantically correct designs that satisfy a priori requirements. This grammar is both a shape grammar (to generate legal projects) and a descriptive grammar (to generate adequate projects). These two grammars are combined in two stages of the rehabilitation methodology: firstly in the generation of the (functional and ICAT) programmes and secondly in the generation of the design solution.

In the first stage a programming grammar is used which involves processing the data for the family and the existing dwelling in order to generate the adapted functional housing programme and the adapted set of ICAT functions. Thus the programming grammar uses a descriptive part (regulations, recommendations and family data) as well as a shape part (the existing dwelling). The designing grammar generates a dwelling transformation using the previous data. The designing grammar uses a descriptive part (previously defined programmes) and a shape part (the transformation grammar).

This approach is similar to the one used by Duarte [13] in the Malagueira grammar except for the use of shape in the programming grammar since, in the present case, a specific dwelling is available from the beginning of the process and is used to define the adapted programmes.

The description of the functional programme includes the most relevant criteria that must be incorporated into the housing rehabilitation. These criteria were adapted from Duarte's [13] criteria for the housing programme for the Malagueira grammar (Fig 8).

## *Rules*

The different experiments undertaken during the process of inferring the rules of the transformation grammar revealed certain rehabilitation patterns. The decision-making processes used by the experimental subjects tended to be similar and to follow the same sequence for major decisions, for instance, the location of private and social areas.

As previously described there are three different rehabilitation strategies for the buildings in the case study, but the goals proposed in this study only included strategies that do not change the overall size of the dwelling, namely the first and second strategy.

The rehabilitation of a dwelling using this transformation grammar includes two major steps: (i) choosing an appropriate dwelling; (ii) adapting the dwelling.

After choosing an appropriate dwelling, the next decision involves adapting the family dwelling to the new functional programme and required ICAT pack.

| | Programming grammar | Designing grammar |
|---|---|---|
| Descriptive grammar | ✓ | ✓ |
| Shape grammar | ✓ | ✓ |

**Fig. 8** Programming and designing grammars (adapted from [13])

The adaptation of the dwelling includes 10 steps, numbered −1, 0, 1, 2, 3, 4, 5, 6, 7, 8, listed in Table 1. These steps may be divided into three different stages, firstly the preparation of the design (step −1), secondly the functional adaptation of the dwelling (step 0 to step 7) and thirdly the integration of ICAT components (step 8).

The first step (step −1) enables a compound representation of the dwelling to be generated and the transformation to begin.

The second step (step 0) enables the first important transformation decision to be made by locating the kitchen in accordance with the chosen strategy.

The third step (step 1) locates the hall. The hall function will be needed to define relationships with the functions of other rooms.

The following—step 2—may be (i) the definition of the private area if the functional programme asks for 2 or more bedrooms, (ii) the definition of the social area if the functional programme asks for less than 2 bedrooms.

Step 3 will be the definition of the private area or the definition of the social area according to what was generated in the previous step.

Step 4 is the definition of the circulation area. Step 5 is the definition of the service area and step 6 is the definition of the storage area.

Step 7 is the adaptation of the room's shape which may also be generated in each of the previous steps if there is no room to include the required rooms in Z.

The last step—step 8—is the integration of ICAT in rooms.

At the end of each step there is a rule which changes to the next step if the previous conditions have been met.

## Types of Rules

The strategy required to transform dwellings involves work on the walls. All the transformations proposed involve assigning functions to rooms and constructing or demolishing walls. The proposed rules therefore include the following types of rules: (i) rules for the assignment of functions to rooms; (ii) rules for permuting room functions; (iii) rules which add walls to enable rooms to be divided and wall openings to be eliminated or reduced; (iv) rules which eliminate walls to enable rooms to be connected or one room to be enlarged; (v) rules for changing the stage

**Table 1** Steps to follow in adapting a dwelling

| | | |
|---|---|---|
| **−1** Preparing the floor plan | −1.1 | Generation of a parallel representation—using surfaces |
| | −1.2 | Generation of a parallel representation—using graphs |
| | −1.3 | Attribution of labels for the spaces in the dwelling |
| | −1.4 | Attribution of weights for the walls in the dwelling |
| | −1.5 | Change from step −1 to step 0 |
| **0** Define kitchen /according to the chosen strategy | 0.1 | Assignment of kitchen (ki) (if strategy 2) |
| | 0.2 | Introduction of new kitchen position weight (if strategy 1) |
| | 0.3 | Assignment of kitchen (ki) (if strategy 1) |
| | 0.4 | Elimination of new kitchen position weight (if strategy 1) |
| | 0.5 | Change Xki label |
| | 0.6 | Change from step 0 to step 1 |
| **1** Assignment of hall | 1.1 | Assignment of hall (hl) |
| | 1.2 | Change from step 1 to step 2 |
| **2** Define private area (if functional programme has 2 or more bedrooms, if not go to Rule 3) | 2.1 | Assignment of bedrooms (be.d) |
| | 2.2 | Assignment of bedrooms (be.t) |
| | 2.3 | Assignment of bedrooms (be.s) |
| | *2.4* | *Change from step 2–step 7 if there are no spaces to satisfy rules 2.1, 2.2, 2.3* |
| | *2.5* | *Change bedroom assignment* |
| | 2.6 | Assignment of first private bathroom (ba.p1) |
| | 2.7 | Introduction of new bathroom position weight |
| | 2.8 → 2.15 | Assignment of second private bathroom (ba.p2) |
| | 2.16 | Elimination of new bathroom position weight |
| | *2.17* | *Permute functions* |
| | *2.18* | *Change from step 2–step 7 if there are no spaces to satisfy rules 2.8–2.15* |
| | 2.19 | Change from step 2 to step 3 |

(continued)

**Table 1** (continued)

| | | |
|---|---|---|
| **3** Define social area (if functional programme has 2 or more bedrooms, if not go to Rule 2) | 3.1 | Assignment of living room (li) |
| | 3.2 | Assignment of dining room (di) |
| | 3.3 | Assignment of combined living/dining room (li/di) |
| | 3.4, 3.5 | Assignment of home office (ho) |
| | 3.6 | Assignment of combined home office/living room (li/ho) |
| | 3.7 | Assignment of combined home office/bedroom (be/ho) |
| | 3.8 | Assignment of media room (mr) |
| | 3.9 | Assignment of combined living/media room (li/mr) |
| | *3.10* | *Change from step 3 to step 7 if there are no spaces to satisfy rules 3.1 to 3.9* |
| | 2.7 | Introduction of new bathroom position weight |
| | 3.11 → 3.14 | Assignment of guest bathroom (ba.g) |
| | 3.15 | Change Xba label |
| | *3.16* | *Change from step 3 to step 7 if there are no spaces to satisfy rules 3.11 to 3.14* |
| | 2.15 | Eliminate new bathroom position weight |
| | 3.17 | Change from step 3 to step 4 |
| **4** Define circulation | 4.1 | Assignment of private corridor (co.p) |
| | 4.2 | Assignment of social corridor (co) |
| | 4.3 | Assignment of service corridor (co.s) |
| | *4.4* | *Change from step 4 to step 7 if rooms with no connections are assigned* |
| | 4.5 | Change from step 4 to step 5 |
| **5** Define service area | 5.1, 5.2 | Assignment of laundry room (la) |
| | 5.3 | Change Xla label |
| | 5.4 | Change from step 5 to step 6 |
| **6** Define storage spaces | 6.1, 6.3, 6.4 | Assignment of clothes storage |
| | 6.2, 6.3, 6.4 | Assignment of general storage |
| | *6.5* | Change from step 6 to step 7 |
| **7***Adapt shape* | *7.1* | *Connect* |
| | *7.2* | *Separate* |
| | *7.3* | *Create circulation* |
| | *7.4* | *Change room size (add and subtract wall)* |
| | *7.5* | *Change room layout and assignment* |
| | *7.6* | *Change a door position* |
| | 7.7 | Change from step 7 to step 8 |

(continued)

**Table 1** (continued)

| | | |
|---|---|---|
| **8** Integrate ICAT | 8.1 | Allocation of detectors/sensors |
| | 8.2 | Allocation of alarms |
| | 8.3 | Allocation of CCVT cameras |
| | 8.4 | Allocation of controlled HVAC appliances |
| | 8.5 | Allocation of controlled lights |
| | 8.6 | Allocation of controlled blinds and screens |
| | 8.7 | Allocation of controlled doors and windows |
| | 8.8 | Allocation of controlled watering |
| | 8.9 | Allocation of controlled domestic appliances |
| | 8.10 | Allocation of interfaces |
| | 8.11 | Allocation of controlled sockets |
| | 8.12 | Allocation of ITED (Telecommunications Infrastructures in Buildings) sockets |
| | 8.13 | Allocation of Home cinema components |

Optional steps are in italics and mandatory steps in ordinary script

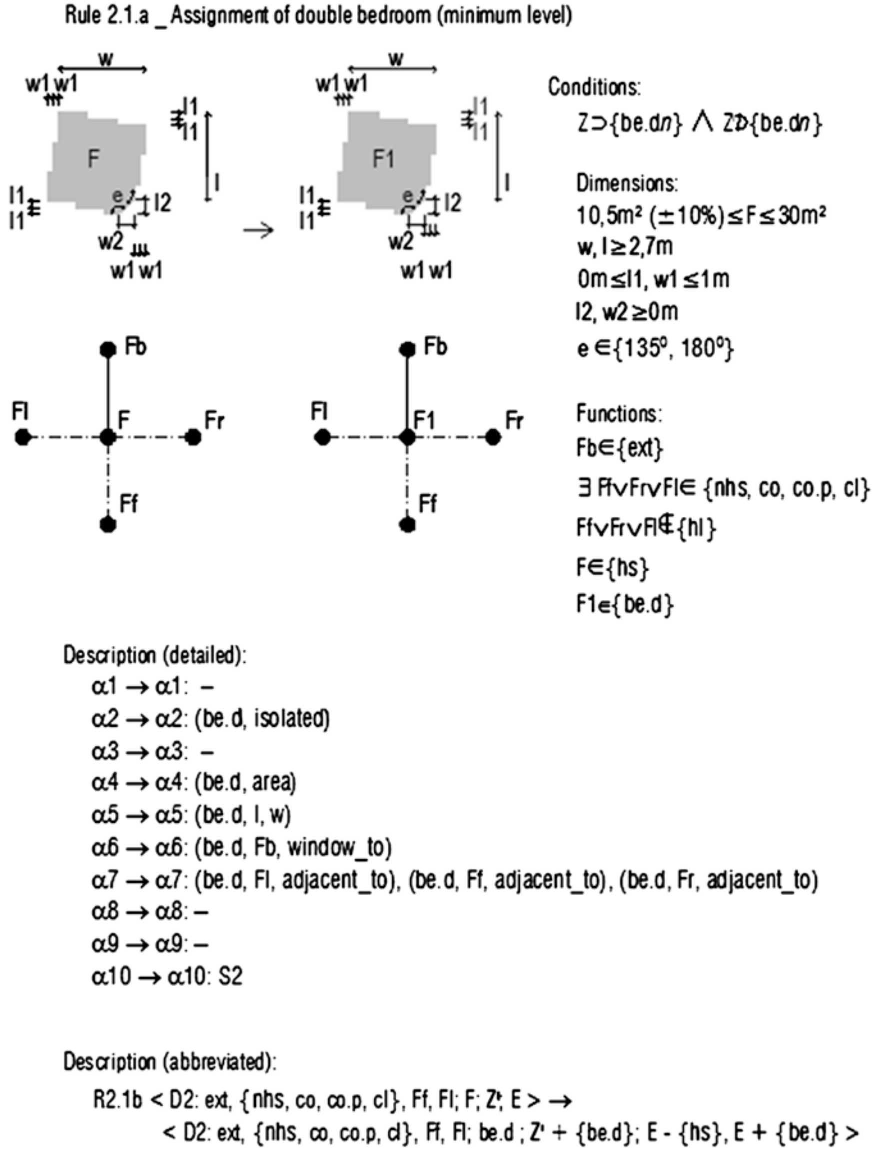
in the derivation; (vi) rules for preparing the floor plan; (vii) rules for integrating ICAT elements.

Next, examples of some types of rule are shown with the shape part, the condition part (functional and dimensional) and the description.

Assignment rules allow the functions and rooms required by the functional programme to be assigned to the existing rooms. The format for these rules is shown in Fig. 9. The detailed description of the assignment rule must include: the room to be assigned and its articulation ($\alpha2$); the net floor area of the room ($\alpha4$); the greater length and width of the room ($\alpha5$); the connections required between the room and the surrounding rooms ($\alpha6$); the adjacency of the room to the surrounding rooms ($\alpha7$); the stage in the derivation ($\alpha10$).

Rules for preparing the floor plan enables the floor plan to be prepared in order to begin the dwelling transformation. These rules include the following actions: generating a compound representation by adding dots, arcs and surfaces to the floor plan; adding labels to the existing rooms; adding weights to the existing walls.

The rules for adding labels have the format shown in Fig. 10. The detailed description of these rules must include: the room to which the label is added and its articulation ($\alpha2$); the type of connection between the rooms and the surrounding ones ($\alpha6$); the stage in the derivation ($\alpha10$).



Rule 2.1.a _ Assignment of double bedroom (minimum level)

Conditions:

$Z \supset \{be.d n\} \wedge Z \not\supset \{be.d n\}$

Dimensions:

$10{,}5m^2\ (\pm 10\%) \leq F \leq 30m^2$

$w, l \geq 2{,}7m$

$0m \leq l1, w1 \leq 1m$

$l2, w2 \geq 0m$

$e \in \{135^\circ, 180^\circ\}$

Functions:

$Fb \in \{ext\}$

$\exists\ Ff \vee Fr \vee Fl \in \{nhs, co, co.p, cl\}$

$Ff \vee Fr \vee Fl \notin \{hl\}$

$F \in \{hs\}$

$F1 \in \{be.d\}$

Description (detailed):

$\alpha 1 \rightarrow \alpha 1$: –

$\alpha 2 \rightarrow \alpha 2$: (be.d, isolated)

$\alpha 3 \rightarrow \alpha 3$: –

$\alpha 4 \rightarrow \alpha 4$: (be.d, area)

$\alpha 5 \rightarrow \alpha 5$: (be.d, l, w)

$\alpha 6 \rightarrow \alpha 6$: (be.d, Fb, window_to)

$\alpha 7 \rightarrow \alpha 7$: (be.d, Fl, adjacent_to), (be.d, Ff, adjacent_to), (be.d, Fr, adjacent_to)

$\alpha 8 \rightarrow \alpha 8$: –

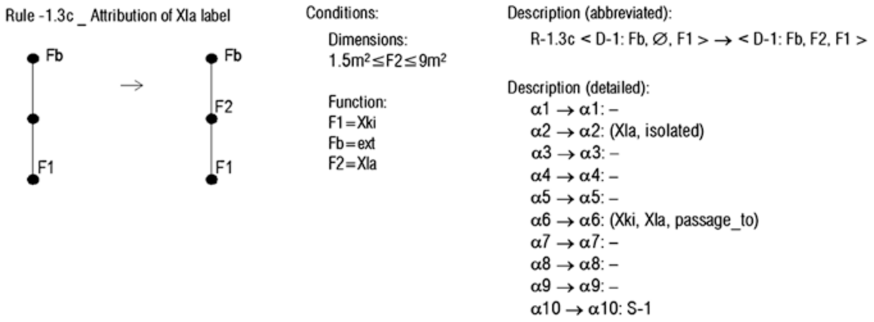
$\alpha 9 \rightarrow \alpha 9$: –

$\alpha 10 \rightarrow \alpha 10$: S2

Description (abbreviated):

R2.1b < D2: ext, {nhs, co, co.p, cl}, Ff, Fl; F; Z'; E > →
< D2: ext, {nhs, co, co.p, cl}, Ff, Fl; be.d ; Z' + {be.d}; E - {hs}, E + {be.d} >

**Fig. 9** Example of an assignment rule (Rule 2.1a), with shape part, condition part and descriptions (detailed and abbreviated)



Conditions:

Dimensions:
$1.5m^2 \leq F2 \leq 9m^2$

Function:
F1=Xki
Fb=ext
F2=Xla

Description (abbreviated):
R-1.3c < D-1: Fb, ∅, F1 > → < D-1: Fb, F2, F1 >

Description (detailed):
$\alpha1 \rightarrow \alpha1$: –
$\alpha2 \rightarrow \alpha2$: (Xla, isolated)
$\alpha3 \rightarrow \alpha3$: –
$\alpha4 \rightarrow \alpha4$: –
$\alpha5 \rightarrow \alpha5$: –
$\alpha6 \rightarrow \alpha6$: (Xki, Xla, passage_to)
$\alpha7 \rightarrow \alpha7$: –
$\alpha8 \rightarrow \alpha8$: –
$\alpha9 \rightarrow \alpha9$: –
$\alpha10 \rightarrow \alpha10$: S-1

**Fig. 10** Example of rule for adding labels to the existing rooms (Rule 1.3c), with shape part, condition part and descriptions (detailed and abbreviated)

## Conclusions

The main goal of this thesis was to define a methodology to support architects involved in the process of adapting existing dwellings and incorporating ICAT. The use of the proposed methodology allows for different rehabilitation solutions according to variable factors, namely the different families and their needs and lifestyles.

The proposed methodology therefore presents a precise sequence of steps which include the family as a key reference point in defining the housing programme and the ICAT pack to be installed. In the absence of a real-life family the methodology can also be followed by using a family type as a reference. This proposed process allows both for rehabilitation on a large scale in one or more buildings, or rehabilitation of only one dwelling, with customisation being the objective in both cases.

This customisation allows for a wider range of typologies within the building, in terms of the number of rooms per dwelling, the number of dwellings per floor and the areas available.

Shape grammar formalism was used as a tool in the rehabilitation of existing "rabo-de-bacalhau" buildings. In this context, formalism is used to encode the rules for transforming existing dwellings into new ones, adapted to contemporary life-styles. In addition, space syntax is used to form a parallel description grammar to guarantee that designs with a suitable functional organisation are generated. The resulting compound grammar is proposed as a way of developing and encoding a methodology for housing rehabilitation that can easily be explained to, and applied by, architects.

The use of a transformation grammar as an integral part of the rehabilitation methodology speeds up the decision-making process by providing a clearly defined framework for the work [13]. In this sense, the grammar enables the way on which decisions are made to be structured so that, in using it, designers can understand how to alter the dwellings according to the data in question. In this context the transformation grammar enables rehabilitation to be customised because it responds to specific requirements and may propose several hypotheses for dwelling layouts.

The transformation grammar is structured as a discursive grammar, which includes a shape aspect and a descriptive aspect that evolve in parallel to ensure that an appropriate dwelling design can be obtained from the description contained in the functional programme for the dwelling.

**Acknowledgements** The research described in this paper was funded by the Portuguese Science Foundation (FCT) through grant SFRH/BD/18225/2004.

## References

1. Reis NA (2000) O Saguão como personagem conceptual estruturador da habitação urbana: o caso de Lisboa—o seu passado recente e o seu futuro: trabalho de síntese. Lisboa, FAUTL
2. Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. In: Freiman CV (ed) Information processing 71, pp 1460–1465 (North Holland, Amsterdam). Available online: http://www.shapegrammar.org/ifip/ifip1.html. Last accessed 2009-04-01
3. Knight T W (2000) Shape grammars in education and practice: history and prospects. (online): Massachusetts: MIT, 14 September 2000. Available online: http://web.mit.edu/tknight/www/IJDC/. Last accessed 2009-03-21
4. Duarte JP (2001) Customizing mass housing: a discursive grammar for Sizás Malagueira houses. PhD thesis, Massachusetts Institute of Technology, Cambridge, EUA
5. Stiny G, March L (1981) Design machines. Environ Plann 8:245–255
6. Bento J, Côrte-Real J (2000) Tópicos de Inteligência Artificial. Notas de apoio às aulas. (online): Curso de Mestrado em Sistemas de Informação Geográfica, DEDCivil—IST. Available online: https://dspace.ist.utl.pt/bitstream/2295/67936/1/Inteligencia%20artificial.pdf. Last accessed 2008-12-02
7. Pedro JB (2000) Definição e avaliação da qualidade arquitectónica residencial. Thesis submitted for the Degree of Doctor of Architecture. Faculdade de Arquitectura da Universidade do Porto
8. Knight TW (1989) Transformations of De Stijl art: the paintings of Georges Vantongerloo and Fritz Glarner. Environ Plann B 16:51–98
9. Knight TW (1994) Transformations in design. A formal approach to stylistic change and innovation in the visual arts. Cambridge University Press, Cambridge
10. Stiny G (1992) Weights. Environ Plann B 19:413–430
11. Eloy S (2012) A transformation grammar-based methodology for housing rehabilitation: meeting contemporary functional and ICT requirements. Thesis submitted for the Degree of Doctor of Architecture, IST, TU Lisbon
12. Stiny G (1981) A note on the description of designs. Environ Plann B 8:257–267
13. Duarte JP (2007) Personalizar a habitação em série: Uma gramática discursiva para as casas da Malagueira do Siza. Lisboa: Fundação Calouste Gulbenkian, Fundação para a Ciência e a Tecnologia

# A Generic Shape Grammar for the Palladian Villa, Malagueira House, and Prairie House

**Deborah Benrós, Sean Hanna and Jose Pinto Duarte**

**Abstract** Shape grammars are formulations consisting of transformation rules that describe design. Previous studies have focused on recreating the style of family-related solutions. This study does not aim to recreate a specific architectural style but is part of wider research aimed at inferring shape grammars. It is believed that more than one grammar can be developed for the same style, but no one has ever demonstrated this possibility. In addition, no one has ever developed a grammar that can describe more than one style. The aim of this work is to demonstrate both possibilities. Firstly, it proposes a shape grammar that can produce three different design styles, and, secondly, it uses a process that is distinctively different from other tested examples yet still produces the same corpus of designs. It also enables a new corpus of designs to be produced, which had not been possible using the previous (or original) grammars. A selected case study of three grammars, namely for Palladian, Prairie and Malagueira houses, allowed for comparison and observation of the different processes and shape rules and for a new set of rules to be proposed, combined in a shape grammar. This was followed by the recreation of a new subdivision type of grammar with a top–down approach and a set of generic design rules. The result is a generic shape grammar that enables three different house styles to be designed from the same formulation.

---

D. Benrós (✉) · S. Hanna
University College London, London, UK
e-mail: d.benros.09@ucl.ac.uk

S. Hanna
e-mail: s.hanna@ucl.ac.uk

J. P. Duarte
TU Lisbon, Lisbon, Portugal

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_18,

## Introduction

Shape grammars, like rule systems, are formalisms that allow rules to be combined in a structured way to create a set of designs that share a similar root, are part of the same family or, in stylistic terms, belong to a certain style [1]. Beirão et al. [2], working on urban shape grammars, proposed a generic grammar for design and illustrated an application for urban design. Stiny [3] showed how shape grammar rules may be classified into a finite set of rule types or schemas expressed algebraically to suggest how different grammars may share a common structure.

In the past, many different shape grammars were used to describe a certain architectural style or signature design. The concept was first introduced by Stiny and Mitchell [4, 5] as a response emerging from the architectural and computing world to address the study of grammars described by Chomsky [6]. The controversial Palladian grammar showed how a style could be recreated using a set of rules. This enabled a set of housing layouts originally designed by the architect and a new corpus of solutions to be produced which obeyed the same rules and were intrinsically related to the existing corpus.

The same grammar theory was later put into practice by Duarte using the work of a living master, the Malagueira housing project [7]. This allowed expert feedback on the rule inference process from the author and, also for the trial and criticism of the new solutions by the creator of the system, as a quality control check.

This study does not aim to recreate any specific architectural style previously tested by others but is part of wider research aimed at inferring generic rules and exploring a generic grammar.

No one has ever tried to create two grammars for the same style. Usually, when capturing a particular style, researchers try to create the most elegant grammar, that is, the one that contains the smaller number of rules. Once a valid grammar is developed for a particular style, the need to create another, less elegant grammar becomes redundant.

In addition, no one has ever developed a grammar that can describe more than one style. It is believed that a grammar needs to be exclusive of a particular style or it fails its descriptive, analytical, and synthetic purposes.

This work aims to demonstrate the possibility of developing more than one grammar for the same style and the possibility of one grammar being able to generate more than one style. Firstly, it proposes a shape grammar that can produce three different design styles and, secondly, it uses a process that is distinctively different from the shape grammars previously inferred for these styles, yet still produces the same corpus of designs. It also enables a new corpus of designs to be produced, which had been impossible using the previous grammars (or the original forms).

The methodology started with the selection of three grammars for a case study, the comparison and observation of their different processes and shape rules and, finally, the proposal for a new set of rules combined into a new shape grammar.

The grammars selected reflect the three types of shape grammars identified and described in Knight's grammar classification as quoted by Prats [8]. The grammars, namely the Palladian villas by Palladio [4, 9], the Prairie houses by Wright [10] and the Malagueira houses by Siza [7], address the grid, addition and subdivision types respectively.

One important feature that influenced the case study selection was the fact that all the grammars are applied to single-family housing and, at the same time, featured distinctive differences in style, layout, history, and architect's background and signature, in addition to different geographic locations and periods of construction.

The next step involved comparing the study case data and identifying differences and similarities within the rule sets.

The proposal for the new set of rules addressed and encoded all three styles. The process selected was the one that seemed more flexible and allowed for the production of all three types—the subdivision process. The subdivision grammar type used is clearly distinct from previous attempts to recreate new Palladian-like designs since involves a grammar formulation and proposes a method combining parametric shape rules based on polygon division. This was the system used to generate the original Malagueira grammar, and also the process selected and described in an alternative Palladian grammar [11]. The alternative grammar for the Palladian villas grammar is, to the extent of our knowledge, the first attempt to propose a grammar that being intrinsically different in structure, rule formulation, and derivation process, still allows the design of the same corpus of designs described by the original grammar, the existing corpus designed by Palladio and a group of villas not addressed by the original grammar. The grammatical structure developed for the Palladian Villas, a top–down subdivision of rectangles, was then applied to the Prairie houses. The house starts with the inclusion of a house boundary and production opts for a top–down approach to the generative process. The system proposes a self-contained boundary shape that encompasses the limits of the building work. Subdivision starts as process of refining a succession of recursive divisions that generate different space subdivisions. A series of parametric shape rules are proposed to provide accuracy in the generation process. It is the manipulation of these parameters that enables a specific house type, either Palladian, Wright or Siza, to be produced. The rules are generic, but potentially specific. Within each rule the parameterisation assures that the right house type is designed even though the same rule is put to practice. Labels and other agents maintain and propose restrictions.

In addition to shape, function was also taken into account (unlike the Palladian grammar). The existing plans were observed and analysed in order to introduce functional spatial meaning. Segregation and integration were considered in the spatial assessment [12]. The spatial adjacency of each room, its dimensions and communication with other rooms were also considered.

As original research, the work aimed to automatically infer shape rules, starting by reflecting on three distinct shape grammars. All three original grammars were extensively tested and proved to be operational. However, the scope of their design was specific and bound by the limits of each designer. As formulations encompass a large amount of information, design know-how and consistent design style, they are not much use outside their corpus of action. This study attempts to make full use of the potential of these generative formulations, commonly known as shape grammars, by attempting to come up with a generic shape grammar formulation. The first step was to prove that one specific style can be produced by more than one formulation. The second step was to create one grammar that can produce more than one style, thus refuting certain assumptions concerning the hermetic nature of these structures. This paper is divided into five sections. The introduction is followed by a description of the case study and of the generic grammar formulation. The fourth section concerns the generation of solutions and derivations and the paper concludes with comments on the results and on future work.

## The Case Study and Grammar Comparison

Three grammars were used as a case study: the Palladian villa [4], Prairie house [10] and Malagueira house [7] grammars. The differences between them enriched the study and the similarities allowed for comparisons to be made, since all refer to single-family housing.

## The Palladian Grammar: A Grid Grammar

The formulation of the Palladian grammar atypically resembles the grid grammar type. It is atypical because in order to generate a grid it goes through an addition process whereby cells are added individually to the design. The basis of the design begins with the construction of a grid or matrix that reflects the interior design. This resembles a traditional orthogonal grid, even though it is constructed by adding individual cells.

The original and first grammar formulation for the Palladian grammar had eight design stages. The grammar formulation is illustrated in Fig. 1 as a grammar tree diagram.

The first stage combines ten rules for grid definition. These rules are additive processes that allow a grid scheme to be recreated by cell addition. The second stage proposes the introduction of a containing boundary shape and the inclusion of a single rule for defining the exterior walls. The third stage incorporates the room layout rules. This stage proposes seven concatenation rules. Special rooms are designed by joining consecutive cells to create larger spaces with spatial complexity. Stage 4 allows for wall realignment. At this point, the manipulation of the grid cell borders can be altered to create less rigid configurations. Stage 5 proposes 23 set of rules to introduce the main entrance. These are additive rules which add an obtrusion to the main envelope in the shape of a portico, loggia or

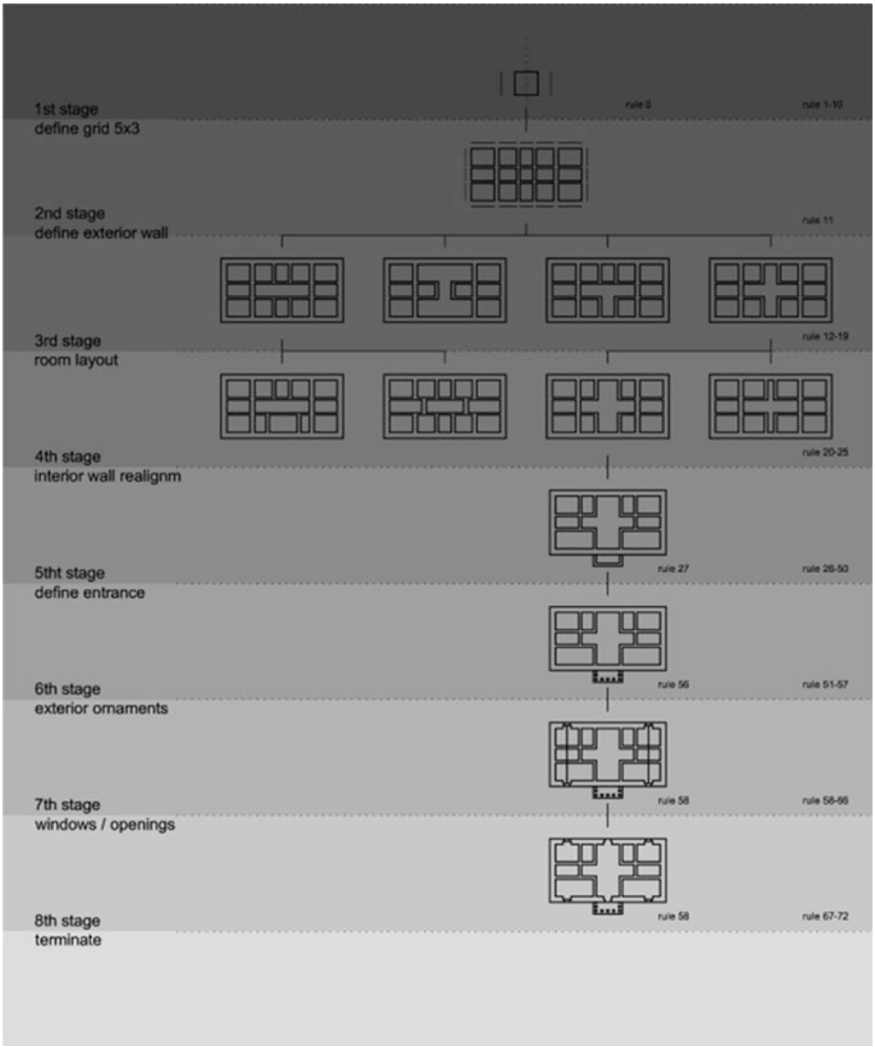


**Fig. 1** Original Palladian grammar tree diagram

ornamented column entrance. Stage 6 is dedicated to the decoration of the façade true to the classical style and is also an additive step that includes classical features. Stage 7 constitutes a typical subtractive process that involves creating openings in the walls for windows and doors. Stage 8 is a technical stage for deleting labels and terminating the design.

The grid grammar described effectively produces design solutions that fit the style criteria. The rules are coherent but the formulation does not necessarily follow an intuitive architectural design procedure.

## The Prairie House Grammar: An Addition Grammar

The Prairie house grammar was one of the first implementations of a 3D grammar and uses a typical addition process. The design follows the prescriptions of Frank Lloyd Wright and is initiated by the insertion of the main focal point of the family dwelling—the fireplace. Spaces are sequentially arranged around it in an orthogonal, biaxial manner, from the living and dining areas (the social areas) to the kitchen and pantries (the service areas) and the intimate family divisions (bedrooms, closets and bathrooms). This grammar structure is shown in Fig. 2.

The grammar is mainly composed of additive steps, as shown in the illustrative tree diagram. The first stage combines four rules for positioning the fireplace. Two possible fireplace placements are available. The second addition stage involves the insertion of the first social area adjacent to the fireplace. This constitutes another innovation in terms of the Palladian grammar, which did not address spatial function. There are four rules for this first room design. The third stage adds and extends the core living space, proposing the addition of an extra space adjacent to the one previously allocated.

The fourth stage is responsible for the so-called obligatory extensions to the social area immediately adjacent to the spaces previously added. It is another additive step. The fifth stage, with six addition rules, allows for the creation of functional social zones. The overall ground floor layout is then concluded. The seventh stage adds detail to address the style features of Wright's Prairie creations. The concave corners of the envelope are filled with exterior elements such as terraces, porches, verandas or small extensions decorated with features that are true to the original style.

The other rules are mainly subtractive and address certain issues relating to the ornamentation of this style.

Some corners are truncated, chamfered or simply subtracted to create the level of detail and refinement evident in the existing design corpus. The eighth stage terminates the design with the deletion of labels. Other stages are proposed by the grammar to design the top floor. However, as it is similar to the entrance floor and the upstairs design recursively repeats the main level procedure, the need to address this is minimal. It can be concluded that the Prairie house generation is mainly an addition process, with the exception of the detailing, for which uses subtraction as the operative means, and the termination stage. The design is conceived using a bottom–down approach in which a single element (such as the fireplace) is included at an early stage and the whole design subsequently develops from this. It is a very different approach to the grammar previously described in which the self-contained boundary is proposed at the outset.

This addition grammar drives its formulation directly from Frank Lloyd Wright's design concept and intentions. It is therefore intuitive and spaces are created in a sequential manner. The additive process seems to an appropriate choice which is easy to implement.

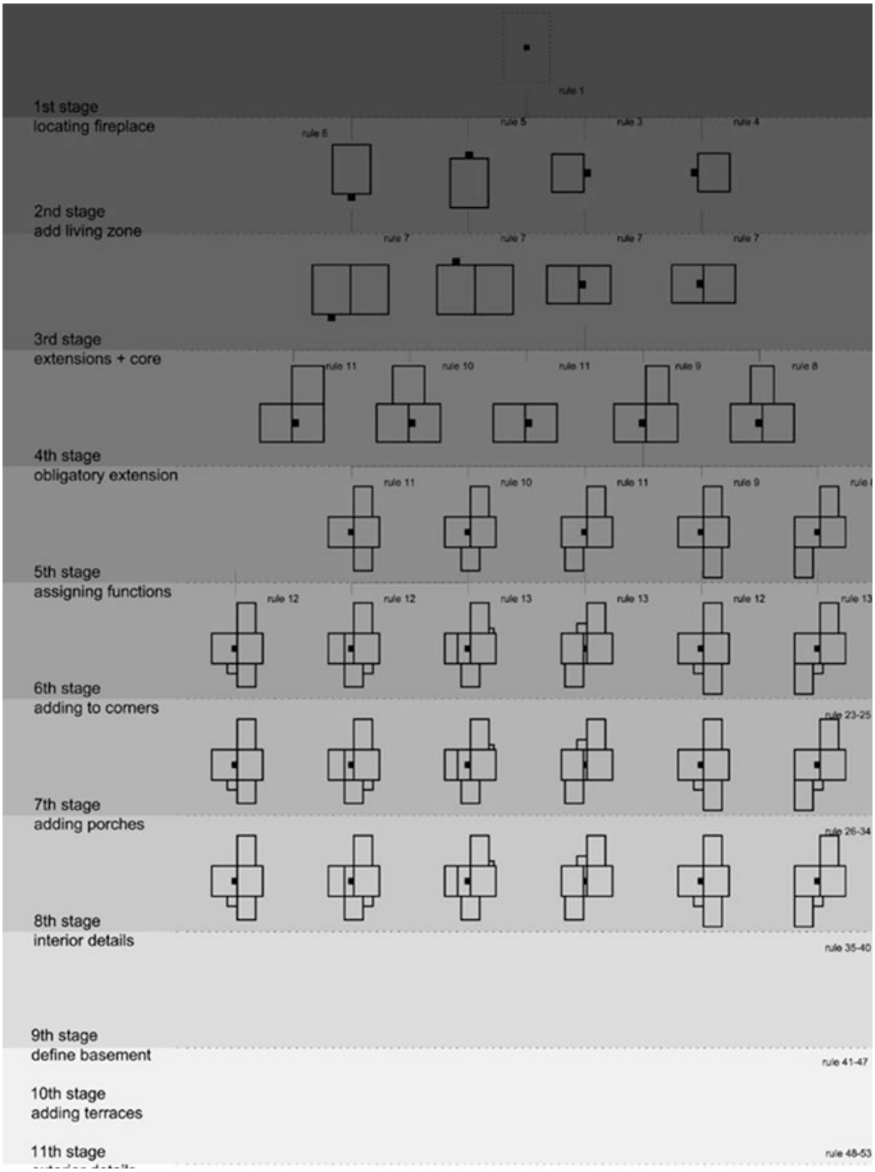


**Fig. 2** Original Prairie houses grammar tree diagram

## The Malagueira House Grammar: A Subdivision Grammar

The Malagueira house grammar uses a top–down approach and a subdivision grammar type to address plot and house design. The design is driven from the fixed plot space and adjacencies between surrounding buildings. The design concept inspired by Siza is based on the plot available. A percentage is allocated for exterior space and for the remaining indoor construction. The indoor construction is then detailed progressively through a series of divisions. The original grammar tree diagram can be seen in Fig. 3.

Although it consists mainly of subdivision stages, the first stage is an additive stage. The design is initiated by the addition of the plot layout and boundary, which establishes the design limits. The second stage is a fundamental step in the design concept. The grammar allows for two subdivision rules that determine the location of the house within the plot and the exterior yard. That is achieved by dividing the surrounding polygon into two smaller elements and is replicated further in the following steps with different aims. The third step starts detailing and assigning functional areas for the interior. Living spaces are designed and selected. The fourth stage, also proposing subdivisions, creates circulation spaces such as corridors or staircases.

The fifth stage subdivides the space further by creating and allocating service areas such as kitchens and pantries.

The sixth stage is the last subdivision stage and applies to small rooms and divisions, utilizing the remaining spaces. It is a design refinement stage that uses particular division rules.

The seventh stage breaks the routine proposing subtraction. It is a detain stage in which openings are created in interior walls for doors and in exterior walls to generate windows, and other elements such as risers and chimneys are detailed. The eighth stage terminates the design and erases the construction labels. Despite the approved efficiency and operability of this subdivision grammar, close observation led to the possibility of converting this into a grid grammar, as the floor plan resembles a complex grid plan. The key question concerned which grammar would be more efficient and thus more economical and the answer is probably subdivision, due to the level of grid complexity involved. However, this gave rise to the idea that more than one grammar could lead to the same design, as explained in the next section.

## The Case Study Analysis, Methodology and Comparison

The three grammars presented above describe three different processes and approaches to shape grammar inference. Firstly, they describe different content and styles. These differences derive from the fact that the existing designs were

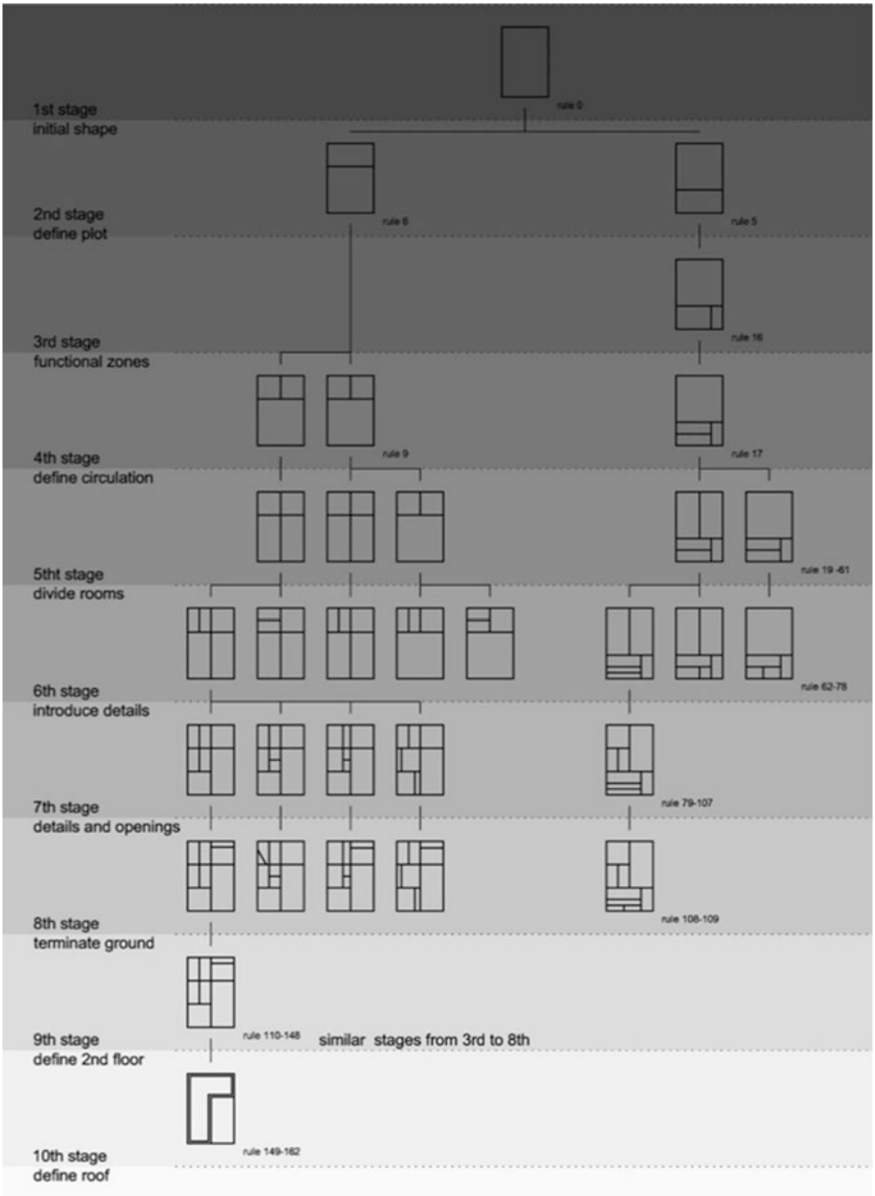


**Fig. 3** Original Malagueira houses grammar tree diagram

originally created by three different architects, of different nationalities, working at different periods, with dissimilar social and cultural backgrounds, specific clients, aesthetics and, more importantly, a characteristic, easily identifiable design

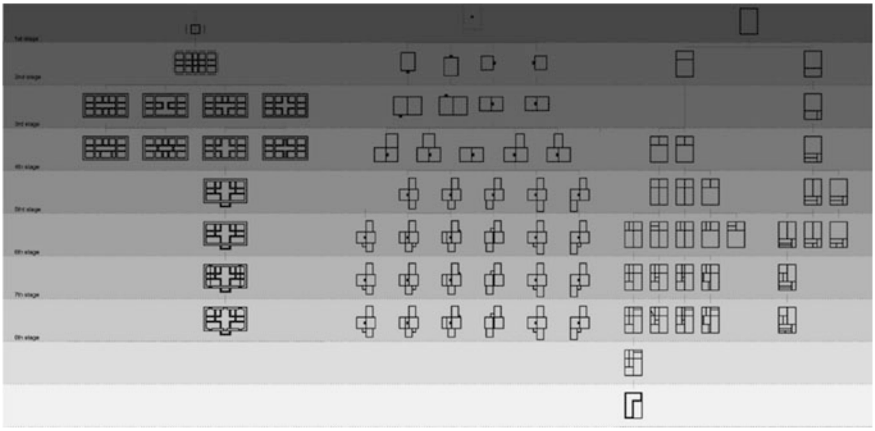


**Fig. 4** Grammar tree diagram comparison

language. The common ground they shared was that all the grammars focussed on single family housing and which allowed comparison.

Their processes follow different approaches. The Palladian villa and Malagueira house grammars use a top–down approach, starting with the general aspect of the building and detailing it as they progress, whilst the Prairie house grammar uses a bottom–up approach starting with one particular feature, 'the fireplace', and progressing to the other parts of the design. This is intrinsically related to the nature of each grammar. Grid and subdivision grammars first consider the big picture, either the boundary or design limits, and then work their way inwards, whereas an additive grammar typically starts with one of its parts and adds other elements sequentially. This also leads to another assumption, namely that grid and subdivision grammars often start with a self-contained element, whereas addition grammars is much harder to envisage the future containing shapes. If all three tree diagrams from each grammar are juxtaposed as shown in Fig. 4, the main differences and similarities between them can be seen. Despite the differences, all three grammars appear to start from a common point. They also present a similar number of steps from start to finish. The first stage is usually an addition process. This initial step always involves the addition of an element to start the design.

Likewise, the final step, involving the deletion of labels and the termination of the design, is very similar. The penultimate or seventh stage, which is the detailing stage and is subtractive, is also very similar in all the grammars. The main purpose of the detailing that occurs in these housing grammars is to create openings, such as windows and doors, in the walls. Other conclusions reflect the similarities between grid and subdivision grammars. They share the same top–down approach and main addition processes. Grid and addition grammars also have some formulations in common. As shown in the comparison tree, both have six additive processes in the early design stages. Despite the differences in the top–down and bottom–up approaches, both mainly use addition to achieve design solutions.

Close observation of the set of shape rules for the grammars revealed other similarities. Four types of rules were identified as very common: addition, offset, subtraction and concatenation. These could be observed countless times in all the grammars, sometimes involving certain particularities and on other occasions with different parameterizations. However, it seemed that a generic rule could be implemented to address all cases. The idea of a generic grammar now seemed feasible.

These core studies on each grammar also led to other conclusions. If so many similarities were found, it seemed only reasonable that there could be more than one way to generate a design. The Palladian grammar proved to be operative and effective in reproducing existing designs and generating a new corpus of solutions. However, it was less intuitive than expected and the grid creation process did seem particularly complex at times. A grammar does not have to recreate the designer's vision and process to produce consistent results, but sometimes this can be an effective way of tackling design or even helping to create a more economical grammar. The idea of developing an alternative grammar for the Palladian grammar seemed feasible when the research indicated that there could be more than one way of achieving results. Due to the similarities between the grid and subdivision grammars and due to the intuitive manner in which subdivision operates, it was chosen for this endeavour. Furthermore, this methodology was also extended to the Prairie house grammar. However, it implied an added level of complexity to obtain the same corpus of results. The overall rules were reconfigured, re-planned and parameterized to accommodate all the examples from the three-grammar case study, as will be explained in the following section.

## Generic Shape Grammar Formulation

A generic shape grammar was developed for the creation of Palladian villas, Prairie houses and Malagueira houses, supported by the choice of a top–down approach using a subdivision grammar. A new grammar structure and a new set of parametric shape rules were developed.

The alternative grammar for the Palladian villa used a methodology developed by Duarte [1] for the Malagueira house and later adapted to an urban context for the Marrakech grammar [13]. The subdivision grammar famously implemented by Stiny in the Ice-ray grammar [14] encompasses a very simple shape rule set of sequential subdivisions of polygons. The grammar showed how a polygon could be divided in two, generating complexity just by introducing diagonal cuts into a rectangle and sequential cuts to triangles and pentagons. The end result was a complex ice-ray window frame that resembled the traditional Chinese ice-ray lattices. This proved how a simple five-rule grammar could lead to an almost endless set of new solutions. The Malagueira grammar used a similar process, starting with the boundaries of the plot and evolving through subdivision into smaller inner spaces.

Since Palladio's villas always propose a rectangular envelope geometry, and taking into account that the use of subdivision is a common design choice for architects, it was the process selected to accomplish this task.

A new set of rules was developed to address all three grammars. The selected grammar does not try to replicate the process of the designers' work or methodology. It aims for a generic formulation that can accommodate three instances of design that are not only independent but also dissimilar. It attempts to achieve a consistent solution as efficiently as possible, whilst also creating a system that is easy to use. The grammar adopts a top–down approach and uses subdivision. The new generic grammar allows for eight stages from start to completion and the design of three types of houses, namely Palladian, Prairie and Malagueira. Any misuse of the grammar or rule manipulation will generate a hybrid version of the houses. This does not mean that the grammar implies an intelligent and careful use. The parametric mechanisms embedded in the grammar account for that, avoiding the occurrence of undesirable hybrids that do not represent any of the three styles. The shape rules should therefore be used intelligently but not freely. The generic grammar does not classify as unrestricted since it does follow a strict ordering process and imposes several restriction in the shape of parametric rules according to Knight's classification [8]. As shown in the tree diagram, the first stage designs the envelope or container shape. In all cases it is a 4-sided polygon or rectangle. For non-rectangular envelope geometries such as those of the Prairie houses, further geometric transformations are required to detail the envelope. The second stage introduces the subdivision. Two rules can be applied at this stage, namely horizontal divisions or vertical divisions. Vertical divisions provide double divisions for cases like the Palladian villas, where bi-symmetry has to be observed. At this stage, both rules can be applied a number of times recursively to the point where the maximum grid with the maximum number of cells is reached (a condition for the Palladian villas) or the divisions reach the minimum space needed to accommodate a living or circulation area. Other proportional aspects are monitored to ensure that design consistency is maintained. At the end of this stage the outline of the interior layout should be patent or at least foreseeable. This outline will be made clearer in the next stage.

Stage Three encompasses two basic design rules of concatenation and cell merging or, to be more accurate, cell border deletion. Both rules follow the symmetry conditions for the Palladian villas and involve a simple formulation for the Prairie and Malagueira houses. Rules Four and Five present two simple border deletion methods for vertical and horizontal situations, as shown in Fig. 5. The final interior layout is now complete and the design progresses to other aspects of construction.

The fourth stage proposes wall thickening. Limits and spaces had previously been represented by bi-dimensional lines but at this point, Rules 6–9 are responsible for converting either lines or corners into proper wall representations with a double line to represent a specific thickness.

In the Palladian case, the thickness of the interior walls reflects the masonry technology used and they are relatively thick in comparison with the other

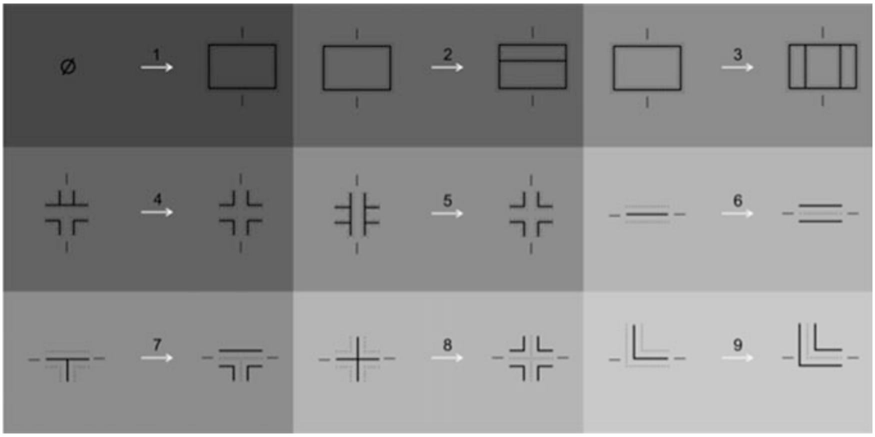


**Fig. 5** Generic grammar shape rules

examples, whilst the exterior walls are incrementally thicker. The rules address these differences and convert corners and intersections from 2D abstractions to standard thicknesses.

The fifth stage of the generic grammar proposes an addition step. This allows for certain features of each style to be added to the exterior, complementing the envelope with entrances, porticos, porches, exterior spaces and/or ornamentation. In the case of the Palladian villa, this is the stage when the entrance is defined and porticos and ornamentation, such as decorative classic columns or loggias, are added. For the Prairie house, elements such as the corner volumes occasionally fitted to concave corners of verandas or terraces are now incorporated. For the Malagueira houses, the small divisions or service spaces also added to concave corners can be introduced at this stage. This constitutes the second additive process in the procedure.

The sixth stage includes internal functions. The introduction of functions constituted an innovation in terms of the original Palladian grammar. Space and function is the issue covered least in Palladio's Four Books [9] and never addressed or labelled in plans or other drawings. It relates to the social nature of the 16th century aristocratic Italian villas built for the purpose of entertaining, whose main floors contained a series of rooms ranging from ante-chambers to ballrooms, drawing rooms, libraries and studies that are not always easy to identify. It was therefore necessary to research this area. Observation and comparison allowed for some spatial assumptions that were introduced into the generic grammar. Both the Malagueira and the Prairie house grammars addressed functions and originally proposed shape rules associated with special meaning.

The seventh stage constitutes a subtractive stage to introduce a greater level of refinement. At this stage openings are made in walls and interior walls can be removed to create internal circulation and incorporate doors. The exterior walls and facades can accommodate entrances and windows. Geometry, proportion and window positioning varies greatly from one house type to another. The rules that were written accommodate these differences and take into account the symmetrical features needed for the Palladian villa.

A specific function is associated with each space, created in accordance with the shape rules in order to maintain spatial flow and coherence and avoid the overlapping of functions, awkward adjacencies or spatial relations not envisaged as part of the original style.

The eighth and final stage completes the design by deleting the construction labels.

## Generation of Solutions and Derivation Process

The recreation of original designs using the method described above is illustrated in Figs. 6, 7, 8 and 9.

Derivation is the exemplification from start to finish of the phased application of the shape grammar rules. Often, the faster the derivation, the more efficient, elegant and easy the grammar is to use.

In this experiment three existing houses designed by the original architects were selected to illustrate the generic grammar. Villa Malcontenta is an example of a typical Palladian villa, the Robbie House, one of Wright's most famous creations, illustrates the existing corpus of Prairie houses and the Malagueira two-bedroom Ab type house (according to Duarte's labelling) exemplifies a typical Malagueira family housing unit.

## La Malcontenta Derivation

La Malcontenta was originally designed, built and completed in Venice between 1559 and 1560 and is pictured in the '*Il quatro libri*' [9]. Its orthogonal features and grid-like floor plan features a matrix that resembles a 5 × 3 grid organisation. Whereas the original grammar used a grid process, achieving the same design with subdivision allows us to economise on certain steps (namely extensive concatenation). The new tree diagram featuring the generic grammar is shown in Fig. 6 and the Malcontenta derivation in Fig. 7.

The envelope is thus designed and established from the start. This results in the first derivation step with the application of Rule 1 (adding a four sided polygon). Step 2 addresses the first main stage of the subdivision process by applying Rule 3, the vertical subdivision. As shown, this subdivision is doubled to address the symmetrical nature of the design. Steps 3–6 use the division rules 2 and 3 recursively (in the case of Rule 3, repeated again and again). Steps 7 and 8 start the space merging or concatenation process. This is a fundamental step for spatial configuration in a Palladian villa. The core space or social area is often the, largest,

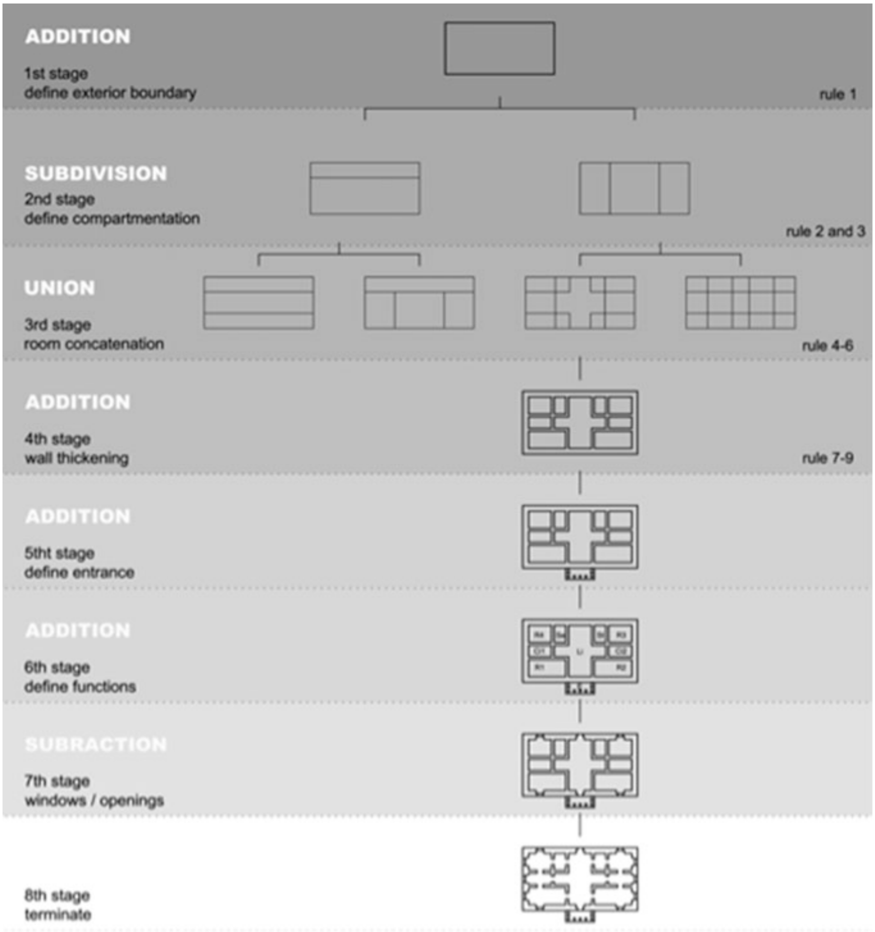


**Fig. 6** Generic grammar tree diagram—Palladio's Villa La Malcontenta

most central and geometrically complex area in the villa. This complexity is achieved by combining adjacent cells to form a broad regular polygon. Other larger rectangular spaces are allocated at the edges of the construction facing the facades.

With the layout settled, the 9th derivation step continues with the wall thickening, applying Rules 6–9. Step 10 adds new elements attached to the exterior of the envelope, namely the entrance portico.

The 12th step assigns functions for the spaces previously designed, and the next step creates detail and prepares the spatial articulation with the insertion of openings such as doors and windows. The villa is finished in Step 14 with the deletion of labels. In comparative terms, the derivation of the Malcontenta using this alternative method is faster than the derivation used in the original grammar.

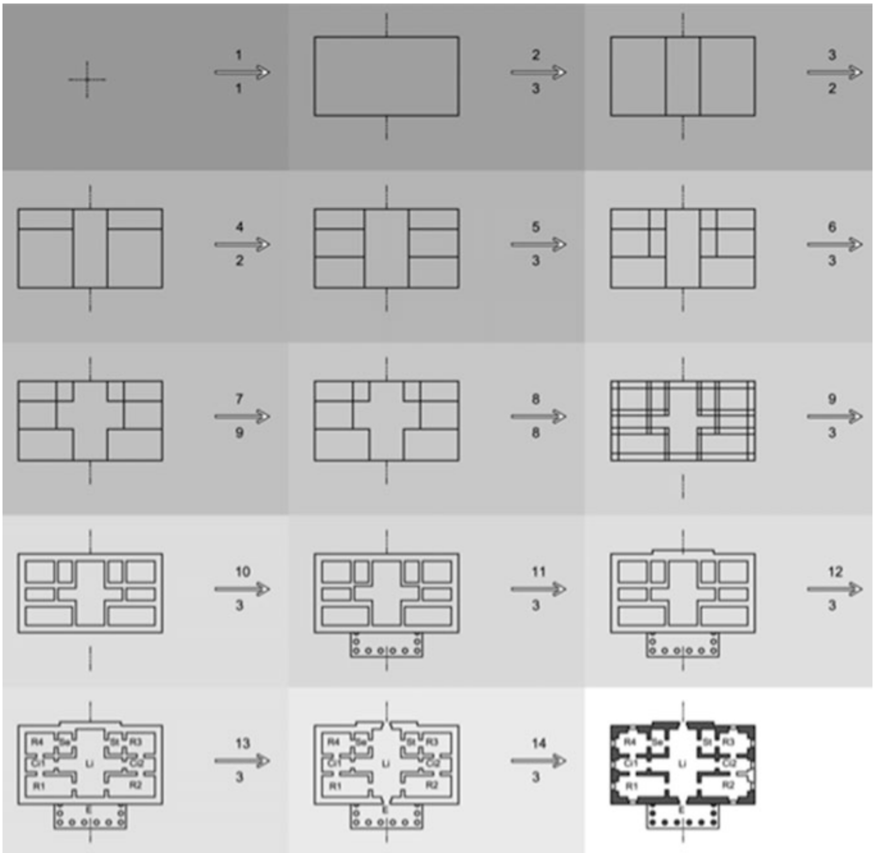


**Fig. 7** Generic grammar derivation—Palladio's Villa La Malcontenta

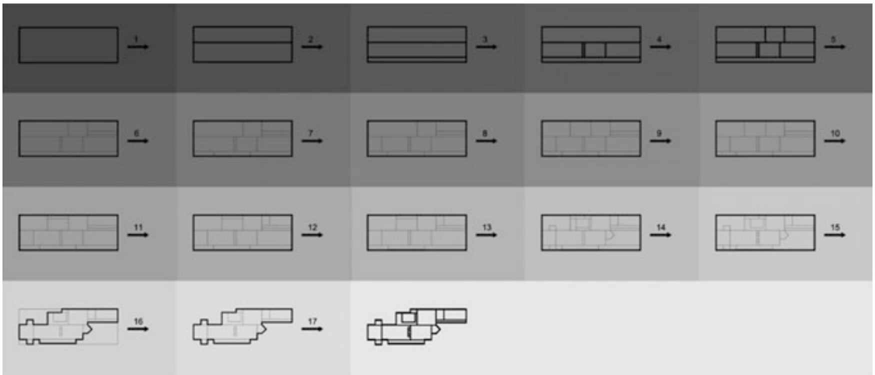


**Fig. 8** Generic grammar tree diagram—Wright's Robie house derivation

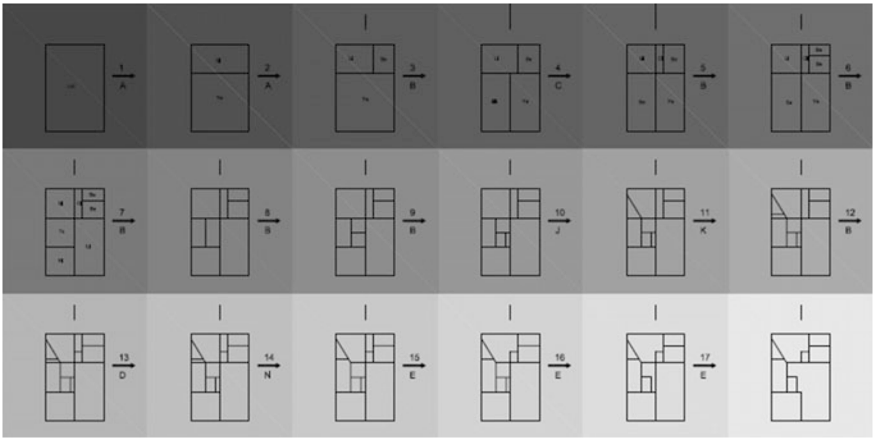


**Fig. 9** Generic grammar derivation—house type Ab derivation

## The Robie House Derivation

The derivation of Prairie houses using the generic grammar takes a certain level of abstraction into account (see Fig. 8). Most of the existing Prairie houses have a floor plan composed but polygonal shapes assembled orthogonally. With the exception of the rectangular, self-contained Wright Winslow House that can easily accommodate the rules and derive an operative alternative derivation in a few steps, most have a butterfly/crossed shape envelope, this is explained further by Granadeiro [15]. Prairie houses are created by the generic grammar using a container that extends to its extreme edges.

A new rule was introduced to erase the container edges after the core house was completed. For this reason, the example chosen to illustrate the Prairie house is the Robbie House, which is a singular, characteristic and well-known example. As in the Palladian example, the containing envelope shape is inserted in the first step using Rule 1. This rectangular shape does not contain the construction boundary, but illustrates a containing shape that encompasses its maximum limits.

Step 2 initiates the subdivision process. The horizontal and vertical division rules, Rules 2 and 3 respectively, are used recursively in several stages until all the floor plan lines have been replicated. Step 2, which extends to the next design stage, resumes the lengthy process of subdivision and is not the most intuitive or efficient of processes at times. The process combines analysis of the maximum lines (the lines that connect opposite sides) and their replication using subdivision. The trick lies in establishing the butterfly cross placing by using the double division first, then continuing to the finer details. Once all the lines are illustrated, the design is one step away from completion. At this point some adjacent cells need to be concatenated to remain true to the style, creating a flow of spaces that are oblong and have large areas. This constitutes Steps 9 and 10. In Step 10 all the basic interior design features are represented, including the container envelope shape. This shape is then modelled to fit the design criteria for the style. The unwanted edges are erased in accordance with Rule 10 and the line from the external vertex to the connecting indoors line is erased, leaving the end results as expected.

Steps 11, 12, 13 and 14 resemble much of what has been described previously and involve wall thickening, the detailing and creation of openings, and the completion of the design, respectively.

## Malagueira Two-Bedroom Single-Family House Type Ab Derivation

The derivation of Malagueira houses using the generic grammar involves a slight adaptation of the original Malagueira grammar rules. The original grammar is a typical subdivision grammar and, as explained, is the driving force behind the design of this generic grammar [7].

The example illustrates a typical two-bedroom, two-storey, terraced, semi-detached house, type Ab under the classification system devised by Duarte. The proposed derivation uses the subdivision rules previously explained, plus particular shape rules that address Siza's special configuration (Fig. 8). After the subdivision is performed, the steps that follow diverge from the original grammar and are closer to those tested in the previous derivations. Step 1 is the plot insertion, which involves applying a self-contained rectangular shape. In the case of the Malagueira houses the envelope shape is not parametric, but has a fixed size that reflects the available plot space with the same dimensions and area for each house. Step 2 applies Rule 3 for horizontal subdivision, segregating interior from exterior space. At this stage the yard/exterior space is allocated. Step 3 applies the vertical division, creating a division between the interior functional areas. The house layout now begins with the allocation of (service versus living) zoning. Due to the true nature of this subdivision, recursive vertical and horizontal divisions are performed to carry out the zoning and spacing. Steps 2–10 continue the recursive application of the division rules. Step 10 includes specific rules for the Malagueira design which replicate Siza's intention to create oblique cuts to produce smaller spaces and generate some spatial complexity. These rules are no more than parameterizations or generalizations of the division rules exemplified. The subdivision continues until Step 14.

In the 14th step concatenation is performed for the first time, offering the designer the flexibility to generate complex polygons with an incremental number of sides. This is replicated up to stet 17, where the design progresses as usual with wall thickening, detailing, the creation of openings and the completion of the design.

## Conclusion: Discussion of Results and Future Work

The work described above presents a generic shape grammar that allows for the generation of not one, but several signature styles. Unlike previous work, this is not a typical shape grammar, but a generic formulation that allows for the replication of more than one design style, which is believed to be a contribution to shape grammar research. To this end, the generic grammar uses shape grammar structure and shape rules. The rules are formulated as parametric and that can be manipulated to generate a particular design. A case study composed of three types of grammars, namely the Palladian villa, Prairie house the Malagueira house grammars, was selected to illustrate the scope of this generic grammar. The aim was twofold: firstly, to produce an alternative grammar that allowed for the alternative generation of a previously developed grammar, and, secondly, to use this new grammar as a generic grammar capable of producing more than one design style. The methodology started with a cross comparison of the grammars previously inferred and a study of their underlying styles. Each grammar was decomposed and its structure analysed. The complex sets of rules for each grammar were also analysed and similar rule formulations were pinpointed. The grammar comparison and knowledge acquired led to the idea of using subdivision grammars to construct the new generic grammar. This choice reflects the ease of use and intuitive nature of this grammar type and the adaptable nature of the subdivision process in comparison with other creative concepts. A new set of rules was developed for this new generic grammar in order to produce stylistically consistent designs. The set of rules incorporated important subdivision rules for the required conditions, such as minimum spacing and bilateral symmetry.

To the best of our knowledge, all the previous work on shape grammars has proposed a unique grammar to describe a particular corpus of designs. No one has proposed more than one grammar for the same style or a grammar that can describe more than one style. However, we believe that the effort of developing of an alternative grammar, different from the original grammar developed for that style, can tell us more about the essence of the style. We also believe that developing a grammar that can describe more than one style, which we call generic grammar, helps us to understand the commonalities among the different styles and the structure of the common underlying type. In this paper we present a grammar for single-family homes of three different styles.

This work refutes certain assumptions regarding shape grammars, namely the uniqueness of the design style that one grammar can produce. Given that there is more than one way to reproduce designs, more than one suitable grammar and that one grammar that can produce more than one style, many different representations are potentially viable.

This represents a breakthrough in shape grammar methodology and research. Shape grammars are no longer exclusive, but can potentially be manipulated to generate a larger corpus of new designs. This allows for efficiency in exploring shapes and analysing results, thus widening the scope of grammars.

Future work will focus on the effectiveness and implementation of the generic grammar, such as the exploration of a new corpus of designs and the analysis of generated design hybrids. It is expected that the mutation of these design styles or the overlapping of rules will produce new consistent designs with a new hybrid style. This is currently not allowed by the restrictions implemented. Moreover, computerised implementation will represent a positive development, allowing for the exploration of design solutions and even the enumeration of design corpus results. The potential of this generic grammar will be fully tested with a computerised tool, as was the case with previous work developed for housing shape grammars, such as the ABC system and the Haiti gingerbread house grammar [11].

**Acknowledgments** The authors would like to thank the contributions of Prof. Stiny and Prof Knight at MIT and Prof. Steadman at UCL at different stages of this research project.

## References

1. Stiny G (1980) Introduction to shape and shape grammars. Environ Plann 7(3):343–351
2. Beirão JN, Duarte JP, Stouffs R (2011) Creating generic grammars from specific grammars: towards flexible urban design. Nexus Netw J 13(1):73–88
3. Stiny G (2011) What rule(s) should I use? Nexus Netw J 13(1):15–47
4. Stiny G, Mitchell WJ (1978) The Palladian grammar. Environ Plann 5(1):5–18
5. Stiny G, Mitchell WJ (1978) Counting Palladian plans. Environ Plann 5(2):189–198
6. Chomsky N (1957) Synthetic structures. MIT Press, Cambridge
7. Koning H, Eizenberg J (1981) Language of the prairie: frank lloyd wright prairie houses. Environ Plann B Plan Des 8:295–323
8. Knight TW (1999) Shape grammars: six types. Environ Plann 26(1):15–31
9. Palladio A (1997) The four books on architecture. MIT Press, Cambridge
10. Duarte JP (2001) Customizing mass housing: a discursive grammar for Siza's Malagueira house. Available online: http://dspace.mit.edu/handle/1721.1/8189. Last accessed 10 June 2010
11. Benros D, Duarte JP, Hanna S (2012) An alternative grammar for the Palladian Villas shape grammar. In: Proceedings of the 17th international conference on computer aided architectural design research in Asia, Chennai, pp 415–424
12. Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge
13. Duarte JP et al (2006) An urban grammar for the medina of Marrakech. In: Gero JS (ed) Design computing and cognition 06, pp 483–502. Springer, Netherlands
14. Stiny G (1977) Ice-ray: a note on the generation of Chinese lattice designs. Environ Plann B 4:89–98
15. Granadeiro V, Duarte JP, Palensky P (2011) Building envelope shape design using a shape grammar-based parametric design system integrating energy simulation. IEEE Africon 2011—Zambia, pp 1–6

# Part VI
# Shape and Space

# Shape Interpretation with Design Computing

**Iestyn Jowers and Chris Earl**

**Abstract** How information is interpreted has significant impact on how it can be used. This is particularly important in design where information from a wide variety of sources is used in a wide variety of contexts and in a wide variety of ways. This paper is concerned with the information that is created, modified and analysed during design processes, specifically with the information that is represented in shapes. It investigates how design computing seeks to support these processes, and the difficulties that arise when it is necessary to consider alternative interpretations of shape. The aim is to establish the problem of shape interpretation as a general challenge for research in design computing, rather than a difficulty that is to be overcome within specific processes. Shape interpretations are common characteristics of several areas of enquiry in design computing. This paper reviews these, brings an integrated perspective and draws conclusions about how this underlying process can be supported.

## Introduction

Throughout a given design process, shapes are used in countless ways, and the different uses require different representation schemes to support necessary operations [1]. For example, shapes are used to represent the status of a design concept, as boundary objects that inform communication about a design, as models for analysing the performance of a design, and as instructions for physically realising

---

I. Jowers (✉)
Technische Universität München, Garching, Germany
e-mail: iestyn.jowers@pe.mw.tum.de

C. Earl
The Open University, Milton Keynes, UK

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_19,

a design. Often, the shapes in question are different views of a single design concept and before they can be used, interpretation into a suitable representation is necessary. Also, during a design task, shapes are interpreted and transformed according to the task's requirements. Information in shapes that is relevant to the task at hand is recognised, and then acted on.

In this paper, two general modes of shape interpretation are identified and explored: interpretation which is *visual*, and informs human performance in design processes, and interpretation which is *analytical* and informs transformations of descriptions used in computational methods. In both these modes, interpretation is concerned with explaining a shape either by applying a meaning or by identifying its structure or parts. In design research, the two modes are typically not explored in parallel, and instead investigations take place within localised contexts, such as conceptual design [2], or CAD/CAM [3]. However, they share strong commonalities and this paper aims to establish interpretation as a general problem for design computing, one that is common across design processes, rather than a local problem that is directly linked to specific contexts.

To this end, the paper presents a review of design research, set within a framework of visual and analytic modes of interpretation, with an emphasis on how humans and machines interpret shapes and apply these interpretations in subsequent operations. It aims to provide a general description of the role that shape interpretation plays, and highlight key similarities between different processes of design. These include the need to manage ambiguity and support the unexpected in design representations, and the importance of context and intended use in driving shape interpretation.

## Part 1: Visual Interpretation

As a visually creative activity, design is dependent on processes of perception—the shapes that surround designers inform and inspire them as they undertake design tasks. It is generally suggested that shapes are recognised and interpreted via decomposition into structural parts or features [4]. Understanding of a shape necessitates recognition of its parts and without this a given shape is an abstract entity void of meaning [5]. However, any given shape can give rise to countless decompositions into parts, and consequently countless interpretations. Also, these interpretations are susceptible to change from moment to moment; Wittgenstein [6] describes such interpretations as hypotheses regarding the structure of a shape, which may turn out to be false and are susceptible to change based on newly acquired evidence, or on the viewer's whim. Experienced designers learn to interact with shapes in this way, to visually explore alternative interpretations and structures [7], and this interaction has been linked to innovative design [8].

## *Interpretation in Conceptual Design*

The role of shape interpretation in conceptual design is well documented: shapes are used to form and inform representations of emerging design concepts. Typically such shapes are externalised using sketches, models, gestures, prototypes, digital tools, or verbalisations [9]. However, they are predominantly represented as sketches which are used to support shape exploration by representing particular aspects of a design concept. In this role, sketches are more than just static representations of imagined concepts; they externalise designers' cognitive activity and are used as devices to support exploration of an emerging design [10].

The shapes represented in sketches are inherently ambiguous, and this leads to a rich interaction between the designer and the shapes, what Schön and Wiggins [11] refer to as a reflective conversation between the designer and the media with which they are working. When shapes are viewed as abstract, ambiguity suggests alternative parts and structures that give rise to potentially countless interpretations [7]. When they are viewed as representing a concept, ambiguity enables designers to read off more than they put in [12]. Ambiguity makes it possible for the viewer to hypothesise about the meaning of a sketch, to interpret it based on context or according to their own knowledge and experiences. It allows designers to bring new insights into exploration process and supports the evolution of a design concept.

The kinds of interpretation used across conceptual design are varied. A given shape could give rise to figural interpretations; these are concerned with gestalts—coherent wholes that are defined by viewers' interpretations of the geometric elements that compose design representations. For example, the shape in Fig. 1 could give rise to figural interpretations as architectural plans or arrangements of tiles. Alternatively, a shape could give rise to other forms of visual interpretations. For example, the shape could be interpreted as a graph representing a schematic abstraction of an object, or as a collection of constructive elements, such as lines, triangles or squares. Alternatively, non-visual interpretations could arise. For example, the shape could be interpreted according to suggested functional properties, or could act as a metaphor for some alternative meaning or philosophy, such as motherhood or unity.

Studies of conceptual design identify the roles that these different kinds of interpretations play in design exploration. For example, gestalts result from interpretations that assign physical meaning to the geometric elements that compile shape representations, and are not fixed. The same set of geometric elements can be reconstructed as many different coherent wholes, and a designer often shifts gestalt during design exploration processes [13]. Gestalts enable designers to reason about design problems. Similarly, metaphorical interpretation is concerned with analogy—with creating a link from one concept to a (possibly indirectly) related second concept [14]. In conceptual design, metaphors enable designers to apply knowledge from a known situation to an unknown situation; they aid in the structuring of design problems, can contribute to unconventional thinking and

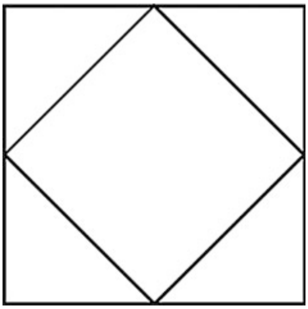


**Fig. 1** An ambiguous shape

stimulate innovation in design activities [15]. Analogies result from interpretations that assign comparative meaning to the geometric elements that compile shape representations.

## Interpretation in Computer-Aided Conceptual Design

The different kinds of shape interpretation used in conceptual design require methods of computational support that are not available in commercial computer-aided design (CAD) systems. This is evidenced in studies reported by Goel [10] and Stones and Cassidy [16], where designers undertook conceptual design tasks using either sketching or commercial computational tools. Both studies found that participants readily use shape interpretation in their design exploration if sketching, but not when using computational tools. Stones and Cassidy observe interpretation did take place cognitively when computational tools were used, but there was no evidence of these interpretations in the creation of new solutions. They suggest the reason for this is that, when participants were using computational tools, they were looking for accuracy in their design concepts and until a form closely resembled their mental picture they were unable to progress to alternative interpretations. Lawson and Loke [17] propose a more pragmatic reason and suggest that development of computational design tools has placed too much emphasis on graphical representation techniques. As such, the resulting tools are unable to support processes essential to creative design, including the process of shape interpretation as a means for supporting shape exploration.

Schön [13], discusses the potential for providing computational support for conceptual design, and distinguishes between methods that recognise designers' interpretations and those that support interpretations. This distinction is concerned with the difference between the semantics of shapes and the syntax of shapes. The semantics of a shape reflect the meaning that is associated with it, such as what it represents figurally, functionally, metaphorically, etc. As discussed, these are an important aspect of creative design, and build on designers' knowledge and past experiences—sources of information not necessarily evident in the shapes that are

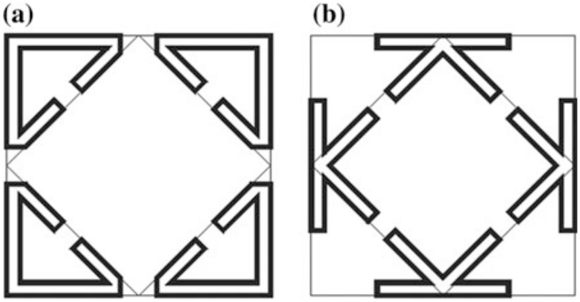


**Fig. 2** Interpretations of a shape, from Stiny (private communication)

used to support conceptual design, or apparent in the situation in which the process takes place. As such, the cognitive processes involved in this level of (semantic) interpretation are difficult, if not impossible, to formalise using computational methods [18]. Consequently, methods that seek to recognise designers' interpretations, such as Setchi and Bouchard [19], are necessarily restricted with respect to context; they provide only limited allowance for the unexpected and unknown, and as such their capacity for supporting innovative design is questionable.

Implicit in any semantic interpretation of a shape is a syntactic interpretation, i.e. a constructive interpretation of the geometric elements used to structure it [4]. For example, any figural interpretation of the shape in Fig. 1 necessitates a supporting syntactic interpretation, as illustrated in Fig. 2, where the shape is interpreted as an architectural plan in two different ways. The spaces that these two interpretations define are very different—in Fig. 2a, the shape is interpreted as four closed triangular wings overlooking an open quadrangle, while in Fig. 2b it is interpreted as a closed square hall with four open vestibules. In both of these examples, the semantic interpretations are implicitly dependent on different, and incompatible, syntactic interpretations of the same underlying shape.

During exploration processes, designers are continually making syntactic interpretations of the shapes with which they are working, in order to support their semantic interpretations. Schön [13] suggests that it is here that design computing can best support conceptual design by allowing shapes to be manipulated according to the parts recognised at any particular moment. However, supporting syntactic interpretations can be problematic because (1) parsing against syntactic structures is difficult and (2) the directions taken by designers in following different syntactic interpretations are wide ranging and surprising. Indeed, it is generally acknowledged that commercial design systems offer poor support for such syntactic shape interpretation, because the data structures on which they are built assume that a given shape has a unique interpretation [20]. This means designers have to adapt their design practice so that they are consistent with the particular systems that they use (and the underlying data structures), and this is often evident in a lack of innovation in shape exploration [21]. Research that seeks to address this problem has considered how shapes can be represented and queried so that the parts recognised by the viewer are apparent for manipulation, e.g.

[7], [20]. Interaction methods that allow designers to intuitively specify their interpretation of a shape at any given moment, according to recognised parts, have also been explored e.g. via sketch-based input [22] or eye-tracking [23].

## *Interpretation in Collaborative Design*

The processes used during conceptual design, to explore and develop design concepts, are not usually conducted by solo designers working in isolation. Instead, it is common for designers to work in teams that develop concepts in collaborative, social processes [24]. The shapes used to support these processes are varied, and include digital models, prints, physical models, flow charts, gestures etc., and these are used in various roles. For example, Ferguson [25] discusses three roles for sketched shapes as media for collaborative design: thinking sketches, talking sketches, and prescriptive sketches. Thinking sketches refer to sketches used in design exploration, and are interpreted in different ways to form and inform design concepts, as discussed above. Talking sketches support design communication in collaborative design. They act as conscription devices, that organise and store knowledge created through group interaction, and as boundary objects that support communication between participants of different disciplines [26]. In this way, shapes foster collaborative idea generation by providing a collective memory and by allowing team members to reflect on and interpret the ideas of other members [27]. Prescriptive sketches are used to record the outcomes of conceptual design, and are used to inform the representations that support downstream processes, such as analysis and fabrication.

The process of collaborative shape interpretation is not straightforward, and introduces additional issues over individual sketching processes. How shapes sketched by one team member are interpreted by others depends not only on form and context, but also on physical actions and social interaction. The process of constructing and transforming shapes conveys important information that is not necessarily apparent in the shapes themselves [28]. Also, when shapes are used to communicate design thinking, the apparent visual ambiguity gives rise to misinterpretations. Complications arise because participants in design teams do not necessarily see shapes in the same way. For example, Maier et al. [29] discuss the difficulties that arise when design engineers and simulation engineers communicate during design processes. Design engineers interpret shapes in terms of apparent geometrical structures, while simulation engineers interpret them in terms of functions, and this leads to difficulty in communication. Specifically, Henderson [26] notes that embedded within shapes are “codes” which are read by different viewers at different levels. These codes act as visual syntax or jargon and are defined within social structures, such as a design disciplines. Obvious examples include the standardised symbols used to annotate mechanical and electrical technical drawings. But, codes can also be more subtly embedded, for example in conventions that define how shapes are constructed and presented. This is

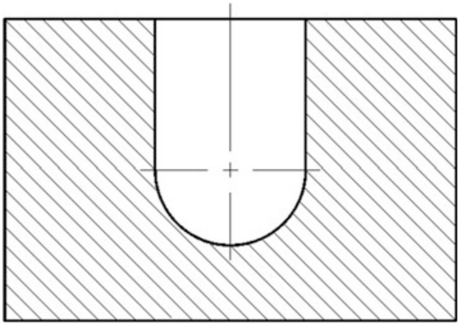


**Fig. 3** Technical drawing of a cylinder, from Henderson [26] (p. 56)

illustrated in Fig. 3, where the inclusion of a line closing a concavity indicates that the shape represents a cylinder, rather than a flat surface. Codes represent visual languages that are obvious to practitioners of a relevant discipline, but may not be obvious to less-experienced practitioners or outsiders. Therefore a shape which means one thing to one member of a design team may be read differently by other members.

Shah [30] suggests that misinterpretations that arise during design communication can be beneficial to a creative design process, since they can lead to exploration of unexpected ideas. This mirrors the use of interpretation in conceptual design but, for the sake of design communication, misinterpretation is not always beneficial. Indeed, Stacey and Eckert [31] emphasise that design communication should lead to a shared interpretation of a design. Ambiguity that results from incompleteness and provisional decisions is a necessary part of design, and can support a collaborative process. But, ambiguity that is cause by vagueness in shape representation can lead to confusion and should be avoided, since this can result in violation of decisions previously made and the contradiction of requirements or constraints. Similarly, van der Lugt [27] notes that lack of clarity about which parts of a shape are available for interpretation can lead to a lack of creativity in design collaboration. The possibility of disrupting the intentions of team members causes hesitation with respect to interpreting and modifying the shapes created by others. To avoid this, members of design teams seek permission to engage in reinterpretation of colleagues' sketches. Participants in a collaborative design process should clarify where negotiation in a design concept is possible, which elements of a shape are provisional, and which are constrained.

## *Interpretation in Computer-Supported Collaborative Design*

When designers work face-to-face, communication occurs using a patchwork of shape representations and human interactions, including verbal communication and body language. For example, gesture is an important communication device in collaborative drawing activities as a method for sharing interpretation of shapes [32].

Gestures can act as a collective memory and can support collective interpretation, in a manner similar to sketched shapes. But they are rarely used in isolation and instead are used to refer to objects, such as shapes or other members of the design team, and are generally accompanied with verbal explanation. This combination of interactions means that any misinterpretations that occur in face-to-face collaboration can be quickly recognised and corrected. Distributed design teams do not have this richness of representation to work with, and this can have a detrimental effect since the discourse that suggests how shapes should be interpreted is missing [28].

Current computational support for collaborative design does not adequately support these human factors of design communication [33]. 3D virtual worlds seek to address this issue by providing team members with avatars that provide a sense of shared presence [34]. But the interactions that these allow are limited and do not support the richness of face-to-face interaction. Indeed, the communication of distributed design teams is limited to the interactions that their tools allow. Because of this, use of communication tools to support design collaboration, necessarily affects design behaviour. For example, Maher et al. [35] compare design behaviour exhibited in face-to-face collaboration with collaboration in a remote sketching system and collaboration in a 3D virtual world. They discovered that collaboration using communication tools results in less time spent on the design process, more time spent on discussing software features, and a decrease in analysis-synthesis activities. They also found that the shape representations that the tools support modify the designers' interactions with them.

## Part 2: Analytical Interpretation

Problems of shape interpretation are also manifest in the computational methods and tools used to support design processes, such as computer-aided design/manufacture (CAD/CAM) systems. Such tools are developed for specific domains, such as mechanical engineering or architecture, and are used to construct, manipulate and interrogate digital models that represent design concepts. Here, interpretation is distinct from visual perception, and instead is concerned with analysing and transforming the descriptions of shapes so that the structures and parts necessary to carry out specific operations are defined. Problems arise due to the need to transfer data into, and between tools. This is because the different domains and different methods have different requirements with respect to the data used to represent shapes [3]. Integration of tools is highly desirable, since without it data needs to be transferred manually. This can be expensive, both temporally and financially, it is potentially disruptive to the design process, and increases the potential to introduce errors. Also, integration of representations is desirable so that design models can reflect the multiple perspectives, and multiple levels of detail, that are necessary to support multi-disciplinary design.

## *Interpretation for Design Analysis*

Throughout design processes, various methods and tools are used to analyse concepts against domain specific requirements. Central to the effective use of these methods is the problem of reducing the complexity of a design model so that desired properties are readily available for analysis. For example, in mechanical engineering, finite element analysis methods are commonly used to assess the structural properties of a design, such as strength [36]. This is achieved by interpreting the shapes in a model according to a simplified mesh of polygons, a process that is guided by the attributes of the original shape in combination with the specified goal of the analysis. In architecture, 'walk-through' and other simulations are used to assess spatial properties, such as 'flow' [37]. In this case, simplification of a model is achieved by defining key aspects of the simulation, e.g. points of interest in a crowd movement simulation or shadows for a light simulation. Similarly, visual aesthetic qualities are analysed, for example to ensure conformity to a brand or style [38]. Such analysis is achieved by identifying characteristic shapes, their allowable variation and the allowable spatial relation between them.

These three examples illustrate a spectrum of analysis problems, differentiated according to how quantifiable they are. Structural analysis exemplifies problems that are fully quantifiable and as such are commensurable with numerical methods of analysis, and implemented in computational systems. Aesthetic analysis exemplifies the opposite end of the spectrum, where problems are very difficult to quantify computationally, and require human interpretation of the results. Simulations lie between these extremes and can be used in distinct ways; firstly, to virtually test designs in use. This depends on human interpretation, and simulations should allow both realistic views by users as well as interpretations by them. The success of analysis can depend on the interfaces, and modes of interaction supported, e.g. the inclusion of user action or tactile feedback [39].

The second, more quantifiable, use involves determining an optimum or 'best' solution within given constraints and resources. In such problems, interpretations are expressed as shape parameters, and the values of these parameters are searched via simulation (or other methods of analysis), within the constraints, and to a given degree of accuracy. The possible designs generated are part of an 'object world' [24] instantiated for a particular project at a specific level of accuracy. These object worlds are context specific interpretations of potential designs, which designers manipulate and optimise. Optimisation can be employed in a wider context to search across design schema or configurations [40], as well as instances within a configuration.

Shape interpretations for design analysis do not end with properties and behaviour of the design itself. They are also key properties in the 'design for x' scenarios of manufacture, assembly and fabrication. These may be absolute in that designs may not be physically realisable, or relative in that the necessary resources are not available at the time. For example, if and how design shapes can be constructed or manufactured within cost and resource constraints is a critical analysis, often required at quite an early stage in the design process [41]. But analysis will yield more than a 'go/no go' result; its purpose is to provide routes to design improvement through understanding of possible changes to design components and assemblies. The initial analysis comes through a particular interpretation with incremental changes involving adjustment to this interpretation. It may yield performance outside acceptable margins leading to radical design changes with corresponding new association and analysis.

In the process of analysing designs, the design intentions, expressed through functions and requirements, play an important role. As designs evolve, analysis, assessment and evaluation determine alignment with intentions and requirements, which themselves evolve alongside design development. But functions and associated descriptions are wide open to interpretation themselves. Alink et al. [42] demonstrate the importance of interpretations in functional descriptions of mechanical devices, which are predominantly about shapes of components—their surfaces, interfaces and interstices. Functional descriptions correspond to shape interpretations. The wide variation in the functional descriptions observed in this study shows a broad spectrum of shape interpretations, which are possible during design processes. This exemplifies the different perspectives that various people, engineers, technical sales and marketing, for example, will hold. All these interpretations play into a design process and product evaluation. The diversity of descriptions integrated in design again points to the critical role of interpretation.

## *Interpretation as Feature Recognition*

The ways that digital shape data is presented throughout design processes varies. For example shape data may be presented as point clouds from laser scans of prototypes or as CAD surface descriptions from CAD processes. These different object descriptions pose their own issues for shape interpretation. In point cloud scanning there is no inherent surface structure in the acquired model [43]. On the other hand CAD surface descriptions are constructed from a series of shape elements and surface approximations. The scanned data may be grouped together in surface patches but a key issue, whether in point cloud or CAD surfaces is the relation between these geometric elements and the meaningful design and manufacturing features of a product or its components. In both cases this step is an interpretation from data to features.

Features are generic shapes used in computational tools, for supporting multiple shape interpretations. They are meaningful in specific application domains such as design, analysis or manufacturing, and they apply semantics to the shapes in design models, that reflect how those shapes will be understood in a particular process [44]. A given design model can be interpreted according to features in different ways, depending on the semantics that need to be represented. The resulting feature-based models are defined either according to a bottom–up or a top–down approach, using either design-by-features methods or feature recognition, respectively.

In design-by-features methods, features are used directly to construct design models. These features can be defined as shapes with specific significance, such as design components, but are more generally generic shapes, such as cylinders or rectangles [45]. Design-by-features methods allow designers to avoid tedious low-level shape definitions, and support spatial reasoning at a higher level of abstraction, whilst also conveying design intent. However, the set of features on which a particular method is based can never be comprehensive and can never support every conceivable situation. This is likely to feel restrictive to some designers, and has the potential to stifle creativity. Also, as a design is developed and modified, maintenance of features and the semantics linked to these features is a challenge [46].

Feature recognition is the problem of interpreting a given shape according to a defined set of features, i.e. the problem of recognising specific geometric shapes embedded in the representation of a design model [47]. This problem is complicated due to the possibility of multiple solutions, and due to the possibility of partial features (recognised by "hints"), which result from the interaction of features. It is a generally unsolved problem, and although various approaches have been defined and successfully applied, they are limited in application. Also, the shapes that can be considered are limited and the recognition of freeform features remains a challenge [48].

Feature-based models are domain specific. For example, the features used by designers to construct a shape are inherently different from the features used to define a process for fabricating the shape, as illustrated in Fig. 4. Because of this, features support different views of a product model, and feature-recognition suggests the potential for integrated design models by making available the information that is relevant for different design processes [49]. However, the domain specificity of features raises the question of how different feature-based interpretations of a model relate to each other—this is not always obvious and is of great concern when feature-based models are modified throughout a multi-disciplinary design process.

Feature mapping methods consider the problem of converting a model defined by one set of features into a model defined by a second set [47]. In theory it is a different problem to feature recognition since methods can take advantage of the features that already exist in the representation. However, there is little evidence to suggest that the information that is represented in a feature model for one domain is useful in another domain. Instead methods generally build on the underlying geometry, and integrated design models are defined by considering the mappings between individual features [50]. In this way, it is possible to manage multiple interpretations of a design model by propagating changes across feature models.

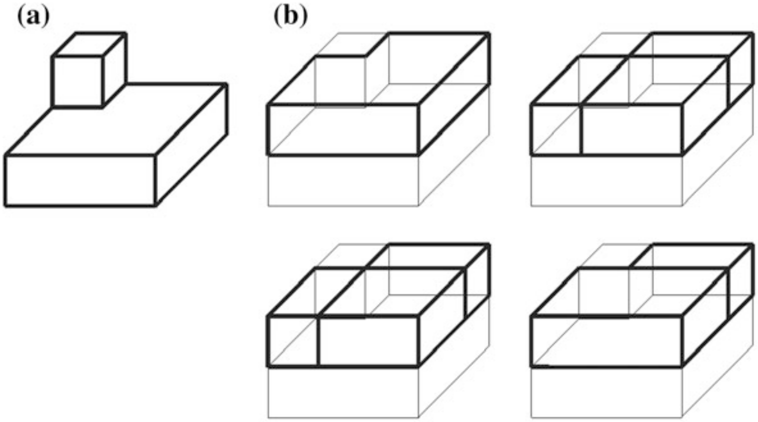


**Fig. 4** Feature interpretations, adapted from Corney et al. [3]. **a** Design features. **b** Manufacturing features, (i) one open pocket, (ii) two intersecting steps, (iii) three non-intersecting volumes, (iv) two maximal non-intersecting volumes

## *Interpretation for Fabrication*

Shape interpretation underlies how designs are evaluated, but designs are not just conceptions and models; their physical construction, through fabrication, demands yet another layer of shape interpretation. Interpreting a model as a specific fabrication processes that can physically realise the design can also be problematic. It is a computationally difficult process, as evidenced by continuing research into process planning, including the CAD/CAM interface [3]. This research aims to make the manufacture of products cheaper in terms of cost and time, by reducing the amount of human input necessary for process planning. This is particularly important as design moves towards a paradigm of mass-customisation and flexible manufacturing systems. Consumers are demanding more individually designed products, and the resultant costs of manufacturing are rising. Flexible manufacturing systems are explored that can provide the required variation at reasonable cost [51]. In particular, autonomous design-to-fabrication systems have the potential to meet the demand for rapid production of high quality products at low cost, by avoiding time-consuming manual re-planning [52].

In CAD/CAM, process planning is supported by interpreting a design as a feature-model, with features defined according to specific manufacturing processes such as milling or casting. Such features support the generation of process plans in computer-aided process planning (CAPP) systems. However, a given design can be interpreted according to manufacturing features in many different ways, as illustrated in Fig. 4b, and different interpretations correspond with different manufacturing processes. This leads to the unsolved challenge of determining the best feature interpretation for a given part and a given manufacturing process. The solution is not straightforward, and depends not only on the geometry of the given shape, but also on manufacturing information, such as tool type. Problems such as these mean fully autonomous process planning is still not possible and process planning tools still require an extensive amount of manual input regarding machine types, setups, fixtures, operations, cutting tools, cutting parameters etc. [53].

An advantage to having human input into process planning is that domain experts can control the details of the manufacturing process. Indeed, Corney et al. [3] suggest that automatic feature detection may not be necessary or required, since there is benefit for humans to make some decisions themselves. A less time-consuming approach would be to encourage designers to consider manufacturing processes as they compose a design shape. Features can be used to support 'design for manufacture' philosophies. Manufacturing features can be recognised as the designer creates a design, in order to identify potential manufacturing difficulties and evaluate alternative plans [41]. Alternatively, design-by-feature methods can be used to force designers to construct designs according to manufacturable features that can be easily recognised for process planning [54]. However such an 'object worlds' approach is characteristically deterministic, and is limited in application.

Rapid prototyping techniques, such as fused deposition modelling or selective laser sintering, provide a cost effective alternative to the more traditional methods of design fabrication [55]. They avoid the need to interpret designs according to features and support flexible manufacturing of complex forms, not realisable with traditional methods. This is because they use additive processes, where fabrication of 3D shapes is simplified according to a 2D layering process. There is limited need for CAM or CAPP processes or human intervention since the pre-processing of a shape simply involves tessellating a CAD model so that is can be efficiently interpreted according to horizontal slices. The only variations in fabrication relate to orientation of the design, which influences both build time and surface finish.

The rapidity and low cost with which physical representations of shapes can be produced using RP technologies provide a substantial reduction in product development time. Physical models are included in the shape exploration process; incomplete and provisional models are fabricated, assessed and visually interpreted, much in the same way that sketches are used [56]. They are used to visualise and physically explore concepts, to verify and optimise design parameters, to support iteration of design ideas, and to communicate those ideas with others.

## Discussion

Shapes have visual properties that lend an ambiguity and richness to creative processes such as design. These properties give rise to interpretations that inform the agents involved in such processes, by applying meaning (semantics) to the shapes and/or by identifying their parts and structures (syntax). The semantics of a shape are intrinsically linked to the context in which the shape is situated, along with its intended use. For example, figural interpretations of a shape depend on the viewers' understanding of the form it represents (as illustrated in Fig. 2), while the features that are important in the shape are dependent on the processes that are to be applied to it (as illustrated in Fig. 4). In general, the problem of identifying the context of a shape remains a formidable challenge, akin to the (as yet unmet and possibly unattainable [57] ) requirements of strong AI. As such, it is likely that human intervention will always be necessary to guide computational methods with respect to the context and intended use of a shape, and methods of human–computer interface that efficiently and intuitively afford such guidance should be explored. This human intervention is not necessarily undesirable, since it means that there is room for human expertise to inform computational design processes, respond to the unexpected, and resolve potential conflict [3], [31].

The syntax of a shape describes the structure of its representation and while visually it is linked to the semantics (and context) of the shape, in design computing it is separated. It is here that interpretation is a tractable problem for design computing [13], and it is also here that a clear distinction can be drawn between the two modes of interpretation, visual and analytic, that have been highlighted in this paper. Visual interpretation is based directly on perception, and there is no distinction between the visual shape and its representation, i.e. the shape *is* the shape. This means that, when designers use physical media (such as sketches, models, etc.) they are able to take advantage of the visual ambiguity and richness of shape, interpret it according to unexpected forms, varying contexts or intent, and in response, directly modify the shape. However, when computational methods are used to support design processes, the visual shape is a rendition of formal data structures. These structures have been developed based on the underlying problem of how to construct, manage and efficiently render digital models that reflect the forms that are apparent in the natural world. To this end, they succeed in representing highly complex forms with increasing accuracy and speed. But, the visual properties of shapes are not accounted for, since the data structures fix on one interpretation, leaving the ambiguity and richness of the visual shape absent. The shape is *not* the shape, but is a visualisation of a specific data structure. Different data can give rise to the same (visual) shape, and analytical interpretation is concerned with identifying transformations between these. Visual properties are typically not apparent in the data, and this can result in designers having to modify their practice to suit the computational tools that they are using [21]. For example, the geometry that can be used to define shapes might be restricted [54], and/or the transformations that can be applied to those shapes constrained [2]. Designers have to construct digital models in an 'object world' approach to meet a specific and limited purpose [24], and these models cannot be freely interpreted according to unexpected forms, varying contexts or intent.

So, at a fundamental level, the problem lies with the shape representations that have become standard within computational tools, e.g. boundary-representations (B-rep), which build on point-set topologies. In particular, there is a disconnect between the visual shape and the underlying representation. The point-set approach defines shapes according to symbolic structures, ordered according to the relationship of inclusion, and these do not reflect the perceptual characteristics of shapes. They do not afford the multiple interpretations that are needed to support design processes, and neither do they support the examination and re-examination of shapes to identify alternative parts and structures. Instead, alternative shape representations require investigation; representations that will afford development of tools that suit design practice, rather than designers having to modify their practice to suit their tools. For example, Salustri [58] suggests that the logic of mereotopology is a suitable alternative that formally describes “real” entities. In this approach emphasis is placed on the continuity of shapes, and the relationship of part-hood. Shapes are represented as occupying regions of space, and other concepts such as points, boundaries etc., result from interactions between these. There is a long philosophical and mathematical background in such ideas as a foundation for geometry including Whitehead [59], Clarke [60] and Gerla [61]. For design computing, the shape grammar formalism of Stiny [20] is based on a similar premise in which shapes are primarily structured according to parts, identified through the querying mechanism of shape rules. This formalism supports the reinterpretation of shapes, according to parts that are identified and manipulated via the application of such rules. Part-based topologies such as these support interpretation of shapes because they allow the structure of a shape to be defined according to whatever parts are relevant in the current (user-defined) context. These contexts can serve to ‘fix’ parts in the structure, forming the basis of a semantics of shapes situated in the current design context.

However, as suggested by Sloman [1], it is unlikely that any single representation scheme will sufficiently capture all the ambiguity and richness of shape, and instead it is likely that a variety of types of representation are necessary to support visual interpretation. A similar conclusion is suggested by Hanna [62] who reports that allowing high-dimensional representation of shapes, according to a variety of schemes, enables interpretation of that representation by an artificial agent. In other words, given enough representational data about a shape, relevant characteristics of the shape, such as neighbourhood type, can emerge. Hanna’s example relates to the classification of buildings, but a similar approach applied to other problems of shape interpretation may be possible.

The problem of interpretation itself is far wider than that considered here. All information that inputs to, is created in, and is manipulated by design processes is interpreted to accommodate specific uses. Stouffs and Krishnamurti [63] suggest that this general problem of information interpretation shares characteristics with the problem of shape interpretation. Accordingly, investigations into how shape interpretation can be supported can potentially inform the problem of how other forms of design information can be computationally represented and interpreted.

**Acknowledgments** Iestyn Jowers’ research is supported by CoTeSys, the Cognition for Technical Systems Cluster of Excellence (www.cotesys.org), funded by the Deutsche Forschungsgemeinschaft (DFG).

## References

1. Sloman A (1984) Why we need many knowledge representation formalisms. In Bramer M (ed) Research and development in expert systems, Cambridge University Press, pp 163–183
2. Prats M, Lim S, Jowers I, Garner SW, Chase S (2009) Transforming shape in design: observations from studies of sketching. Des Stud 30:503–520
3. Corney J, Hayes C, Sundararajan V, Wright P (2005) The CAD/CAM interface: A 25-year retrospective. J Comput Inf Sci Eng 5:188–197
4. Bruce V, Green PR, Georgeson MA (2003) Visual perception: physiology, psychology and ecology (4th Edition). Psychology Press, UK
5. Krstic D (2004) Computing with analysed shapes. In: Gero JS (ed) Design computing and cognition '06. Springer, Netherlands, pp 397–416
6. Wittgenstein L (1991) Remarks on the philosophy of psychology. GEM Anscombe, GH von Wright, H Nyman (eds). Wiley-Blackwell, New Jersey
7. Liu Y-T (1995) Some phenomena of seeing shapes in design. Des Stud 16:367–385
8. Suwa M (2003) Constructive perception: coordinating perception and conception toward acts of problem finding in a creative experience. Jpn Psychol Res 45:221–234
9. Wiegers T, Langeveld L, Vergeest J (2011) Shape language: how people describe shapes and shape operations. Des Stud 32:333–347
10. Goel V (1995) Sketches of thought. MIT Press, Cambridge
11. Schön DA, Wiggins G (1992) Kinds of seeing and their functions in designing. Des Stud 13:135–156
12. Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? a protocol analysis. Des Stud 18:385–403
13. Schön DA (1988) Designing: rules, types and worlds. Des Stud 9:181–190
14. Gentner D, Bowdle B, Wolff P, Boronat C (2001) Metaphor is like analogy. In: Gentner D, Holyoak KJ, Kokinov BN (eds) The analogical mind perspectives from cognitive science. MIT Press, Cambridge, pp 199–253
15. Casakin HP (2007) Factors of metaphors in design problem-solving: implications for design creativity. Int J Des 1:21–33
16. Stones C, Cassidy T (2010) Seeing and discovering: how do student designers reinterpret sketches and digital marks during graphical design ideation? Des Stud 31:439–460
17. Lawson B, Loke SM (1997) Computers words and pictures. Des Stud 18:171–183
18. Dreyfus HL (1992) What computers still can't do: a critique of artificial reason. MIT Press, Cambridge
19. Setchi R, Bouchard C (2010) In search of design inspiration: a semantic-based approach. J Comput Inf Sci Des 10:40–47
20. Stiny G (2006) Shape: talking about seeing and doing. MIT Press, Cambridge
21. Mitchell WJ (2001) Vitruvius redux. In: Antonsson EK, Cagan J (eds) Formal engineering design synthesis. Cambridge University Press, Cambridge, pp 1–19
22. Saund E, Moran T (1994) A perceptually supported sketch editor. In Symposium on user interface software and technology, ACM, pp 175–184
23. Jowers I, Prats M, McKay A, Garner S (2011) Design exploration with useless rules and eye tracking. In Culley SJ, Hicks BJ, cAloone TC, Howard TJ, Dong A (eds) ICED11—the 18th International conference on engineering design, The design society, pp 443–455
24. Bucciarelli L (1994) Designing engineers. MIT Press, Cambridge
25. Ferguson ES (1994) Engineering and the mind's eye. MIT Press, Cambridge
26. Henderson K (1999) On line and on paper: visual representations, visual culture and computer graphics in design engineering. MIT Press, Cambridge
27. van der Lugt R (2005) How sketching can affect the idea generation process in design group meetings. Des Stud 26:101–122
28. Nielson I, Lee J (1994) Conversations with graphics: Implications for the design of natural language/graphics interfaces. Int J Hum Comput Stud 40:509–541

29. Maier AM, Kreimeyer M, Lindemann U, Clarkson PJ (2009) Reflecting communication: a key factor for successful collaboration between embodiment design and simulation. J Eng Des 20:265–287
30. Shah JJ (1998) Experimental investigation of progressive idea generation techniques in engineering design. In Proceedings of ASME design engineering technical conference
31. Stacey M, Eckert C (2003) Against ambiguity. Comput Support Coop Work 12:153–183
32. Tang JC (1991) Findings from observational studies of collaborative work. Int J Man Mach Stud 34:143–160
33. Lang SYT, Dickenson J, Buchal RO (2002) Cognitive factors in distributed design. Comput Ind 48:89–98
34. Gu N, Kim MJ, Maher ML (2011) Technological advancements in synchronous collaboration: the effect of 3D virtual worlds and tangible user interfaces on architectural design. Autom Constr 20:270–278
35. Maher ML, Bilda Z, Gül LF (2006) Impact of collaborative virtual environments on design behaviour. In: Gero JS (ed) Design computing and cognition '06. Springer, Netherlands, pp 305–321
36. Shapiro V, Tsukanov I, Grishin A (2011) Geometric issues in computer aided design/computer aided engineering integration. J Comput Inf Sci Eng 11(2):021005
37. Pelechano N, Malkawi A (2008) Evacuation simulation models: Challenges in modeling high rise building evacuation with cellular automata approaches. Autom Constr 17:377–385
38. Li AI-K (2004) Styles, grammars, authors and users. In Gero JS (ed) Design computing and cognition '04, Kluwer Academic Publishers, Dordrecht, pp 197–215
39. Mengoni M, Colaiocco B, Peruzzini M, Germani M (2011) Design of a tactile display to support materials perception in virtual environments. IEEE Virtual Reality 2011:227–228
40. Cagan J, Mitchell WJ (1993) Optimally directed shape generation by shape annealing. Environ Plan 20:5–12
41. Gupta SK, Nau DS, Regli WC (1998) IMACS: a case study in real-world planning. Intell Syst Appl 13:49–60
42. Alink T, Eckert C, Ruckpaul A, Albers A (2010) Different function breakdowns for one existing product: experimental results. In Gero JS (ed) Design computing and cognition '10, Springer, Berlin pp 405–424
43. Fischer A (2011) Engineering-oriented geometry methods for modelling and analyzing scanned data. J Comput Inf Sci Eng 11(2):021002–021010
44. Regli W, Kopena J (2010) Challenges in semantics for computer-aided designs: a position paper. In AAAI Cognitive shape processing, Available online: http://www.spatial-cognition.de/CSP/Regli.pdf Last accessed March 2013
45. Venkataraman S, Shah JJ, Summers JD (2001) An investigation of integrating design by features and feature recognition. In IFIP Conference, FEATS, Valenciences, Frances
46. Bidarra R, Bronsvoort WF (2000) Semantic feature modelling. Comput Aided Des 32:201–225
47. Shah JJ, Anderson D, Kim YS, Joshi S (2001) A discourse on geometric feature recognition from CAD models. J Comput Inf Sci Eng 1:41–51
48. van den Berg E, Bronsvorrt WF, Vergeest JSM (2002) Freeform feature modelling: concepts and prospects. Comput Ind 49:217–233
49. Bronsvoort WF, Noort A (2004) Multiple-view feature modelling for integral product development. Comput Aided Des 36:929–946
50. Jha K, Gurumoorthy B (2000) Automatic propagation of feature modification across domains. Comput Aided Des 32:691–706
51. da Silveira G, Borenstein D, Fogliatto FS (2001) Mass customization: literature review and research directions. Int J Prod Econ 72:1–13
52. Shea K, Ertelt C, Gmeiner T, Ameri F (2010) Design-to-fabrication automation for the cognitive machine shop. Adv Eng Inform 24:251–268
53. Bourne D, Corney J, Gupta SK (2011) Recent advances and future challenges in automated manufacturing planning. J Comput Inf Sci Eng 11:021006–021016

54. Ahn et al (2001) Cyber cut: an internet-based CAD/CAM system. J Comput Inf Sci Eng 1:52–59
55. Yan X, Gu P (1996) A review of rapid prototyping technologies and systems. Comput Aided Des 28:307–318
56. Paterson G, Earl C (2010) Line and plane to solid: Analyzing their use in design practice through shape rules. In Gero JS (ed) Design computing and cognition '10, Springer, Berlin pp. 251–267
57. Searle J (1980) Minds, brains and programs. Behav Brain Sci 3:417–457
58. Salustri FA (2002) Mereotopology for product modelling: a new framework for product modelling based on logic. J Des Res 2(1)
59. Whitehead AN (1929) Process and reality. Cambridge University Press, Cambridge
60. Clarke B (1985) Individuals and points. Notre Dame J Formal Logic 26:61–75
61. Gerla G (1995) Pointless geometries. In Buekenhout F, Kantor W (eds) Handbook of incidence geometry, North-Holland, pp 1015–1031
62. Hanna S (2010) Design agents and the need for high-dimensional perception. In Gero JS (ed) Design computing and cognition '10, Springer, Berlin pp 115–134
63. Stouffs R, Krishnamurti R (2004) Data views, data recognition, design queries and design rules. In Gero JS (ed) Design computing and cognition '04, Kluwer Academic Publishers, Kluwer Academic Publishers, pp 219–238

# Algebras of Shapes Revisited

**Djordje Krstic**

**Abstract** This paper aims to contribute to a better understanding of algebras of shapes as they are applied in a design theory, which involves shape grammars. Three different formats of these algebras, which (surprisingly) differ in behavior, will be explored. Some earlier informal ideas on algebras of shapes will be revisited and a formal underpinning will be provided. In particular, two different sums of algebras of shapes will be defined.

## Algebras of Shapes: Alternative Definitions

Shapes are envisioned as abstractions of drawings and models designers use to specify designs. They are either finite arrangements of points, or lines, or planar regions, or solids, which are *basic elements* of dimension $i = 0$, 1, 2, and 3, respectively. These are defined in a space of dimension $j$, which is equal or higher than $i$. This classifies shapes based on the two dimensions $i$ and $j$ into sets of shapes $U_{ij}$. Shapes have parts, which are also shapes. Part relation can be intuitively described in the following way: If we draw shape $\boldsymbol{a}$ on top of shape $\boldsymbol{b}$ and what we see is $\boldsymbol{a}$, then $\boldsymbol{b}$ is a part of $\boldsymbol{a}$, or $\boldsymbol{a}$ is greater than $\boldsymbol{b}$, or $\boldsymbol{b} \leq \boldsymbol{a}$—all of which are equivalent statements. If we see $\boldsymbol{b}$ then it's the other way around. If we see both, then both $\boldsymbol{b} \leq \boldsymbol{a}$ and $\boldsymbol{a} \leq \boldsymbol{b}$, or $\boldsymbol{a} = \boldsymbol{b}$. Part relation $\leq$ partially orders set $U_{ij}$ of shapes and distinguishes the smallest shape, which is the empty shape. The latter is denoted by 0 and depicted by and is a part of every shape. The same relation elevates the set of shapes into an algebra $\boldsymbol{U}_{ij}$, a lattice, and gives a rise to lattice operations of meet and join which dub as shape operations of sum +, and product ·, respectively. Product $\boldsymbol{a} \cdot \boldsymbol{b}$ is the greatest shape which is a part of both $\boldsymbol{a}$ and $\boldsymbol{b}$.

---

D. Krstic (✉)
Hetzel Design Inc., Calabasas, CA, USA
e-mail: gkrstich@aol.com

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_20,

In contrast, sum $\boldsymbol{a} + \boldsymbol{b}$ is the smallest shape that has both $\boldsymbol{a}$ and $\boldsymbol{b}$ as parts. There is another operation of interest in algebras of shapes, which is a shape difference. Intuitively, this operation facilitates erasing of shapes. It is formally defined with the aid of relative complements. Shape lattice $\boldsymbol{U_{ij}}$ doesn't have the top element so there are no complements, but it does have relative complements, which are unique because the lattice is distributive. The latter complements are local, relative to intervals in a lattice. For an interval $[0, \boldsymbol{a} + \boldsymbol{b}]$ relative complement of $\boldsymbol{b}$ is shape $\boldsymbol{d}$ such that $\boldsymbol{b} \cdot \boldsymbol{d} = 0$ and $\boldsymbol{b} + \boldsymbol{d} = \boldsymbol{a} + \boldsymbol{b}$. Shape $\boldsymbol{d}$ is a difference of shapes $\boldsymbol{a}$ and $\boldsymbol{b}$, or $\boldsymbol{d} = \boldsymbol{a} - \boldsymbol{b}$. The difference has all of the parts of $\boldsymbol{a}$, which contain no parts of $\boldsymbol{b}$ as parts and nothing else. There is yet another operation of interest which comes in handy when shapes and their boundaries are considered. That is an operation of symmetric difference $\oplus$ defined as $\boldsymbol{a} \oplus \boldsymbol{b} = (\boldsymbol{a} - \boldsymbol{b}) + (\boldsymbol{b} - \boldsymbol{a}) = (\boldsymbol{a} + \boldsymbol{b}) - (\boldsymbol{a} \cdot \boldsymbol{b})$. A relatively complemented distributive lattice with bottom element but without a top one is a *generalized Boolean algebra* [1] or a GBA for short. Algebras of shapes $\boldsymbol{U_{ij}}$ are GBA-s.

Because designers move, rotate, scale, and otherwise transform shapes Stiny [2] requires for algebras of shapes to be closed under (the actions of) Euclidean transformations. However, he fails short of including the transformations into the structure of algebras. Krstic [3] suggests two alternative ways of including transformations into the structure of algebras of shapes.

The first is to include them into the *signature*—i.e. specification of operations—of the algebra. This way each transformation is treated as an operator—i.e. unary operation—of the algebra rendering it as a GBA with operators. Operator t acts on shape $\boldsymbol{a}$ to produce transformed shape $\mathrm{t}(\boldsymbol{a})$. Formally, an algebra of shapes is a pair $\boldsymbol{U_{ij}} = (U_{ij}, \{+, \cdot, -, \mathrm{t}_1, \mathrm{t}_2, \ldots\})$ with $U_{ij}$ as *carrier*—i.e. the set of elements of the algebra–and an infinite signature. This approach seems also apparent in Stiny's [4] later work.

The Euclidean transformations augment Boolean operations in the algebras of shapes $\boldsymbol{U_{ij}}$ with additional operators. They're defined for basic elements and extend easily for shapes. (Page 212)

The second way is to include transformations into the carrier of the algebra while adding group operations to its signature. This turns an algebra of shapes into a two-sorted one, operating on elements of two different sorts: shapes and transformations. The algebra has a Boolean part and a group part. The former is a GBA with set $U_{ij}$ of shapes and operations +, $\cdot$, and $-$, while the latter is a group with set $T_{ij}$ of transformations and operations of group composition $\circ$, inverse$^{-1}$, and identity $\iota$. The Boolean part handles the structure of shapes while the group part takes care of their symmetries. The connection between the two is facilitated by a binary operation of group action (), which operates on elements of both sorts. It takes shape $\boldsymbol{a}$ and transformation $\boldsymbol{t}$ to produce transformed shape $\boldsymbol{t}(\boldsymbol{a})$. Formally, an algebra of shapes is a pair $\boldsymbol{U_{ij}} = (\{U_{ij}, T_{ij}\}, \{+, \cdot, -, (), \circ, ^{-1}, \iota\})$ with a two-sorted carrier and a seven operation signature.

Three alternative definitions of algebras of shapes have been presented: GBA closed under transformations (external to it), GBA with infinite operators, and two-sorted algebra. These all operate on shapes but each of them do it in a slightly different fashion. In the next two parts we will investigate these differences as they relate to the main application of algebras of shapes: shape grammars.

## Shape Grammars

Stiny and Gips [5] introduced shape grammars four decades ago as production systems capable of generating shapes to define languages of designs. They have evolved since into a formal design theory that views design as a formal enterprise in which the rules are adopted and then followed to compose or describe designs.

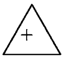
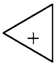
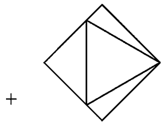
Shape grammars are simply sets of rules, which—starting with an initial rule—apply recursively to create designs, and by extension languages of designs. Rules work as simple algorithms operating in the framework of algebras of shapes to recursively change shapes to which they are applied. A rule is given by two shapes with an arrow between them. The shape on the left side of the arrow is rotated, translated, scaled, or otherwise transformed in order to match a part of the shape to which the rule is applied. The (matched) part is erased and the shape that remains is augmented with the shape on the right side of the arrow transformed the same way the shape on the left side was. The partial ordering relation from the algebra of shapes together with the appropriate transformation informs the matching, while erasing and augmenting shapes is handled by the operations of difference and sum, respectively.

This can be expressed by:

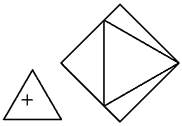
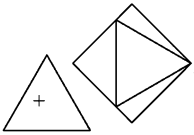
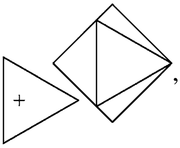
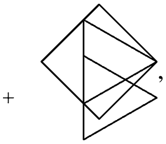
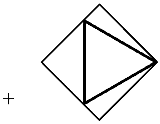
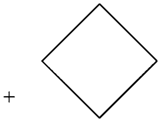
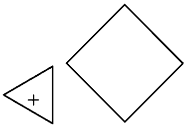
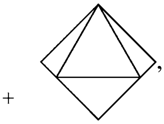
$$\mathrm{t}(\boldsymbol{a}) \leq \boldsymbol{c} \tag{1}$$

$$\boldsymbol{c}' = (\boldsymbol{c} - \mathrm{t}(\boldsymbol{a})) + \mathrm{t}(\boldsymbol{b}) \tag{2}$$

where $\boldsymbol{a} \rightarrow \boldsymbol{b}$ is a rule, $\boldsymbol{c}$ is the shape to which the rule is applied, t is a transformation under which $\boldsymbol{a}$ becomes a part of $\boldsymbol{c}$, and $\boldsymbol{c}'$ is the result of the rule action. Traditionally, a shape grammar will need an initial shape—besides the rules—to start the computation. We opted here for an initial rule instead. The latter is a rule of the form $0 \rightarrow \boldsymbol{c}$ which applies to an empty shape under whatever transformation $\boldsymbol{t}$ we choose and creates the initial shape $\boldsymbol{t}(\boldsymbol{c})$. This has an intuitive appeal, mimicking what designers do when they start drawing, modeling, sculpting on whichever part of their drawing screens, boards, or studios.

Let rule  →  be applied to shape , which may be seen as a square with an inscribed equilateral triangle, or as four triangles—with areas related as $2\sqrt{3}$:2:1:1—or in some other way. The little crosses appearing next to shapes denote the position of the origin of the reference coordinate system.

The rule, in effect, rotates an equilateral triangle around the origin 90° counterclockwise. Both, the shape and rule are defined in $\boldsymbol{U_{12}}$ algebra, which manipulates shapes made of lines in a plane. Let's also assume that $\boldsymbol{U_{12}}$ is defined as a GBA with operators. According to (1), we need to match the equilateral triangle appearing on the left side of the rule with a similar triangle that is a part of the four-triangle shape above—shapes $\boldsymbol{a}$ and a part of $\boldsymbol{c}$, respectively.

Shapes $\boldsymbol{a}$ and $\boldsymbol{c}$ are spatially related as  and sequence , , ,  describes the matching process. Triangle $\boldsymbol{a}$ is transformed at each step, and sequence $s_{1.5}(\boldsymbol{a})$, $r_{30}(s_{1.5}(\boldsymbol{a}))$, $x_5(r_{30}(s_{1.5}(\boldsymbol{a})))$, $y_2(x_2(r_{30}(s_{1.5}(\boldsymbol{a}))))$ describes the changes of $\boldsymbol{a}$. The latter is first enlarged 1.5 times via operator $s_{1.5}$ to match the size of the triangle in $\boldsymbol{c}$, and then rotated 30° via $r_{30}$ to match its orientation. A five-unit translation to the right via operator $x_5$ aligns the two triangles vertically, which is then followed by a two-unit rise via $y_2$ to finally align them—as shown in thick line above. The (matched) equilateral triangle in $\boldsymbol{c}$ is erased via the difference operation, in accordance with the in-parenthesis part of (2), resulting in shape $\boldsymbol{c} - y_2(x_5(r_{30}(s_{1.5}(\boldsymbol{a}))))$, which is square  shown again  together with triangle $\boldsymbol{b}$, which is the right side of the rule. The latter is transformed, using the same sequence of transformations that was used on $\boldsymbol{a}$, and added to the square above resulting in shape $\boldsymbol{c}'$ , in accordance with (2),

The initial four-triangle shape appears rotated by 90° although only one of its triangles has been rotated. "Miracles" like this one are standard when computing with shapes. Many others may be found in works by Knight [6], Stiny [4], March [7], and even Wittgenstein [8], to mention a few. Formulas (1) and (2) now look more elaborate if not complicated:

$$y_2(x_5(r_{30}(s_{1.5}(a)))) \leq c \tag{3}$$

$$\boldsymbol{c}' = (\boldsymbol{c} - y_2(x_5(r_{30}(s_{1.5}(\boldsymbol{a}))))) + y_2(x_5(r_{30}(s_{1.5}(\boldsymbol{b})))) \tag{4}$$

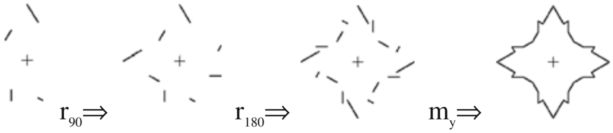
Because transformations form a group, it is possible to construct group composition of the operators above, $t = y_2 \circ x_5 \circ r_{30} \circ s_{1.5}$. This way the formulas (3) and (4) may be turned back to (1) and (2). Group composition—together with other combinations of transformations—is beyond $\boldsymbol{U_{ij}}$ algebras defined as GBA-s with operators, but native in their two-sorted counterparts. The above matching process, in a two-sorted algebra, is a matter of combining transformations in a group part of the algebra and then applying the result via the group action operation.

It seems advantageous to be able to compute with transformations—and to have this capability conveniently built into the algebra. For example, doing the inverse of a transformation allows undoing some of the erroneous moves. A limited set of transformations is combined in CAD systems to produce arbitrary transformations. Typically, rotations around coordinate axes are combined with translations to achieve rotations around arbitrary axes. Semidirect products of groups play a role there. Transformations deal with symmetries of shapes, which have many design implications [9].

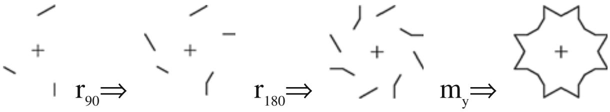
The shapes will be generated here using transformations (only) in a simple schema x → x + t(x). This and other such schemas are championed by Stiny [4, 10], to define rules (of shape grammars), and then use them to compute with shapes according to formulas (1) and (2). We will use the schema in a different way to generate shapes in accordance with formula

$$c' = c + t(c) \tag{5}$$

That is, make new by adding a transformed copy of whatever you see to its original version. Unlike the shapes defining traditional shape grammar rules the shapes here are not given ahead of a computation. Instead they change at each step of it as what we see (shape $\boldsymbol{c}$) changes. In contrast, transformations are given before the computation starts so that they—and not the shapes—determine the rules. These are a different kind of rules, *shapeless rules* where a transformation is, in fact, a rule.

A sequence 

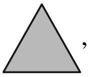
is such a computation. It starts with the most left shape and progresses in accordance with (5) using a 90° rotation ($r_{90}$), followed by a 180° rotation ($r_{180}$), followed by a reflection (mirror) about the y-axes ($m_y$). The resulting star shape has the symmetry of a square—D4 dihedral group. Indeed, a 90° rotation and a reflection about the y-axis are sufficient to recreate all of the 8 elements of D4 group. Because the rules are transformations, the same sequence of rules may generate a different shape—if the initial shapes are different. Although the shapes are different, their symmetry groups always contain the group generated by the transformations that are the elements of the rule sequence.

For example, 

is a computation with the same sequence of rules as the one above and the resulting star has the same symmetry as the one above.

Shapeless rules may be recast to work with (1) and (2) and appear like more traditional shape grammars. The rules of form $c \rightarrow c + t(c)$ will do the job. These $\mathrm{x} \rightarrow \mathrm{x} + \mathrm{t(x)}$ variable x becomes shape $\boldsymbol{c}$ and variable t becomes shapeless rule $\boldsymbol{t}$. Such a rule applies to shape $\boldsymbol{c}$ satisfying (1) under the identity transformation. It first—in accordance with the in-parenthesis part of (2)—erases shape $\boldsymbol{c}$ only to change its mind and add $\boldsymbol{c}$ back together with its transformed copy $\boldsymbol{t}(\boldsymbol{c})$—thus completing (2). This has the same effect as (5) resulting in shape $\boldsymbol{c} + \boldsymbol{t}(\boldsymbol{c})$. We could also be parsimonious. Who needs the identity transformation, erasing, and mind changing? One shape—the one to be changed—and one transformation—a shapeless rule—is all we need. Another Stiny's rule schema $0 \rightarrow \mathrm{x}$ comes in handy. By assigning $\boldsymbol{c}$ to x we get rule $0 \rightarrow \boldsymbol{c}$. Because the left side of this rule is (an) empty (shape), it may apply to any shape, and, because a transformation of an empty shape is an empty shape, this may be done under any transformation $\boldsymbol{t}$. The latter may be chosen to be a shapeless rule. Then, (1) and (2) are both satisfied: (1) because $\boldsymbol{t}(0) \leq \boldsymbol{c}$ always holds, and (2) because (5) may be written as $\boldsymbol{c}' = (\boldsymbol{c} - \boldsymbol{t}(0)) + \boldsymbol{t}(\boldsymbol{c})$. A shapeless rule grammar is a traditional shape grammar, after all, but with a triple twist: the rules are redefined at each step of a computation, the rule schema remains the same for all rules, and transformations are fixed ahead of the computation. Needles to say that with traditional shape grammars rules are fixed ahead of a computation, they may be defined with a different schema for every rule, and transformations change as we compute.

## Combinations of Algebras: Compound Shapes

Shapes made of basic elements of one kind only—ike the shapes in the previous section, which were made of lines only—are rare in practice. Usually they contain different kinds of basic elements like shape , which has a planar region and three lines. Such shapes do not belong to a single $\boldsymbol{U_{ij}}$ algebra, but to a combination of two or more of such algebras. Typically algebras combine in direct products. The latter are algebras consisting of two or more component algebras, which all have the same signatures. Direct products operate on Cartesian products of their component's carriers using componentwise defined operations. The shape above may be seen as an ordered pair $(\boldsymbol{a}, \boldsymbol{b})$, where $\boldsymbol{a}$ is a shape made of lines and $\boldsymbol{b}$ is a shape made of planar regions. It is defined in the direct product of $\boldsymbol{U_{12}}$ and $\boldsymbol{U_{22}}$ which is a GBA denoted by $\boldsymbol{U_{12}} \times \boldsymbol{U_{22}}$ with carrier $U_{12} \times U_{22} = \{(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{x} \in U_{12}, \boldsymbol{y} \in U_{22}\}$ and operations given by $(\boldsymbol{a}, \boldsymbol{b}) * (\boldsymbol{c}, \boldsymbol{d}) = (\boldsymbol{a} *_{12} \boldsymbol{c}, \boldsymbol{b} *_{22} \boldsymbol{d})$, where $*_{\mathrm{ij}}$

stands for +, −, or · from $\boldsymbol{U_{ij}}$. The behavior of a direct product of algebras of shapes depends on how the latter are defined.

For example, if $\boldsymbol{U_{12}}$ and $\boldsymbol{U_{22}}$ are defined as GBA-s with operators, then operator t from $\boldsymbol{U_{12}} \times \boldsymbol{U_{22}}$ acts on a compound shape ($\boldsymbol{a}, \boldsymbol{b}$) so that $\mathrm{t}(\boldsymbol{a}, \boldsymbol{b}) = (\mathrm{t}_{12}(\boldsymbol{a}), \mathrm{t}_{22}(\boldsymbol{b}))$. Note, that GBA-s with operators may be combined in a direct product only if there is a 1–1 and onto correspondence (bijection) between their operators. Corresponding operators transform their respective shapes in the same way. If $\mathrm{t}_{12}$ is rotating lines +30° around the origin, then $\mathrm{t}_{22}$ is doing the same with planar regions. Consequently, only algebras defined in the same space—or the spaces of the same dimension—may be combined in direct products.

In contrast, there are no restrictions on how algebras of shapes, if defined as two-sorted, may be combined in direct products. We may have $\boldsymbol{U_{12}} \times \boldsymbol{U_{33}}$ and a compound transformation ($\boldsymbol{t_1}, \boldsymbol{t_2}$) from $T_{12} \times T_{33}$ acting on a compound shape ($\boldsymbol{a}$, $\boldsymbol{b}$) such that $(\boldsymbol{t_1}, \boldsymbol{t_2})\ ((\boldsymbol{a}, \boldsymbol{b})) = (\boldsymbol{t_1}(\boldsymbol{a}), \boldsymbol{t_2}(\boldsymbol{b}))$ with $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ being arbitrarily transformations from their respective algebras.

If $\boldsymbol{U_{ij}}$ is defined as GBA closed under transformations, then there are two ways of looking at their direct products. First, one may take a direct product of GBA-s and require for it to be closed under transformations. The new algebra behaves like a direct product of GBA-s with operators—where all components have to be in the same space and altered by the same transformations. The second way requires for a direct product of GBA-s to be closed under the direct product of corresponding groups of transformations. Such an algebra behaves like a two-sorted algebra of shapes allowing for component algebras to engage in direct products without any restrictions.

Compound shapes may be used to define rules of shape grammars in a straightforward way. In our example a compound rule is of the form $(\boldsymbol{a_1}, \boldsymbol{a_2}) \rightarrow (\boldsymbol{b_1}, \boldsymbol{b_2})$ and expressions (1) and (2) may be given componentwise and they depend on which definition of algebra is used. For GBA with operators the expressions are

$$(t_{12}(a_1),\ t_{22}(a_2)) \leq (c_1, c_2). \tag{6}$$

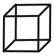
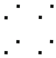
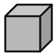
$$\left(c_1', c_2'\right) = ((c_1 -_{12} t_{12}(a_1)) +_{12} t_{12}(b_1),\ (c_2 -_{22} t_{22}(a_2)) +_{22} t_{22}(b_2)) \tag{7}$$

and for two-sorted algebra they are

$$(t_1(a_1), t_2(a_2)) \leq (c_1, c_2). \tag{8}$$

$$\left(c_1', c_2'\right) = ((c_1 -_{12} t_1(a_1)) +_{12} t_1(b_1),\ (c_2 -_{22} t_2(a_2)) +_{22} t_2(b_2)) \tag{9}$$

Compound shapes are treated differently in different algebras of shapes. In GBA with operators they are homogeneous in a sense that all of their components appear in the same space and are transformed without changing their spatial relations or being pulled apart.

Take a cube  defined in $U_{33}$ with its boundary consisting of 6 planar squares  in $U_{23}$, and its boundary consisting of 12 lines  in $U_{13}$, and its boundary consisting of 8 points  in $U_{03}$ and form a compound shape  defined in a direct product $U_{33} \times U_{23} \times U_{13} \times U_{03}$. There is no way of changing this hierarchy of boundaries among component shapes by using any of the operators from the algebra. We may completely transform the cube, even use affine transformations to turn it, say, into a rhomboidal prism, but it will still have all of the boundaries accounted for.

In contrast, two-sorted algebras provide plenty of possibilities for destroying this hierarchy of boundaries. A different transformation may be used on each component of the compound shape to disturb the boundary alignment. Compound transformation $(x_1, x_2, x_3, x_4)$, where $x_k$ is a $k$ unit translation in the direction of x-axes, will do the trick. Each boundary will be misaligned by one unit thus failing to be a boundary.

Although, the added freedom in using transformations did not serve the above example well, it does allow for some neat applications. Heterogeneous compound shapes of two-sorted algebras, which may have each of their components defined in a different space and moved, mirrored, rotated, and scaled independently allow for simultaneous generation—in the framework of shape grammars—of different representations of the same object. Duarte [11] defines rules, which simultaneously generate six different representations of Malaqueira house designs in a stile of Alvaro Siza. These include the envelope and spaces axonometric views, three floor plans, and an elevation. Krstic [12] defines rules to simultaneously generate medieval Rascian church plans together with the related justified permeability graphs. Such rules could, in some cases, be defined in direct products of GBA-s with operators, but would be difficult if not impossible to implement. That's because even if the component shapes are defined in the same space they may require a different transformation each in order to satisfy inequality (6). A direct product of GBA-s with operators cannot handle this. Two-sorted algebras are up to this task, but they have a problem when the same transformation is needed across the components of compound shapes—like in the cube example above. Even though one can use only n-tuples with all of the components being the same transformation, this is not a very elegant solution and some new construct may be needed.

Often we compute not only to get the result (design), but also to gain understanding. Although in the case of direct products one size fits all, wearing something that exactly fits better shows our shape. In the next section two different sums of algebras of shapes will be defined as constructs smaller than direct products and better suited for certain tasks.

## Sums of Algebras

Sums of algebras of shapes have been appearing in shape grammar theory for many years without a formal definition. More recently Knight [13, 14] introduces them informally and gives an account of their properties. We will follow her lead as well as Krstic [3] and some considerations from the previous section to provide a formal underpinning for such sums. If not stated differently we will, throughout this section, consider algebras of shapes to be two-sorted ones.

Unlike direct products of algebras, which stem from the universal algebra there is no readily available way of constructing sums of algebras of shapes. Sums exist for Boolean algebras [15], but they are duals of direct products and do not work for our purpose.

Knight requires for component algebras to be defined in the same space in order to be summed [13]

Different algebras can be summed to form a composite algebra only when the shapes … in them are defined in the same 0D, 1D, 2D, or 3D space. (Page 170)

We will also require the use of the same transformations on all of the components in order to prevent the compound shape from being pulled apart [3].

Let $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$ be two algebras of shapes defined in the same space, or in spaces of the same dimension *j*. A subset of the direct product $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$ with:

(a) the Boolean part coinciding with that of $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$, and
(b) the group part enumerating ordered pairs of the form $(\boldsymbol{t_1}, \boldsymbol{t_2})$, where

$$t_1 = t_2, t_1 \in U_{ij}, \text{ and } t_2 \in U_{kj}$$

is the *sum* of the two algebras, denoted by $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$.

Note that $\boldsymbol{t_1}$ and $\boldsymbol{t_2}$ may act on shapes of different dimensions—because $\boldsymbol{t_1}$ is defined in $\boldsymbol{U_{ij}}$ and $\boldsymbol{t_2}$ in $\boldsymbol{U_{kj}}$, and *i* and *k* may not be equal. The equal sign above is related only to the actions of the transformations. If, for example, $\boldsymbol{t_1}$ is acting on shapes made of points and $\boldsymbol{t_2}$ on shapes made of lines, then the transformations are considered equal if they transform the respective shapes in the same way—say rotate around x-axes +30°.

The sum $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ is a subdirect product of $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$ [3].

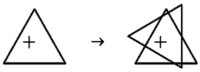
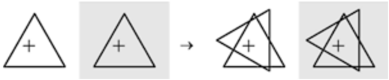
In accordance with (a) above Boolean part of $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ is a subalgebra of its counterpart in $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$. It follows from (b) above that the group part of $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ enumerates all of the transformations from both $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$ and forms a group isomorphic to the group parts of both algebras. Carrier $U_{ij} \times U_{kj}$ of the Boolean part of $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ is closed under the group actions—of transformations from its group part—because it contains all of the ordered pairs of shapes from $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$. Consequently, $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ is a subalgebra of $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$ and because it contains all of the elements of both $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$, it is a subdirect product of the two algebras.

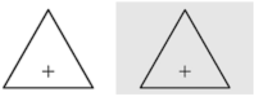
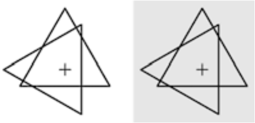
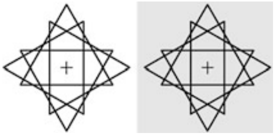
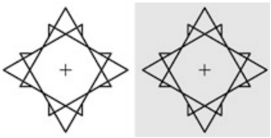
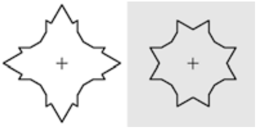
Note that the sum of two-sorted algebras of shapes is equivalent to the direct product of their GBA with operators counterparts.

The sum comes in handy for modeling a variety of situations appearing in modern CAD systems.

For example, a shape made of points, lines and planar regions defined in a plane may be seen as an element of an algebra, which is the sum of algebras for points, lines, and planes.

Another example is a CAD drawing appearing on different layers. It may be seen as an element of a sum of algebras for each layer. For example a plan may have different layers for walls, windows, doors …

A sum of algebras will be used to simultaneously generate the two star shapes generated earlier with shapeless rules. Because both shapes are defined in $\boldsymbol{U_{12}}$ the computations may be seen as taking place in two different spaces of the same dimension so that sum $\boldsymbol{U_{12}} + \boldsymbol{U_{12}}$, or $2\boldsymbol{U_{12}}$, provides the framework. The spatial relation behind the earlier rule—that rotates an equilateral triangle according to the schema x → t(x)—is reused with a new schema x → x + t(x) to create a new rule  that works for both star shapes. Thus, the same rule is given for both spaces , where the gray background distinguishes the spaces.

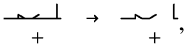
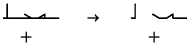
We start by applying the latter rule to compound shape  resulting in compound shape . After two more applications of the same rule we get compound shape , which contains both star shapes. Now it is a matter of erasing the excess lines to expose the stars. The compound erasing rule , when applied four times, does half of the job  while rule , applied eight times, finishes the work .

There are a couple of notes to be made starting with a classical one [16]: note that nice numeric relationships are buried in the stars. Because of an equilateral triangle and 90° rotations one may expect some $\sqrt{3}$-s and $\sqrt{2}$-s, respectively. Indeed, the 8-pointed star is inscribed in an octagon with side lengths alternating between $\sqrt{2}$ and $\sqrt{3}$. The 12-pointed star should be inscribed in a dodecagon, but the latter degenerated into a square with each of its sides divided by the star into three parts related as $1{:}\sqrt{3}{:}1$.

On a non-classical note [16] the last compound rule is the only one with different component rules. Even so the same transformations worked for both of the components facilitating the matching and replacements of the lines. This happy coincidence (or a great plan by the author) allowed for the computations to take place in $2\boldsymbol{U_{12}}$ defined as a two-sorted algebra and also to remain unchanged if conducted in the direct product $\boldsymbol{U_{12}} \times \boldsymbol{U_{12}}$, or $\boldsymbol{U_{12}^2}$ defined as a GBA with operators. However, a slight alteration of the rule may make a huge difference. We shall try one but first we need some tools like the *equivalence of rules*.

Two rules are equivalent if whenever applied to the identical shapes the resulting shapes are identical.

Trivially, the identical rules are equivalent. Less trivially, rules $\boldsymbol{a} \rightarrow \boldsymbol{b}$ and $\boldsymbol{g}(\boldsymbol{a}) \rightarrow \boldsymbol{g}(\boldsymbol{b})$, where $\boldsymbol{g}$ is a transformation, are equivalent. Both rules do the same thing only they do it under different transformations. If the former rule satisfies (1) by matching a part of some shape $\boldsymbol{c}$ under some transformation $\boldsymbol{t}$ then the latter achieves the same under a different transformation $t \circ g^{-1}$. In both cases the shapes resulting after the rules have been applied are identical. A simple exercise in shape arithmetic shows it. In the first case the resulting shape according to (2) is $(\boldsymbol{c} - \boldsymbol{t}(\boldsymbol{a})) + \boldsymbol{t}(\boldsymbol{b})$ and in the second $(c - t \circ g^{-1}(g(a))) + t \circ g^{-1}(g(b)) = \left(c - t \circ g_{\circ}^{-1} g(a)\right) + t \circ g_{\circ}^{-1} g(b) = (c - t(a)) + t(b)$.

Rule , which is the first component of our second erasing rule, may now be replaced with  creating an altered compound erasing rule . The two versions of the first component are equivalent because the latter one is obtained by reflecting both sides of the former rule with respect to y-axis. In contrast, the original and the altered erasing rules are not equivalent although their components are. The altered rule needs different transformations for the two components to match the corresponding parts of the stars. Consequently, the rule neither works in $2\boldsymbol{U_{12}}$, defined as two-sorted, nor in $\boldsymbol{U_{12}^2}$, defined as a GBA with operators.

To salvage the star grammar we may break up the new rule into two rules  and  such that each rule uses one of the components of the original rule and an identity $0 \rightarrow 0$ in place the other component. The two rules apply in sequence first altering one of the star shapes and then the other. The original parallel computation becomes a less efficient sequential one. With GBA-s with operators this is the only way to go. With two-sorted algebras we have a choice: use this option with sum $2\boldsymbol{U_{12}}$, or go to direct product $\boldsymbol{U}_{\boldsymbol{12}}^{2}$ and use the altered erasing rule and parallel computation. In a direct product of two-sorted algebras of shapes compound rules are equivalent if their components are equivalent.

The sum as defined above is not commutative, that is, $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ differs from $\boldsymbol{U_{kj}} + \boldsymbol{U_{ij}}$ whenever $i$ differs from $k$. In contrast, a shape made of points, lines and planar regions defined in a plane, when seen in the context of a traditional pencil on paper drawing, may call for a commutative sum. It is basically the same drawing regardless of the order in which we drew the points, lines, and shaded regions.

Defining a commutative sum is not as straightforward as defining the non-commutative sum above.

We will start by denoting such a sum with $+_{\mathrm{c}}$ and the carrier of the Boolean part of $\boldsymbol{U_{ij}} +_{\mathrm{c}} \boldsymbol{U_{kj}}$ by $U_{ij} + U_{kj}$. We will also require for algebras that are summed not to be of the same kind—or $i \neq k$—to keep things simple.

The carrier is given by $U_{ij} + U_{kj} = \{\{\boldsymbol{x}, \boldsymbol{y}\} \mid \boldsymbol{x} \in U_{ij}, \boldsymbol{y} \in U_{kj}\}$. It consists of two-element sets or unordered pairs in place of the ordered pairs of $U_{ij} \times U_{kj}$. There is a simple relation between the two carriers, which stems from the standard (Kuratowski's) definition of an ordered pair, $(a, b) = \{\{a\}, \{a, b\}\}$. Because the latter can be turned into an unordered pair via the operation of set union, $\cup\{\{a\}, \{a, b\}\} = \{a, b\}$, one may define a mapping *uni*: $U_{ij} \times U_{kj} \rightarrow U_{ij} + U_{kj}$ given by $uni(\boldsymbol{x}) = \cup\, \boldsymbol{x}$ to map elements of one carrier to the elements of another. Now carrier $U_{ij} + U_{kj}$ may be redefined $U_{ij} + U_{kj} = \{\cup\, \boldsymbol{x} \mid \boldsymbol{x} \in U_{ij} \times U_{kj}\}$.

Because $\{\boldsymbol{x}, \boldsymbol{y}\} = \{\boldsymbol{y}, \boldsymbol{x}\}$, sum $+_{\mathrm{c}}$ is commutative in contrast to the ordered pairs of sum +, 'where $(\boldsymbol{x}, \boldsymbol{y}) \neq (\boldsymbol{y}, \boldsymbol{x})$ whenever $\boldsymbol{x} \neq \boldsymbol{y}$.

Boolean operations cannot be defined in a componentwise fashion as there are no components, but elements instead. However, there is a possibility to define the operations by combining the elements of their argument sets exhaustively.

Lets take shapes $\boldsymbol{x}$ and $\boldsymbol{u}$ from $U_{ij}$ and $\boldsymbol{y}$ and $\boldsymbol{v}$ from $U_{kj}$ and create sets $\{\boldsymbol{x}, \boldsymbol{y}\}$ and $\{\boldsymbol{u}, \boldsymbol{v}\}$, which belong to $U_{ij} + U_{kj}$—in accordance with the definition above. Their Boolean sum—when defined exhaustively—is $\{\boldsymbol{x}, \boldsymbol{y}\} + \{\boldsymbol{u}, \boldsymbol{v}\} = \{\boldsymbol{x} + \boldsymbol{u}, \boldsymbol{x} + \boldsymbol{v}, \boldsymbol{y} + \boldsymbol{u}, \boldsymbol{y} + \boldsymbol{v}\}$. However, not all elements of the sum are defined. In particular, $\boldsymbol{x} + \boldsymbol{v}$ is not defined because its arguments belong to two different algebras. The same is true of $\boldsymbol{y} + \boldsymbol{u}$, so that only two elements remain $\{\boldsymbol{x}, \boldsymbol{y}\} + \{\boldsymbol{u}, \boldsymbol{v}\} = \{\boldsymbol{x} +_{\mathrm{ij}} \boldsymbol{u}, \boldsymbol{y} +_{\mathrm{kj}} \boldsymbol{v}\}$.

Boolean sum is an unordered pair with one element in $U_{ij}$ and the other in $U_{kj}$, which makes the sum an element of $U_{ij} + U_{kj}$. Consequently, $U_{ij} + U_{kj}$ is closed under the operation of Boolean sum. Operations of shape product and difference could be defined in the same way, which completes the definition of the Boolean part of the commutative sum of algebras.

The group part, on the other hand, may be defined with carrier $T_j$ as a set of unordered pairs of transformations with one element in $T_{ij}$ and the other in $T_{kj}$. The group operations are also defined exhaustively. For example, group composition $\{\boldsymbol{t_1}, \boldsymbol{g_1}\} \circ \{\boldsymbol{g_2}, \boldsymbol{t_2}\} = \{\boldsymbol{t_1} \circ \boldsymbol{g_2}, \boldsymbol{t_1} \circ \boldsymbol{t_2}, \boldsymbol{g_1} \circ \boldsymbol{g_2}, \boldsymbol{g_1} \circ \boldsymbol{t_2}\} = \{\boldsymbol{t_1} \circ_{\mathrm{ij}} \boldsymbol{t_2}, \boldsymbol{g_1} \circ_{\mathrm{kj}} \boldsymbol{g_2}\}$, where $\boldsymbol{t_1}, \boldsymbol{t_2} \in T_{ij}$ and $\boldsymbol{g_1}, \boldsymbol{g_2} \in T_{kj}$. Set $T_i$ is closed under group operations, which completes the definition of the group part of the commutative sum of algebras.

Group action when defined exhaustively is $\{\boldsymbol{t}, \boldsymbol{g}\}(\{\boldsymbol{x}, \boldsymbol{y}\}) = \{\boldsymbol{t}(\boldsymbol{x}), \boldsymbol{t}(\boldsymbol{y}), \boldsymbol{g}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{y})\} = \{\boldsymbol{t}(\boldsymbol{x}), \boldsymbol{g}(\boldsymbol{y})\}$, where $\boldsymbol{t} \in T_{ij}$ and $\boldsymbol{g} \in T_{kj}$. Carrier $U_{ij} + U_{kj}$ is closed under this operation.

Note that the operations carried on in a commutative sum of algebras reassemble the operations carried on in their non-commutative counterparts. The only difference is that the ordered pairs of the latter are replaced with the unordered pairs of the former. This means that *uni* mapping is a homomorphism between the two algebras. For example, *uni* preserves Boolean sums so that $uni(\boldsymbol{x}, \boldsymbol{y}) + uni(\boldsymbol{u}, \boldsymbol{v}) = uni(\boldsymbol{x} + \boldsymbol{u}, \boldsymbol{y} + \boldsymbol{v})$ holds in $\boldsymbol{U_{ij}} +_c \boldsymbol{U_{kj}}$. Because $\boldsymbol{U_{ij}} +_c \boldsymbol{U_{kj}}$ is commutative, the same mapping *uni*, but with a different domain $U_{kj} \times U_{ij}$ is a homomorphism between $\boldsymbol{U_{kj}} + \boldsymbol{U_{ij}}$ and $\boldsymbol{U_{ij}} +_c \boldsymbol{U_{kj}}$.

The commutative diagram below shows the relation between the three algebras.

$$\begin{tikzcd} \boldsymbol{U_{ij}} + \boldsymbol{U_{kj}} \arrow[rr, shift left, "inv"] \arrow[dr, "uni"'] & & \boldsymbol{U_{kj}} + \boldsymbol{U_{ij}} \arrow[ll, shift left, "inv"] \arrow[dl, "uni"] \\ & \boldsymbol{U_{ij}} +_c \boldsymbol{U_{kj}} & \end{tikzcd}$$

Note that mapping *inv* defined as $inv(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{b}, \boldsymbol{a})$ is an isomorphism between $\boldsymbol{Uij} + \boldsymbol{Ukj}$ and $\boldsymbol{Ukj} + \boldsymbol{Uij}$.

Although commutative sums have been defined in the framework of two—sorted algebras of shapes, they can easily be adopted for GBA-s with operators—just replace $\boldsymbol{U_{ij}} + \boldsymbol{U_{kj}}$ and $\boldsymbol{U_{kj}} + \boldsymbol{U_{ij}}$ in the diagram above with $\boldsymbol{U_{ij}} \times \boldsymbol{U_{kj}}$ and $\boldsymbol{U_{kj}} \times \boldsymbol{U_{ij}}$, respectively. In contrast, non-commutative sums do not apply to GBA-s with operators although nothing is lost as one may use direct products instead.

The commutative sum as given above depends on the condition that algebras that are summed are not of the same kind—that is $i \neq k$. What will happen if we try algebras of the same kind? Knight predicts [14].

When the algebras that are summed are of the same kind, then the entities (say, lines) in the parallel computations can interact with one another. If algebras are not of the same kind, the entities simply exist in the same space in separate, transparent layers. (Page 11)

It is clear for non-commutative sums that no interaction between component algebras is possible even if these are of the same kind. Non-commutative sums are subalgebras of direct products, which guarantees that shapes are kept separate, ordered in n-tuples. In contrast, commutative sums have sets in place of n-tuples and no such separation.

To better understand commutative sums we will recast their definition in context of algebras of complexes or complex algebras. Given an algebra $\boldsymbol{A}$, a complex algebra $\wp(\boldsymbol{A})$ is defined by extending the operations of $\boldsymbol{A}$ to its nonempty subsets in an exhaustive fashion. To construct $\boldsymbol{U_{ij}} +_c \boldsymbol{U_{kj}}$ we start by piling up the two algebras in a set union. The result is a partial algebra with elements either belonging to $\boldsymbol{U_{ij}}$ or to $\boldsymbol{U_{kj}}$ and operations that are defined only when their arguments belong to the same algebra. The partial algebra is extended into a complex algebra. This is again a partial algebra because Boolean operations with elements of one argument belonging to $\boldsymbol{U_{ij}}$ and elements of the other belonging to $\boldsymbol{U_{kj}}$ are not defined. In contrast, subset $U_{ij} + U_{kj}$ of the complex algebra is an algebra—it is the commutative sum of $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$. Note that this only works when $i \neq k$. When $i = k$, the union of $\boldsymbol{U_{ij}}$ and $\boldsymbol{U_{kj}}$ is $\boldsymbol{U_{ij}}$ and the subset $U_{ij} + U_{kj}$ of $\wp(\boldsymbol{U_{ij}})$ has all of its singletons and unordered pairs. The latter have elements that are shapes belonging to the same algebra and may interact with each other in computations in accordance with Knight's statement.

Unfortunately, $U_{ij} + U_{kj}$ is only a partial algebra. Because all shapes now belong to the same algebra, Boolean sum $\{\boldsymbol{x}, \boldsymbol{y}\} + \{\boldsymbol{u}, \boldsymbol{v}\} = \{\boldsymbol{x} + \boldsymbol{u}, \boldsymbol{x} + \boldsymbol{v}, \boldsymbol{y} + \boldsymbol{u}, \boldsymbol{y} + \boldsymbol{v}\}$ has a 4-element result, which does not belong to $U_{ij} + U_{kj}$. It is interesting to note that $U_{ij} + U_{kj}$, when $i = k$, is a set of all singleton and 2-element decompositions of shapes from $\boldsymbol{U_{ij}}$. That is because any finite set of shapes belonging to the same algebra is a decomposition of their Boolean sum. This places parallel computations, done in a single algebra, into a domain of shape decompositions [17], which may prove an interesting topic for further research.

## Conclusion

We have examined three different formats of algebras of shapes, which have been used as a framework for shape grammars. Although all of the algebras are capable of computing with shapes in a meaningful way, they all do it differently. The difference does not stem from how Boolean operations on shapes are done, as all three algebras are generalized Boolean algebras, but from the way they treat transformations of shapes.

GLB-s closed under transformations treat the latter as external to the algebra, only requiring that a transformation acting on a shape produces another shape.

GLB-s with operators incorporate transformations into the algebraic structure in a form of additional operations on shapes. This is a neat approach because shapes are the only entities to manipulate. However, the structure of transformations, which allows for their combinations, is ignored.

Finally, two-sorted algebras capable of combining the shapes as well as transformations leave little to be desired, though at the cost of more complicated algebraic structure.

All three algebras work well on their own in supporting rule actions. They do it with more or less the same efficiency. GBA-s closed under transformations and two-sorted algebras have a small advantage there, as they allow for combinations of transformations—whether external to the algebra or within it—to streamline the matching phase of a rule action.

The main differences between the algebras surface when their combinations are attempted.

Direct products of GBA-s with operators are severely limited as they can only have component algebras defined in the same space and closed under the same transformations. This is sufficient when we have shapes of different dimensions considered as parts of a whole, which exists and is manipulated in the same space. Computing with different representations of the same object is only marginally possible with GBA-s with operators. For example, Duarte's rules combining the plans, elevations, and axonometric views of a house cannot be defined in direct products of GBA-s with operators.

In contrast, direct products of two-sorted algebras of shapes allow for any combination of component algebras. Moreover, certain subalgebras of direct products of two-sorted algebras behave as direct products of GBA-s with operators and are well suited for manipulating homogeneous compound shapes. Such subalgebras are non-commutative sums of two-sorted algebras of shapes. We also developed the commutative sum of algebras of shapes in order to model what designers do when they draw lines and shaded areas with classical tools like pencil or ink.

There is another interesting feature of two-sorted algebras of shapes: they are of the same structure as shapes themselves—or the other way around. The set of all parts of a shape is a Boolean algebra closed under its symmetry group of transformations. When defined as two-sorted algebra it becomes a subalgebra of the related two-sorted algebra of shapes [17]. Any part of a shape being a shape itself is also a subalgebra of the algebra of shapes.

Finally, the main contributions of this paper are in recognizing that differently defined algebras of shapes differ in how they compute, are combined, or handle shape transformations; identifying the two different sums of such algebras and providing their formal definitions; pinpointing the difference between the sums and direct products in how the former restrict and the latter allow combinations of transformations; defining the shapeless rules and equivalence of rules—both not to be overrated; giving a 3-page up-to-date account of shape, algebras of shapes, and shape grammars; and last, but not the least, providing for some nice shapes.

## References

1. Birkhoff G (1993) Lattice theory, 3rd edn. American Mathematical Society, Providence
2. Stiny G (1991) The algebras of design. Res Eng Design 2:171–181
3. Krstic D (1999) Constructing algebras of design. Environ Plan 26:45–57
4. Stiny G (2006) Shape: talking about seeing and doing. MIT Press, Cambridge
5. Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. In: Frieman CV (ed) Information Processing '71. North-Holland, Amsterdam, pp 1460–1465
6. Knight T (1994) Transformations in design: a formal approach to stylistic change and innovation in the visual arts. Cambridge University Press, Cambridge
7. March L (1996) Babbage's miraculous computation revisited. Environ Plan 23:369–376
8. Wittgenstein L (1956) Remarks on the foundations of mathematics. Basil Blackwell, Oxford
9. Economou A (1999) The symmetry lessons from Froebel's building gifts. Environ Plan 26:75–90
10. Stiny G (2010) What rule(s) should I use? Nexus Netw J 13:15–47

11. Duarte JP (2005) Towards mass customization of housing: the grammar of Siza's houses at Malagueira. Environ Plan 32:347–380
12. Krstic D (1990) Language of the Rascian school: Analyzing Rascian church plans via parallel shape grammars. Design and computation research seminar GSAUP UCLA (unpublished)
13. Knight T (2003) Computing with ambiguity. Environ Plan 30:165–180
14. Knight T (2004) Interaction in Visual Design Computing. MIT Press, Cambridge
15. Givant S, Halmos P (2009) Introduction to Boolean Algebras. Springer, Berlin
16. Knight T, Stiny G (2001) Classical and non-classical computation. Architectural Res Q 5:355–372
17. Krstic D (2005) Shape decompositions and their algebras. AIEDAM 19:261–276

# Representing 3D Shape Grammars in a Generative Product Design System

**Jia Cui and Ming-Xi Tang**

**Abstract** A new representation of 3D shape, referred to as Dynamic Shape Representation, is introduced in this paper. This representation is aimed at better supporting the generative application of shape grammars in product design. Inspired from real design process at a cognitive level, shape transforming actions in product design instead of geometric and topologic features of the product form are represented and manipulated in our system. This Dynamic Shape Representation offers better support to generative design of conceptual product forms with flexible shape creations. Two product design examples are presented in this paper in order to evaluate the feasibility of this new representation for 3D shape grammar applications in design.

## Introduction

Shape Grammars (SG) have been developed for more than three decades since its first introduction by Stiny and Gips [1] in 1972. Its outstanding visual description, recursive usage and shape vague tolerance (emergence) play an important role in design formalization, stylization and simulation. Shape Grammars can build a strong connection between a design domain, which is traditionally recognized as a human dominated area, and computer technology, which is the modern logic science, for the development of advanced Computer Aided Design (CAD). Designers are more sensitive about the shapes deriving from semantics, feelings and emotions, than geometric and topologic representations of shape which are the fundamental elements used in shape grammars. There is a need to develop a better

---

J. Cui (✉) · M-X. Tang
The Hong Kong Polytechnic University, Kowloon , Hong Kong
e-mail: cuijia1247@163.com

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_21,

integration between shape grammar and conceptual design in order to utilize the mature symbol computation developed in computer sciences.

A generative product design system can scatter one existing conception into a huge solution space for more alternatives. Working with a shape grammar representation, a generative product design system can enhance the usability of the analyses collected by shape grammars to generate novel design candidates. As early as 1981, Knight [2] proposed two basic ways to use the generative product design method, i.e., changing the partial rules and changing the conditions on the shape parameters. After that, he introduced redundant rules [3], and Liew illustrated equivalence rules [4], and Trescak introduced a data-sensitive and unordered production rules [5] enlarging the generative functions in shape grammar applications.

In this paper, we introduce a Dynamic Shape Representation (DSR) which can provide better support not only to design simulation and analysis, but also to design creation at a conceptual level. By using the DSR, several logic clues can be found for generative production of 3D conceptual designs. A shape grammar in product design using the DSR can record the design process in terms of design actions, and support emergent shapes in a more flexible way than traditional approach which is based mostly on the shapes.

## Significance

Designers are those with visual sensitive insights and emotional feelings towards forms, who focus on the shapes, or the ways of transformation. In a design domain, all the design methodologies and design philosophies serve the purpose of finding novel design solutions or new creations. Simply speaking, the status of a design in terms of shape is the most interesting issue for a designer to start with. However, for design researchers, the process of generating a good design is more essential than finding the design solution itself. In this context, the design process means the way in which the shapes are transformed from an original concept to a final solution. The dynamic nature of this design process holds more useful information than the shape itself on the design evolution. This dynamic nature needs to be understood by researchers and modeled in computer based design support systems. When CAD researchers appreciate a design, they will consciously or unconsciously try to find the reasons in terms of how to generate (transform) it from the shapes with which they are familiar.

In the development of shape grammars, the shapes are the focuses. A shape set is one of the four key factors in the definition of Shape Grammars [6]. The main inference body of shape grammars, i.e., the rule, is formed based on shape replacement. How to represent a shape can directly influence the effectiveness of shape grammar applications. Popular representations of geometric shapes include: B-Rep (Boundary Representation), CSG (Constructive Solid Geometry), Variation Geometry and Feature Representations. Briefly, B-Rep approach focuses on the boundary information such as faces, edges and vertices. The CSG approach uses a set of primitives and a set of Boolean operations (the Set operation [7]). Variation modeling allows designers to use formulas to model design components, and feature based approach builds shapes by feature primitives extracted from the design samples.

The current shape representations mostly describe the geometric and topological characteristics of shapes in a static way. In this paper, we use a dynamic viewpoint to approach the shapes, which can help CAD researchers to find clearer clues for generative production. Adopting the CSG method, we chose the basic primitives with a rule-based approach to describe the generating process, named Dynamic Shape Representation (DSR). The DSR can support more flexible representations through controlling the variables embedded in the initial shapes or changing the primitive shapes without changing the rules. In the following sections, the details of the definition and the application of the DSR are presented.

# Method

## *Understanding Design Feature*

A design cannot merely be explained by logical information, as there are some intangible factors which are more significant than tangible aspects. Some of these intangible factors cannot be fully understood by the researchers in AI and Computer Sciences. Therefore, there is an assumption which says that before designating a tool to support design, one should learn from designers first. When developing new tools or computational methods supporting design activities, it is necessary to try to follow the thinking of designers instead of the information process which is more convenient only for computer representation and manipulation.

At conceptual design stage of the design process, there are three essential features governing the complexity of the matter. These are dynamic feature, environment sensitive feature, and multilateral feature. Addressing these features in design research is useful for pushing the design technology to an advanced level.

Dynamic feature—From an original design conception to the final design solution, there are always dynamic changes until the design process terminates either by a designer's decision, satisfaction of the objectives, or some other reasons. In the design process, a designer's thinking process develops in highly activating manner. This process may inspire the designer because of a tiny change in the information and the knowledge that are both evolving. In a shape grammar system, it is therefore difficult to enlist all the potential shapes in advance since many of these shapes may only appear in a designer's mind as the design process goes on. It is also difficult to model and trace the whole evolving process, as it is most likely to be influenced by some intangible factors, such as the feelings, emotions, experiences, subjectivity, and even chances. A common factor is that the changing will not stop until a final design solution is confirmed. There is therefore a need for a representation scheme capable of not only describing the tangible design shapes in static terms such as geometric or topological features, but also in a dynamic way by the rules of grammatical mechanism.

Environment sensitive feature—At conceptual design stage, a designer may change parts of the design not because the parts are unsatisfactory, but because they are unsuitable for the entire design. In this situation, a family of rules instead of a single one can describe the shapes and its variations better in terms of supporting the design thinking rather than simply creating the design results. This is because some better results may be created unexpectedly by the ever changing of design thinking of the designers.

Multilateral feature—A designer's thinking has unlimited boundary and scope. Even designers themselves sometimes cannot speak clearly how an idea is emerged and why it is so emerged. The same design may require different imaginations of different designers with different backgrounds. With no technical restrictions, if we ask several designers to create one same shape, there will be much more than one way for them to accomplish it. In other words, there are diverse ways which can build the same result. Nevertheless, some of these diverse ways will be better than others in different design contexts.

## *Dynamic Shape Representation*

The shape grammar mechanism has powerful analyzing abilities to keep the styles and characters of the design of architecture, product or art applications. Most applications of shape grammar are concerned with simulating some specific designs, and then generatively generating large amounts of similar candidates for designer's selections. In the rule base of a shape grammar system, every single rule represents one replacement of shape/sub-shape in a shape set. Rule replacement only focuses on the partial transformation of a shape, not the entire effect. When we restrict the rule application conditions, the possibilities of generating novel solutions will be reduced. When we relax the limit on rule application conditions through a generative product system, large useless productions will be created which will decrease the effectiveness of the system.

In this paper, we emphasize the three features of design generation process discussed above and introduce a rule-based shape description on three dimensions, referred to as Dynamic Shape Representation (DSR), with the aim of increasing the efficiency of shape grammar with generative capability.

The CGA shape [8, 9] (Computer Graphics Architecture), a novel shape grammar for the procedural modeling of CG architecture, produces building shells with high visual quality and geometric details. Complex modeling can be generated by the operations of several primitives including cylinder, rectangle, sphere and so on. Similar with the CGA shape, we use some fundamental primitives as the elements for basic rules, named Elemental Rule (ER). In this approach, a shape is composed of several ERs which can be applied in a sequential way. Each single ER is the same as the rule in a general shape grammar process, which describes a partial transformation of the object shape. The formal definition is as follows:

**Definition 1** *Elemental Rule* An Elemental Rule is a finite set of shape rules of the form $\alpha \rightarrow \beta$, where $\alpha$ and $\beta$ are the shapes being included in the permitted primitive shape set with an orientation, called ER(i).

Normally, the primitive shape set contains the basics which are easily expressed in a computer system, such as cube, sphere etc. For the consideration of feasible implementation in computers, any shapes that can be generated by definite Boolean operations do not belong to the primitive shape set.

**Definition 2** *DSR Initial Shape* A DSR initial shape is a collection of definite primitive shapes by union operation.

**Definition 3** *DSR Shape* A DSR Shape is a finite set of shapes, which are generated by using a family of Elemental Rules in a specific order. A DSR Shape can be formally represented by the family of ERs with the application sequence, {ER(0), ER(1), … ER(i)}.

A simple example is shown in Fig. 1.

**Definition 4** *DSR Shape Grammar* A DSR Shape Grammar has four components:

1. S is a finite set of DSR Shapes;
2. L is a finite set of symbols;
3. R is a set of rules of A → B, where A is a labeled shape in S, B is a labeled shape in $S^+ (S^+ \supseteq S)$; and
4. I is a labeled DSR Initial Shape.

Figure 1a shows a DSR Initial Shape, which is composed of a union of eight cubes. Figure 1b shows a DSR Shape, following a sequential application of ER(1), ER(2), ER(3) and ER(4), formally represented as {ER(1), ER(2), ER(3), ER(4)}. Figure 1c is the default direction of all the shapes in the working environment. For 3D shapes, one elemental rule can be divided into several concrete elemental rules based on different directions. For example, ER(1)–ER(4) shown in Fig. 1 actually belong to one rule. With different directions in a three-dimensional space, this 3D rule can be divided into 8 different ERs.

From this example, we can know that a DSR shape is not similar to the current status-based shape description. DSR shape is a family of ERs without limiting the size, construction and color of the object shape.

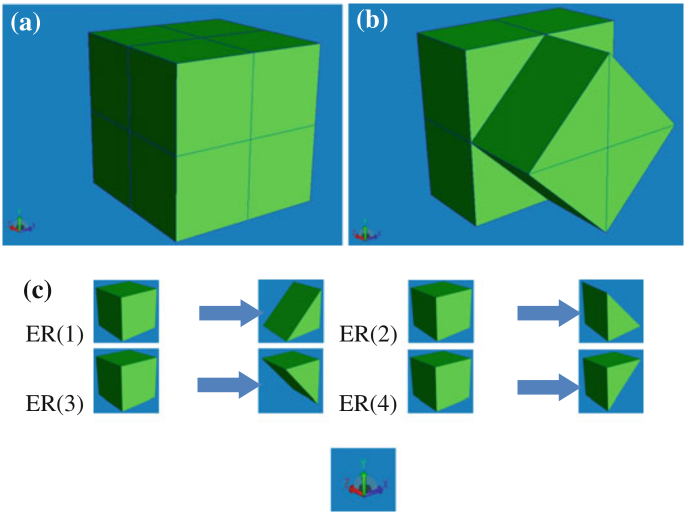


**Fig. 1** Example of dynamic shape representation. **a** Initial shape, **b** shape after transformation, and **c** rules for transformation

Based on a cognitive theory of computer design introduced by Ramscar et al. [10], in a computer system, all the proposed geometric systems can be a closed-world system aiming at expressing one category or more general real-world objects. Any ‘building’ in the system is represented within geometrically numeric information. Unless we can generate all the real-world objects by the system, we cannot claim that the system is completely close to the real world. Therefore, it is difficult to prove any ‘closed-world’ in a computer that can strictly correspond to the real world. However, a DSR shape is not a closed system of description. With the DSR, only the transforming actions are recorded in the form of Elemental Rules. For the shape set S in the DSR shape grammar, there is no restriction on the configuration of initial shapes. Theoretically speaking, any types of shape can be the shapes belonging to S that is operated by the DSR shape grammar. The DSR shape is not focused on static shape description, but the design generating process, which can describe the sequences of design operations for creating a conceptual form or a detailed form.

The DSR shape is an open description. We create a DSR shape on the basis of current classical modeling technique, i.e., CSG shapes. For different composition of primitives, the application of DSR rules can generate various novel shapes.

The definition of the DSR shape is based on the DSR initial shape. In practical usage, the DSR shapes are not only generated from DSR initial shapes. The Elemental Rules can be applied to any types of initial shapes. So we need to define the details of Elemental Rule application and primitive matching in ERs.

**Definition 5** *Primitive Matching* If $\mathrm{Vpi} \geq \mathrm{Vec(pi)} * \mathrm{TS}$, then the primitive(i) is found, named 'Match', else primitive(i) is missed, named 'MisMatch', where Vpi is the volume of primitive(i), Vec(pi) is the volume of external cube of primitive(i), TS is a threshold $\in (0, 1]$.

Normally, we can use a cube as the primitive of DSR shape. During DSR shape applications, if the volume of object primitive is less than the required value, then the object primitive will be recognized as missing.

As the DSR is based on the combination of several primitives, it is therefore a shape description but not a design description. There is no need to consider too much of the sub-shape detection problem. Primitive Matching plays the same role as sub-shape detection comparing with the static shape representation approaches.

**Definition 6** *Application of Elemental Rule* For ER(i) $\alpha \rightarrow \beta$, if object primitive matches $\alpha$ successfully, then object primitive will be replaced by $\beta$, else the application of $\alpha$ will be ignored and deleted.

The primitives are already the fundamental elements of the DSR shape. If a primitive is less than the standard requirement, decided by Vec(pi) * TS, then we consider that it will not work for the final DSR shape. This primitive will be dismissed. Using the same DSR shape in Fig. 1, the three DSR Initial Shape in Fig. 2 can generate the same DSR shape after using ER(1)–ER(4).

Considering the practical effect, Definition 5 and Definition 6 will not reduce the function of DSR shape for minute shape transforming. If we define a DSR initial shape with a high granularity (such as 10*10*10), or even higher, then tiny changes can be captured by ERs with efficiency.

## Experiment Results

DSR shape can support generative methods to increase the effectiveness in creating design alternatives. In this section, we present two 3D experiments to show the usage of DSR shape in grammatical applications.

### *Case 1: Generative Power of DSR*

There is one 5*5*5 cube combination as the DSR Initial Shape. We support three basic rules, Delete Rule, Slice-edge Rule, and Slice-corner Rule. In the 3D coordinate system, the orientation will increase the number of Elemental Rules. So, in a DSR shape application, the orientation factor needs be considered.

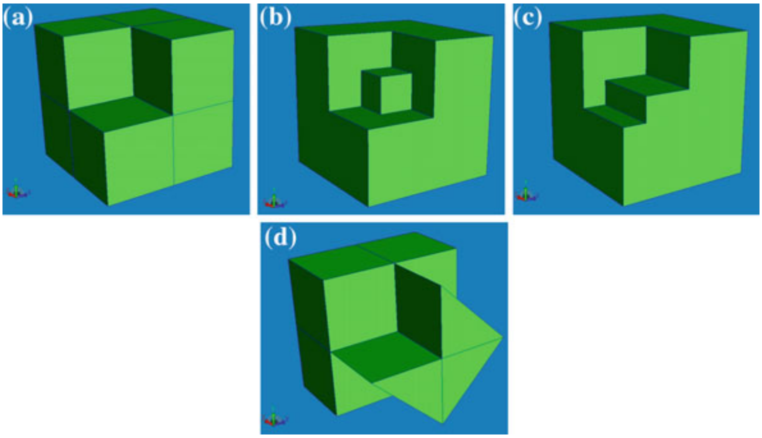
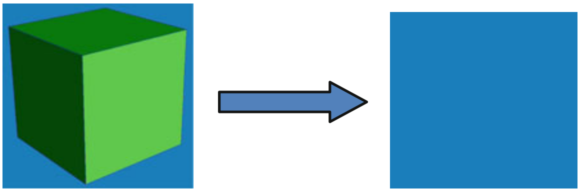
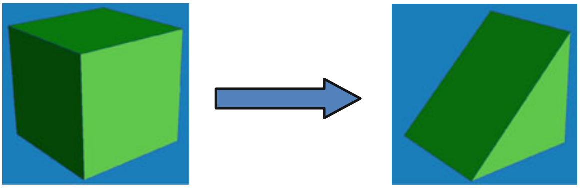


**Fig. 2** DSR shape application; **a**–**c** are three different initial shapes, and **d** is the DSR shape after applying ER(1)–ER(4) shown in Fig. 1, TS = 0.3

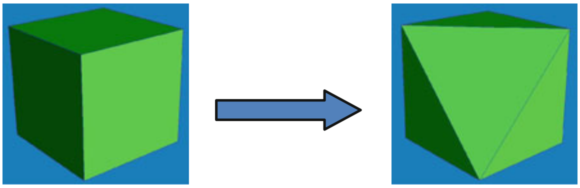
- Delete Rule. To empty the object primitive.



- Slice-edge Rule. To slice the object by one edge of the object primitive



- Slice-corner Rule. To delete one corner of the object primitive



Concerning the different orientations, there is one ER1 generated from Delete Rule. There are ER2–ER13, generated from Slice-edge Rule, as one cube has 12 edges. There are ER14–ER21, generated from Slice-corner Rule, as one cube has 8 vertices.

The DSR Initial Shape is shown in Fig. 3.

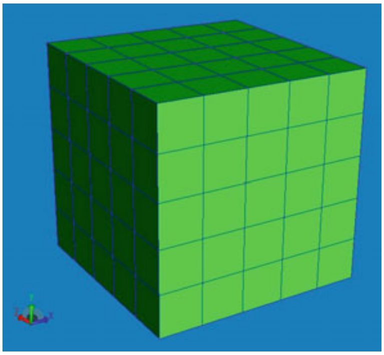


**Fig. 3** The DSR initial shape for experiment 1

After sequentially using Elemental Rules (ER1–ER21), there are two types of DSR shape shown in Fig. 4. Figure 4a, b are the balanced shapes, while Fig. 4c, d are the imbalanced shapes.

There are 45 sequential ERs to generate the balanced DSR shape1. There are 85 sequential ERs to generate the imbalanced DSR shape2. As the Elemental Rule has orientation, when we change the orientation, the results will be different. The primitives of DSR Initial Shape belong to a collection with the inside relationships of each other. Each primitive will only be replaced once by ER application or none replacement. Therefore, when we change the ERs' application sequence along x-dimension, y-dimension or z-dimension respectively, the replacements will be changed, which will transform the entire results of the DSR shape.

For DSR shape1, we use the first method to change the orientation of the ERs. In Fig. 5, there are many new shapes generated based on DSR shape1. Some of those are not interesting enough. However, there are some common characters which are kept. The Slice-corner rule is used to the eight corner areas and the Slice-edge rule is used to the mid part of the body. Because only the external changes of 3D shape can be observed, there are some replacements inside which cannot be shown in Fig. 5. From this experiment, we can see that when the orientation is changed, the DSR shape will be transformed. Clearly, the orientation of ER is important to the aesthetic value of the result. The massy feeling of Fig. 5 is due to the random orientation selection. In our future research, how to change the orientation in a regular way will be tackled.

For DSR shape2, we use the second method to generatively create some new shapes. After changing the application sequence along three axial directions, some of the results are shown in Fig. 6.

Obviously, there are two acceptable ways to generatively produce new shapes based on DSR shape. Furthermore, the DSR approach can increase the novelties of shape generation deriving from the existing solution. DSR shape can also provide more clues to improving the generative production method with complex 3D shapes. This is the reason why we want to record the shape generating process, in addition to the geometric information.

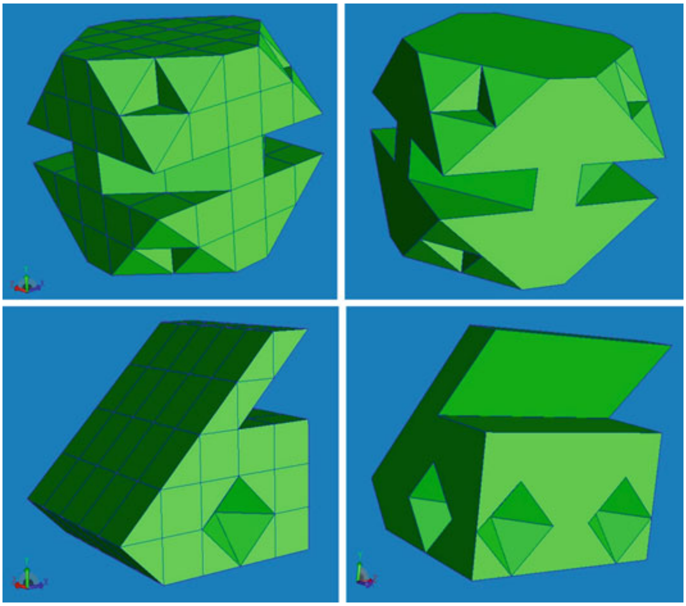


**Fig. 4** The DSR shape for experiment 1

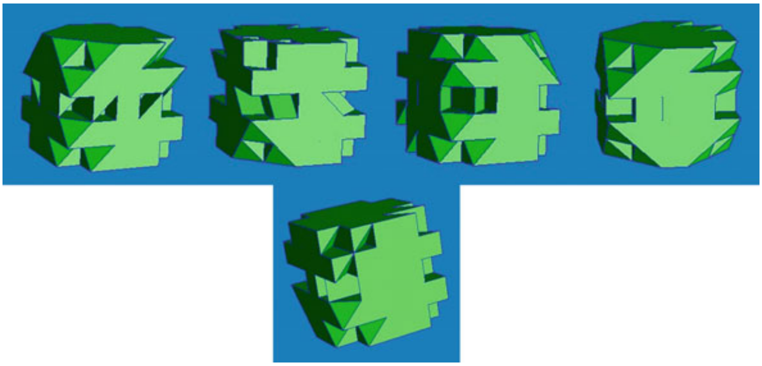


**Fig. 5** Generative generation of DSR shape1

## *Case 2: Application in Shape Grammar*

Using DSR shape in shape grammar can also create more alternatives while keeping the design intention. In this section, we illustrate the generation of a tea table design with the DSR shape grammar, Fig. 7.

We can create five rules expressed by DSR shape. The '*cave_surface*' rule is used to generate the tabletop. The '*chop_width*' rule is used to curve the side of

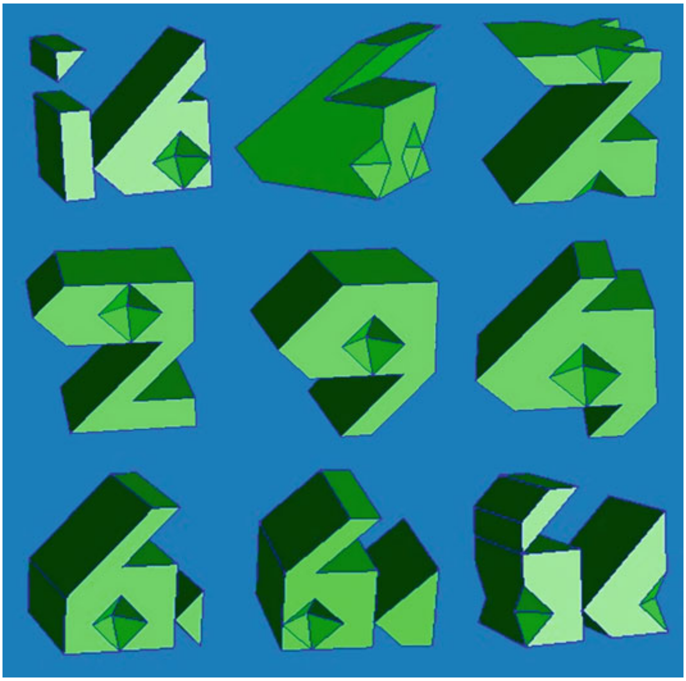


**Fig. 6** Generative generation of DSR shape2

tabletop. The '*framework*' rule is used to generate the main structure of the table. The '*side_leg*' rule is used to cave the shape of table leg. The ***'leg_even'*** rule is used to even the table leg.

As the DSR shape records generating actions, not only the geometric action, we do not need to consider the final result of the tea table when we create the rule. During the rule creating process, the thing that we need to be concerned with is how to transform DSR Initial shape with design intention. In this application, all the DSR Initial shapes are based on cube.

In this application, the initial shape is a cuboid of 10*5*5, shown in Fig. 8a, and the standard tea table generation is shown in Fig. 8b. With the shape rules generated in the form of DSR shape, we can generatively transform the 5 rules to create more new options. There are some new tea table solutions shown in Fig. 9.

## Discussions

A single Elemental Rule (ER) in our Dynamic Shape Representation (DSR) represents the shape transformation aiming at modeling a specific design intention. Using a family of ERs to represent and transform a shape is considered a new approach to supporting the dynamic nature of design process.

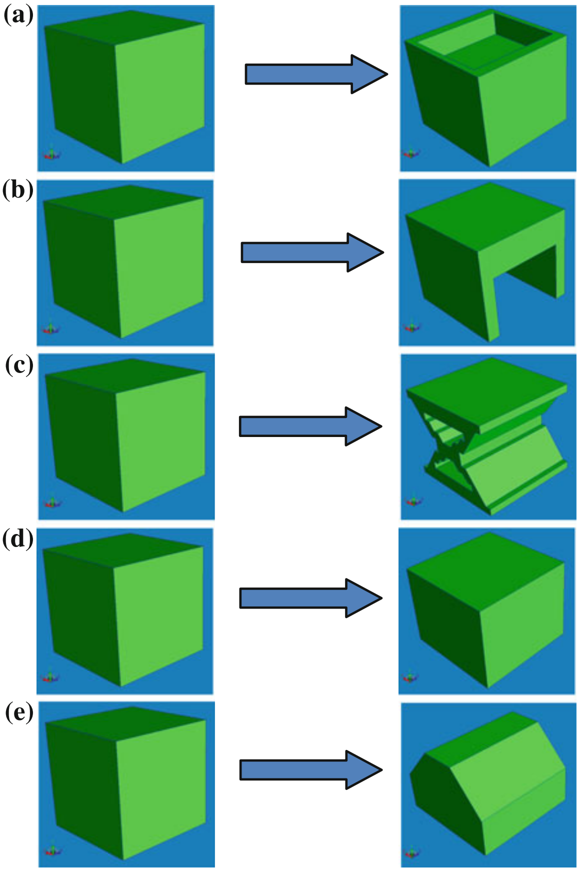


**Fig. 7** DSR Shape rules for tea table generation; **a** 'cave surface' rule, **b** 'framework' rule, **c** 'side leg' rule, **d** 'leg even' rule, and **e** 'chop width' rule

In DSR, a group of ERs focuses the transforming actions on the initial shape. Different initial shapes using the same ERs will generate variable effects. The essential idea of DSR is to record shape changes in the design process. This will provide more flexible options for the rule developers. Therefore, the DSR shape grammar can be used in design simulation, as well as in the creation of novel design candidates by the designers. Our ultimate objective is to allow designers to create their own shape grammar rules reflecting their styles or preferences.

For a DSR shape, each primitive is one member of the whole shape set, which means that there are certain relationships among them. When the ERs are created by the rule developers the influence of a single transformation on the entire design is considered. In other words, the group of ERs has some inherent connections, not just the simple union of each single one. In our current research, we have used the sequential order to control these inherent relationships among ERs. Different from parameter-driven CAD technology, the requirement for a DSR shape is only on the final design effect. The ERs are applied with sequences and we do not limit the

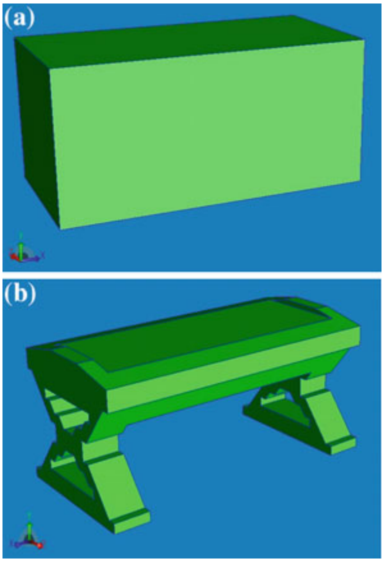


**Fig. 8** Initial shape and tea table generation. **a** Initial shape and **b** tea table generation

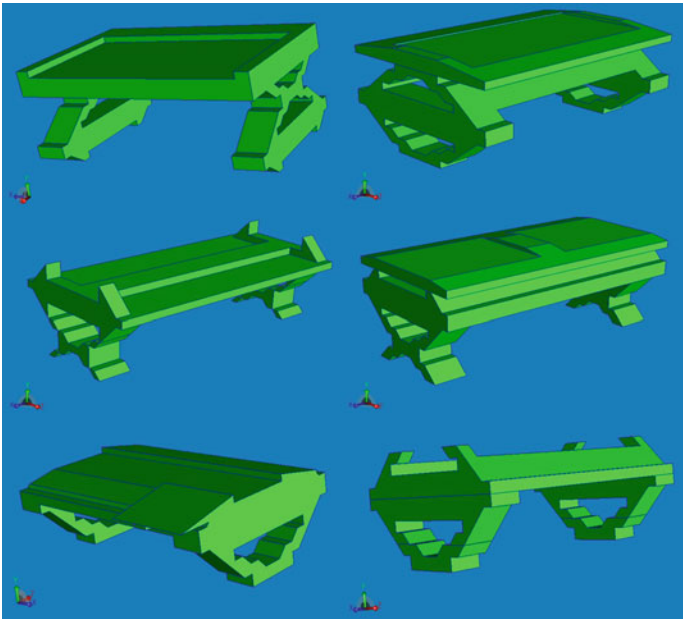


**Fig. 9** Generatively production of tea table

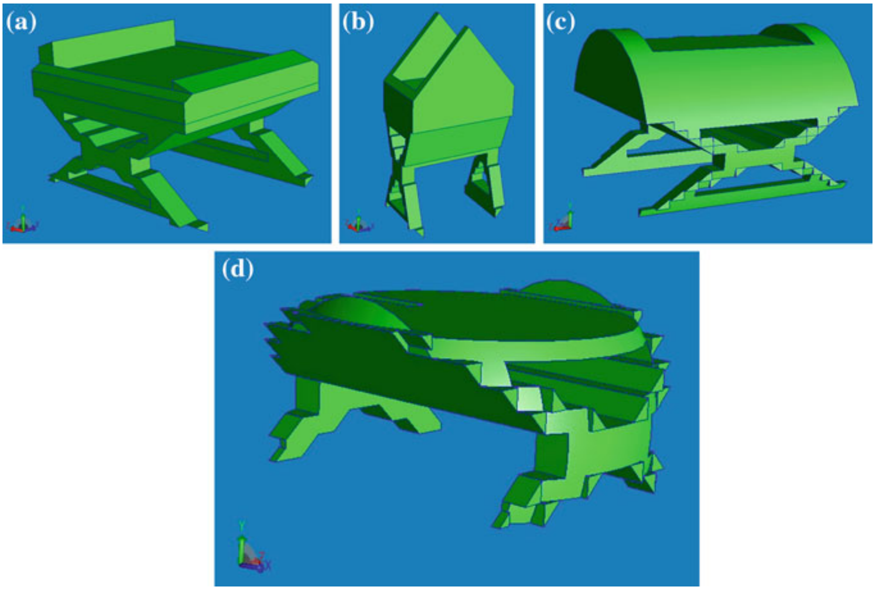


**Fig. 10** Tea table generated from different initial shape. **a**, **b**, **c** and **d** unexpected shapes

order of these sequences. Therefore, for the same DSR shape, the representation may be non-unique. This can allow designers create their desired shapes in a free environment. However, the non-uniqueness in the representation is also a potential problem for design computation, which is being studied further in our research.

Excepting for advanced generative power, another merit of DSR is the support for emergent shapes. Since we only use ERs to transform a shape, applying the same rules to a different initial shape can create some unexpected design effects. As shown in Fig. 10, when we used the DSR rules of standard tea table shown in Fig. 8a to the initial shapes of sphere, cylinder, and prism, three different effects were generated.

There are some unsatisfactory characters of our current DSR shape, too.

When we use the DSR shape rules to an irregular initial shape, such as sphere, similar to the one shown in Fig. 10d, the rule application mechanism which is based on shape replacement will create some edges and corners that are unsuitable for maintaining smooth surfaces. How to make the DSR shape rule created based on orthogonal primitives more suitable for curvilinear surface shapes such as sphere, cylinder or other irregular shapes, is a problem for our further investigation.

Because the DSR shape is not only aimed at design simulation, but also design creation, a real-time visualization support will be necessary. A 3D visual shape interpreter is required for further development of our DSR shape grammar system.

DSR shape focuses on the shape interpretation, not the design description. The ERs can describe one shape well for the effective support to the applications of shape grammar. For practical applications, however, the DSR shape should be working with a structured shape assembly system in order to support more complex product design tasks.

## Conclusions

In this paper, a new representation of 3D shape different from traditional geometric shape description is presented. We propose a Dynamic Shape Representation (DSR) as the basis for reasoning with 3D shapes in a generative system supporting product design. This representation focuses on design actions which can be formulated as shape grammar rules that are applied to a design session in the orders determined by the designers. As such, the DSR supports shape transforming actions, suitable for generative product design with the flexible usage of different kinds of initial shape. Several examples are provided to validate this representation, which have shown promising results that can be further explored. Our current work focuses on how to extend this representation to have more rules that simulate the intention of the designers. Our ultimate objective is for developing generative 3D shape grammars with which designers' knowledge and styles can be captured, while computational efficiency in the process of design exploration and evaluation is achieved through the DSR by allowing designers to create their own rules. At the current stage of the development of this research project, we are exploring the exploration power of the DSR in complex transformation actions. Our next step will be examining the rules, initial shapes and application orders of rules in order to create the designs that represent the styles of designers, or the types of products.

**Acknowledgments** This research is sponsored by the Hong Kong UGC research grant, the general research fund (project number BQ-20L).

## References

1. Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. Inf process 71:1460–1465
2. Knight TW (1981) Languages of designs: from known to new. Environ Plan B 8(2):213–238
3. Knight TW (1983) Transformations of languages of designs: part 2. Environ Plan 10(2):129–154
4. Liew H (2004) SGML: a meta-language for shape grammars. Massachusetts Institute of Technology
5. Trescak T, Rodriguez I, Esteva M (2009) General shape grammar interpreter for intelligent designs generations. In: 2009 6th international conference on computer graphics, imaging and visualization. IEEE, Tianjin, China, pp 235–240
6. Stiny G (1980) Introduction to shape and shape grammars. Environ Plan B 7(3):343–351

7. Todd S, Latham W (1992) Evolutionary art and computers, Academic Press
8. Parish YIH, Müller P (2001) Procedural modeling of cities. In: Proceedings of ACM SIGGRAPH 200, ACM Press
9. Müller P, Wonka P, Haegler S, Ulmer A, Van-Gool L (2006) Procedural modeling of buildings. ACM Trans Graph 25(3):614–623
10. Ramscar M, Lee J, Pain H (1996) A cognitively based approach to computer integration for design systems. Des Stud 17(4):465–488

# On the Evolution of Thoughts, Shapes and Space in Architectural Design

**Kinda Al Sayed**

**Abstract** With a common vision, studies that looked into the configurations of shapes and structures in architectural spaces agree on the possibility of externalizing a universal language to interpret architectural artifacts. Focused on the product of architecture, it is less clear how this language evolves in the course of design. Studies within the framework of protocol analysis have been devised to decode design cognitive activity. In what concerns architecture as a social artifact, these studies were detached from the syntactic and grammatical readings of architecture. In an attempt to bring the relationship that couples form and structure together with the pronounced mental activity we aim at capturing regularities that tie thoughts, shapes and structures as they coevolve. For the purpose of the analysis, verbal comments along with their associated hand-drawn sketches were tracked in progress focusing on how space is partitioned in real-time. The addition of partitions marks changes on shapes and structures of the designed spaces. The paper discusses parametric relationships between shapes and structures as they change over the course of design and tracks their associated cognitive behavior on a linkograph. Our findings suggest that architects even though starting from different design preferences plot similarities in the course of design thinking and doing.

## Introduction

During the last decades, considerable developments were made in the area of design methodology and design process [1]. Parallel to that, a significant progress was made on understanding and decoding spatial layouts. Decoding was either

---

K. Al Sayed (✉)
University College London, London, UK
e-mail: k.sayed@ucl.ac.uk

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_22,

seen as a matter of outlining formal relationships between shapes [2] or as a matter of understanding spatial relationships and their inherent social logic [3]. These methods were commonly used in evaluating architectural layouts. Less insight was given on investing such techniques in understanding spatial transformations as design progresses. Architects remained on the skeptical side of any effort made to outline models in design thinking and doing. As a result, no studies were known to link spatial transformations in design to their associated cognitive activity. For the most part, cognitive design studies particularly those based on protocol analysis [4, 5] were built on taxonomical models or graphs of progressive design actions that are extracted from designers' verbal statements rather than their sketches. Other studies have considered both mental and drawing activities in understanding design [6]. However, because of the broader understanding of what design is no specific case was made to architecture as a socio-spatial phenomena. This yields the need to incorporate a socio-spatial theory of architecture [7]. With such framework we might be able to understand how architects act as observers, users and makers of the built environment.

Space Syntax [3, 7] is a theory that represents space by its topological structure and translates its social meaning. The measures to identify a spatial structure are mostly borrowed from network theory with a value-laden reading of their social meaning. Spatial relations are interpreted through making sense of adjacency and accessibility relationships that link spaces together. With that, network measurements that capture adjacency and accessibility form the instrumental part of Space Syntax theory. The majority of Space Syntax studies read space as an ultimate product and shed less emphasis on the process of spatial partitioning that continuously transforms spatial structures. This is in spite of references made on the role of the syntactic model in filtering design possibilities [7]. In his model, Hillier identifies the generic function of space as the first filter used in any design process. Hillier explained the generic function as a configurational structure that makes all spaces in a design permeable by interconnecting them with an intelligible structure. The role of the generic function is essential in design seeing that design goes through a continuous process of evaluation to control the search for possible solutions and to filter out design possibilities. The evaluative power of space syntax remained to be invested at the final stages of design despite suggestions that an integrative role of such language in problem-solving would enhance on design outcomes [8]. Similar to space syntax, but lacking the social element; shape grammar were devised to recognize formal prelateships in a layout. With a computational perspective to comprehend architectural form, shapes were found to have some inherent grammar that can be obtained by modeling the proportions and translations of shape components [2, 9]. This approach was marked to be purely formal. It is based on recognizing rules that govern the relationships between shapes in a given layout or a given volume. The main sets of rules that govern shape grammars can be categorized as those that recognize the state of shapes at a certain stage, rules that govern the transformation of shapes and rules that identify the final state of shapes. The transformations are mainly based on geometric positioning and alterations hence have no known associations with an explanatory social theory such as that of Space Syntax [7]. A study to integrate both approaches has looked into the evaluation capacity of Space Syntax in determining the universe of design solution for shape grammar [10]. While this work brings the two methods in one design approach, it does not foresee a relationship between them. Instead it sees structural attributes as a mean to evaluate the socio-spatial performance of possible shape variations. Before building models to integrate both methods, it is essential to understand how the relationship between shape and spatial structure builds up over the course of design. For us not isolate the psychology of design thinking from its graphical product it is crucial to track how mental activities are associated with major transformations on shapes and spatial structures in sketches.

An interpretation of the design rationalization that precedes design actions can be revealed through investigating the nature of the cognitive activity that reflects the deployment of design knowledge into design decisions materialized through sketches. It is acknowledged that drawing in design is a technique to externalize design ideas in the form of visual representations [11, 12]. After configuring a situated design problem, architects retrieve knowledge by recalling memories of relevant design situations or user-based experiences. They construct analogical models in which they implement memories into restructuring user experience in space and use that to support the way they reason about their drawing actions. There is no clue on how to indicate when architects reason about their drawing actions. Goldschmidt [4, 13], and [14] attempted to reveal that through a model that clarifies how design actions link backwards to preceding design actions and how it links forwards to subsequent ones. It is yet ambiguous whether this reasoning process takes place before finding solutions or whether architects build conjectures about possible design solutions and follow that by post-rationalizing their design proposals.

Goldschmidt's methodology was adapted in the current research and crossed-over with drawing activity to look into regularities in the cognitive and drawing activities of architects approaching their designs with different preferences. For that, we take two extreme cases from a previous study [8] where one architect starts from socio-spatial preferences and another whose preferences are driven by natural lighting requirements. The two architects are compatible in terms of professional design experience. The targeted model incorporates the geometric dimension along with the configurational evolution of sketches and relates both to the linkograph as a linear plotting of the network structure of design thinking. First introduced by Goldschmidt [4], the linkograph is a model that represents the semantic segmentation and relational structure of cognitive activity in a form of a network. The network is projected against the sequential progression of *design moves* as 'acts of reasoning' that transform the state of design. Whereas Goldshmidt's definition for design moves is based on pronounced design thoughts. How to delimit these thoughts to distinguish design moves is subject to common sense agreement by arbitrators. We measure these design moves of Goldshmidt's against transformations in the layout itself to capture any regularities that couple thoughts and drawing actions.

## On Coupling Cognitive and Drawing Actions

The scope of this paper is to introduce a novel methodological approach towards understanding the relationship between evolution of spatial structures in sketches and the parallel cognitive activity that is made discursive through speech and drawings. This is not to say that there is no non-discursive cognitive activity that is internal to the design process. What we are mainly concerned with is the cognitive and physical activity that externalizes design thoughts. We distinguish here between the process of cognitive activity that is pronounced through verbal comments only and the moments when drawing activity takes place. Where both activities meet marks time periods of high productivity. The verbal externalization of design thoughts is intermediate between the deep level of concept formation where fuzzy conceptualization of designs is being processed through mental imagery and the physical design actions manifested by sketches. Sketches reflect on the state at which design concepts are confirmed through design decisions. Although this state might be preceded by a verbal statement, the physical manifestation of design decisions through drawing will mark the point at which an understanding of a prior state of design problem will lead to actions that will change the design problem. At these points, the design problem transforms from an ill-defined problem to become more defined in a progressive series of actions that are set to produce a unique design solution. The constraints assigned to define the design problem are subject to the architect's background knowledge and the sequence of knowledge retrieval and implementation that is applied towards defining design solutions. The prioritization of certain constraints over others will lead to divergent approaches towards problem–solution definition. While it is acknowledged that individual differences always arise as a result of different academic or personal backgrounds, it is expected that architects will implement relevant knowledge into their design decisions. Studies were made to investigate cross-knowledge by looking at multi-constraint requirements, see [15]. Unlike them, we are mainly concerned with the within-domain knowledge and how it is utilised in filtering design solutions. The knowledge of 'Generic Function' is the core within-domain knowledge that all architects—no matter how different their backgrounds are—implement consciously or subconsciously in design. In addition to this knowledge architects cast other types of constraints such as lighting requirements, emergency planning and cultural idiosyncrasies in further defining design solutions.

Focus in this research is centered on outlining similarities that designs share as they progress. On that basis, a population of two is may be sufficient to give an insight into the associations between spatial transformations and further particularities in the structure of cognitive activity. The study presents an attempt towards understanding these associations given different design preferences.

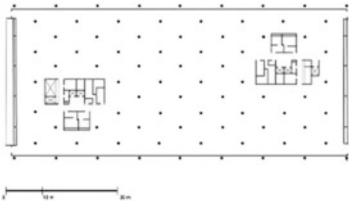
## Design Task

In this paper, two extreme cases were taken chosen from a population of twelve cases previously studied in [8] to represent divergent design approaches towards one design problem. We aim to prove that in spite of their differences, these two cases share similarities that couple pronounced cognitive activities together with visual representations. We distinguish activities that result with drawing lines that have direct effect on the geometric and configurational structure of the design outcome. The configurational structure is reflected through the topological network properties of convex spaces. The breaking of a layout into convex spaces involves defining the fewest and fattest spaces. These spaces are then mapped in a network to link those that are physically adjacent and share access points. The geometric properties of convex spaces as shapes are mainly defined by the area and proportional ratio of width to height. This may be further simplified by representing two diagonals if convex maps are composites of quadrilaterals or one diagonal in the special case where quadrilaterals are rectangles. This defines a parametric shape grammar model that differs from [10] approach. A parametric shape grammar model accounts for the generative capacity of Space Syntax to be coupled with the production of shapes rather than a pure evaluative capacity of the syntactic model. The generative capacity of Space Syntax can be explained as the mechanism in which a configurational structure directs the course of subdivisions in the layout.

The design brief that was given to the two architects was intentionally limited to a set of functional spaces that form the basic requirements for an architect's office. They were asked to allocate the required spaces in a given layout (see Table 1). The architects were asked to prioritise circulation while allocating functions into occupational spaces within the layout. The layout was intentionally restricted to one level rather than multi-levels in order to simplify the design problem and consequently limit design solutions. The layout was a hypothetical rectangular floor plan in a skyscraper with two access points from two cores. The given layout already presents some challenging problems that are related to its massive size, the number and pattern of columns that appear in it and the two cores that link it with the external environment.

The two participants; KS and CR, were selected upon the criteria of having a considerable design practicing experience. KS has 11 years of academic and professional experience in architecture. CR is an architect with 19 years of academic and professional experience in architecture. CR's experience was involving Space Syntax knowledge for over 9 years of what can be recognized as mostly an academic practice. In spite of the overall difference in the academic experience, the professional experience that is design-based is compatible (around 9 years each). Architects were asked to develop design solutions intuitively using sketches within a time duration of 15 min. KS finished the design task within 32 min, whereas CR finished the task within 15 min. To externalise their design thinking we have implemented the thinking aloud method [4], in which they were invited to

**Table 1** Design task including a brief for an architect's office and an existing layout

| The design brief | Design task layout |
|---|---|
| • Head office and private secretary space<br>• Waiting area with small exhibition<br>• Two meeting rooms<br>• Management offices (number: 3–4)<br>• Telecommunication offices (number: 2)<br>• Three spaces for consultants<br>• Five spaces for design teams<br>• Two IT offices<br>• Two technical studies units<br>• One construction expertise unit<br>• Two service areas with kitchenette |  |

declare their thoughts verbally. Their verbal comments were recorded and transcribed. Their sketching activity was also recorded using a webcam for us to track progress in their sketches. The process of partitioning space was then digitised and analysed in UCL Depthmap [16] to capture spatial transformations.

## Modeling Cognitive and Physical Design Moves

Attempts made to decode architectural design process were often focused on cognitive activity without much reference to how design solutions change over time. In this study we attempt to reflect on the mutual evolution of design thinking and drawing. We further observe changes in shapes and changes in structures and highlight relations in-between. We observe how reasoning and goal-driven actions contribute to decision making in design and how and when such decisions are executed in the form of drawing. Drawing activity is considered to be critical when lines drawn change the geometric and configurational properties of the design. We examine cognitive and physical activities and track them on a linkograph [13]. In a linkograph model, the cognitive activity is recorded, segmented and rebuilt in a relational structure that links design moves by matching their semantic meaning. In Goldschmidt's scheme the protocol is segmented to a set of 'design moves' with directed links. A 'design move' is explained as "an act of reasoning that presents a coherent proposition pertaining to an entity that is being designed" [13]. Links among moves, are determined arbitrarily by the observer, and are notated in a network. The design process can then be interpreted as a pattern of linked moves that comprise the graphic network of the linkograph. Goldschmidt identifies two types of directed links: Links connecting to preceding moves 'backlinks' and links connecting to subsequent moves 'forelinks'. Goldschmidt [4, 13] recognized that moves that have a dense linkage connection, both forwards and backwards, namely critical moves (CM), can be considered as indicators for design productivity. The trend of sequential, critical moves represents the critical path of a design process. Each design move involves retrieval of knowledge in a process of reasoning or goal identification.

An example of a linkograph structure is represented in Fig. 1. In part of this example the transcribed verbal comments have been segmented to design moves (moves 1 to 12). The segmentation of the design moves is mapped in a linkograph. Design moves are linked by nodes whenever design moves exhibit some association in terms of content. In this way it was possible to map the progress of design thoughts.

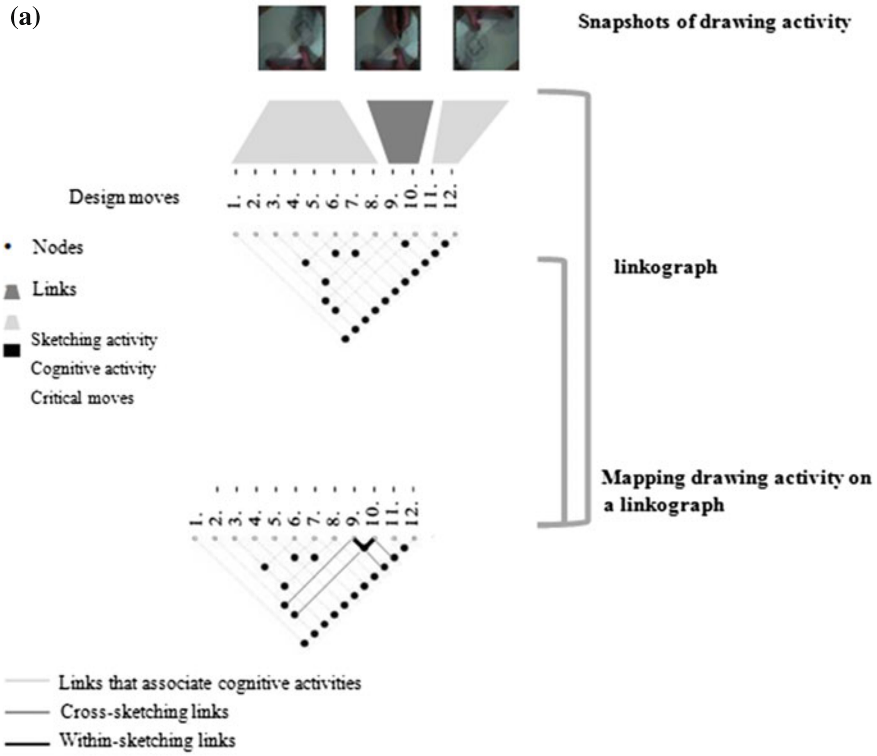
Tracking the evolution of designs themselves is made possible by segmenting the hand-drawing process into sketches that are correspondent to coherent sets of design moves. Setting major circulation corridors exemplifies one coherent set of actions. Dividing the edges of a layout to accommodate for a certain type of functionality is another. Drawing in the design process under investigation is beheld as a set of progressive actions that result with the partitioning of a given space. The partitioning allows for functional properties of occupation and circulation to be allocated in a spatial structure that links geometric shapes together. Architects partition space to fit in occupational spaces by means of optimizing user access and maintaining high level of communication between different members and groups in the prospective social organization. On the linkograph (see Fig. 1a, b), periods of critical drawing actions were marked to reveal any regularities or differences that mark the association between thoughts and acts.

Within-sketching links are identified as links that connect design moves within a continuous period of critical drawing activity. Cross-sketching links connect design moves that associate critical drawing activity, but are separated by intervals of verbalized cognitive activity. The remaining links connect moves that represent verbally translated cognitive activity with no drawing undertaken. It is acknowledged that cognitive activity runs throughout the course of design. With our segmentation we distinguish design moves that are associated with drawing activity; that is the translation of thoughts into design actions.

In order to detect regularities and distinctions in the data of the semantic transcripts, the linkography models in both design processes will be examined in direct relation to the drawing activities carried out by architects in the form of sketches. By distinguishing verbal protocol from executive protocol, more substantiation on how thoughts manifest into sketches may be externalized to expose design logic.

## How Do Thoughts Translate into Drawings in Design?

The continuities of drawing actions are delimited in the linkograph and separated by intervals where no drawing activity was undertaken. It is important to note that no hand gestures were considered. Drawing actions represent what appears to be critical drawing activity that results with partitioning, marking and linking spaces on the



**(b)** 1. So regarding the office plan layout, it is a rectangular shape,
2. two main cores,
3. a lot of columns,
4. I have two main elevations left and right,
5. I am not quite sure about this area here, is it just for the shape of the building from outside? Or is it not?
6. What about this columns here,
7. is there any neighborhoods here, can I open the side, can I have open views, I need like to think about this things,
8. regarding the inside. There is a clear network for the columns, which will affect the divisions of this functions,
9. but I need to understand first how can I reach the functions according to the main points which will affect the circulation around the cores;
10. and how this two cores will be working together,
11. because I am designing something for one team or for one firm,
12. this is in general the first impression about what I can see now

**Fig. 1** **a**, **b** A model of linkograph segmentation where the transcribed verbal comments are mapped. Further detail is introduced to this model by identifying how drawing activity is associated with cognitive activity

layout. The distinctions marked on the linkographs with regards to the structures of thinking expressed verbally and executed physically through drawing may highlight imperative differences and similarities (see Figs. 2 and 3).

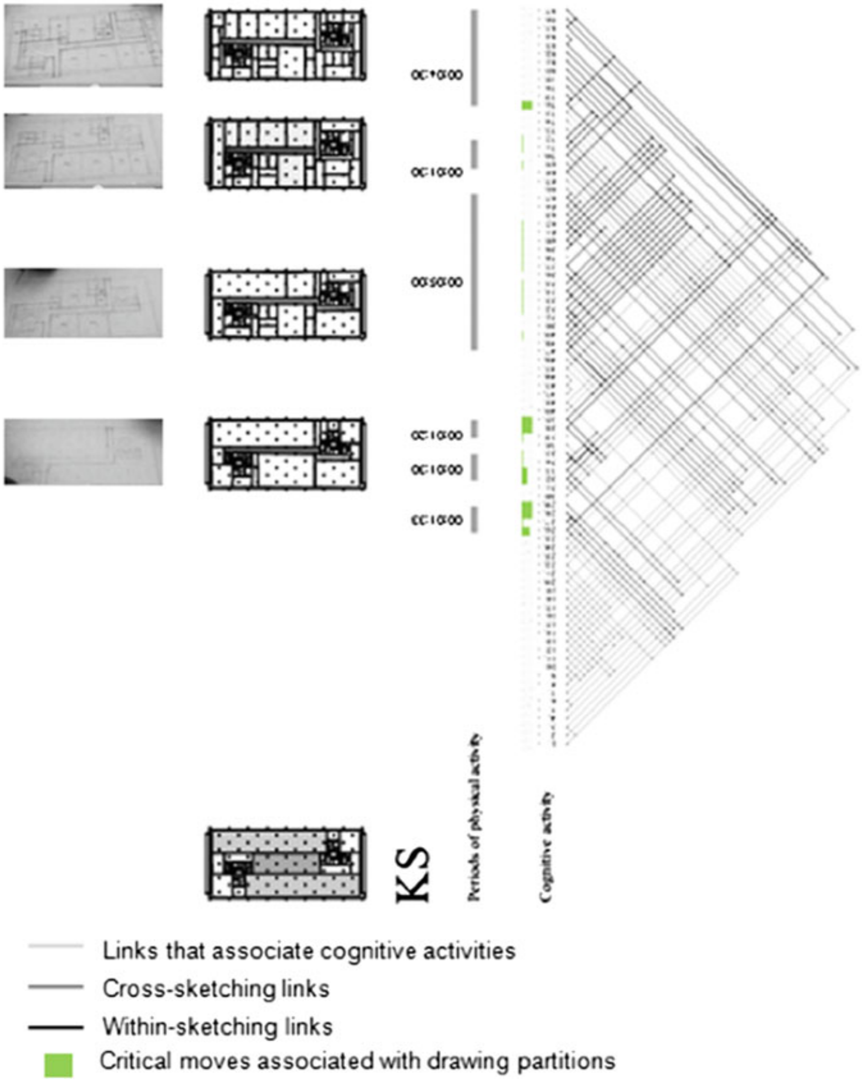


**Fig. 2** A linkograph representing the design activity of KS and marking the association between critical moves and drawing new partitions. The evolutionary stages of design as well as periods of sketching activity are attached

Some distinctions can be marked by distinguishing one aspect of the linkograph complexity, where design moves link within continuous periods of drawing activity or across these periods. By periods we refer to time durations that cover design moves whether associated with drawing activity or not. We differentiate between links that interconnect design moves within periods of drawing and those that connect in-between separate periods of drawing activity. The first is to be

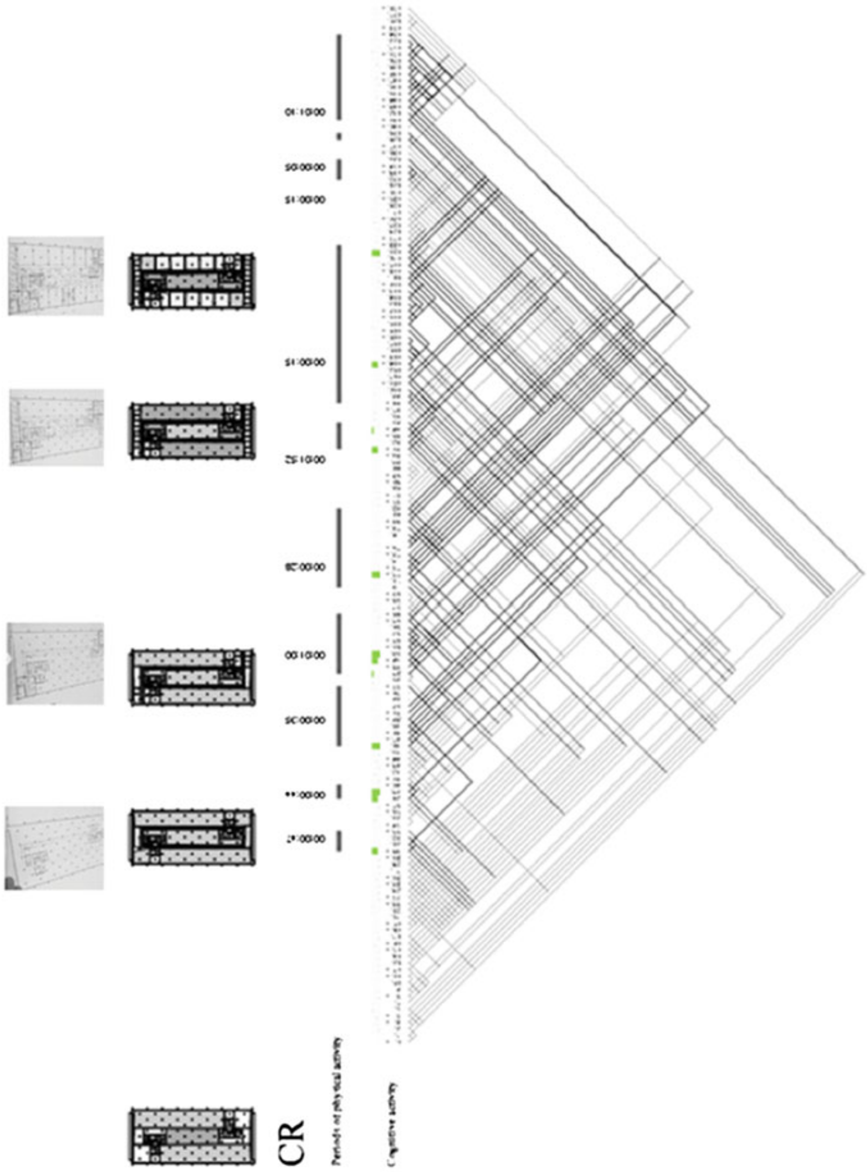


**Fig. 3** A linkograph representing the design activity of CR and marking the association between critical moves and drawing new partitions. The evolutionary stages of design as well as periods of sketching activity are attached

named as within-sketching links and the second is referred to as cross-sketching links. In both cases the aggregate number of cross-sketching links double the number of within-sketching links. This means that drawing actions are interlinked in terms of the semantic meaning they hold in spite of the fact that they might be separated by periods of cognitive activity that is not reflected through drawing.

In spite of the fact that within-sketching links are slightly higher than half the number of cross-sketching links in the case of CR there are events where the cross-sketching links are remarkably higher than the within-sketching links. These periods that have a sequence of (3, 5, 6, 8) seem to include high functional activity. In KS's case there are two outliers were number of cross-sketching links are by large higher than within-sketching links. These are in periods (3 and 6). In both cases, the number of backlinks between sketching activity and non-sketching activity go well above the number of forelinks. In CR's case it triples the number of forelinks, and similarly in KS case, the number of backlinks is five times bigger than the number of forelinks. This may indicate to the idea that before executing drawing actions, architects are likely to refer back to previous actions to accumulate a better understanding of the design problem.

It is remarkable to see that the average ratio of time consumed for verbally-explained design moves is 8 s per design move in the case of CR, whereas KS consumes approximately 32 s per each design move. This means that performing one design move in the design process would take CR remarkably less time than KS. The rate of drawing performance seems to double in the case of CR compared to that of KS. CR draws twice the number of lines and twice the number of convex spaces within periods of drawing activity that KS can draw. The rates of partitioning space per time unit do reasonably distribute uniformly across the design process with the exception of the initial stage of drawing in CR's case where CR takes more time to define the first partitions in her design.

In what concerns the way periods of drawing activity are distributed, there are some regularities that both cases share. Both KS and CR consume one third of the total duration of the design process to the design problem they are set to approach before undertaking any drawing activity. A differentiation can be made through noting that during the first third of the design process, the verbal statements of CR involve less than a quarter of the total design moves in the design process. Whereas one third of the design moves in KS's case are progressed through the first third of the design process. This may mean that, subject to the way architects' thoughts were externalized; CR was less willing to express his mental activity than KS. This might be an effect of individual differences or divergent preferences.

In order to externalize the pattern in which design thinking actively engages with critical drawing actions, an evaluation is needed to reveal the degree of integrity between the design process as a whole and segments where drawing and thinking activities are associated. For this, hubs on the linkograph are marked where associations can be detected between the actual process of drawing new partitions and critical design moves (see Figs. 2 and 3). According to Goldschmidt [4, 13], and [14], a 'Critical move' (CM) is an indicator of productivity. We apply this definition to find design moves that have more than (5, 6, 7, 8) connections whether backwards or forwards or in both directions. The phases at which 'critical moves' are aligned to drawing activity are basically where well-connected thoughts synchronize with critical drawing actions that materialize and execute these thoughts. By drawing lines on the layout, a partitioning of space takes place that would necessarily lead to changes on the spatial structure hence evolving the design situation. Partitioning space will also necessarily lead to shape fragmentation. We identify these critical drawing actions that lead to transformations on the layout structure and consequently lead to shape fragmentation as critical drawing actions. This process when associated with highly linked design moves will highlight critical decision making points where actions that transform the state of the layout correspond with higher order thinking activity that operate on a global scale. In essence, having outlined events in the design process where both Critical design moves and critical drawing actions appear simultaneously might prove to be very meaningful.

If we match up the occurrence of critical drawing activity with the peaks of critical moves, we can observe two patterns of design thinking and acting that appear in both design processes. In both cases, critical design moves are mostly associated with drawing activity. In KS's case they spread along periods of drawing activity. In CR's case, they are mainly positioned at the start of periods of drawing activity. This may waive with the idea that CR goes through intensive cognitive activity before undertaking critical drawing decisions While KS does that while drawing.

## The Evolution of Shapes and Configurations in Design

This part of the analysis is mainly focused on a quantitative evaluation of sketches as they progress over the course of design. The quantitative analysis will consider changes in shapes in relation to the spatial configurations calculated by means of Space Syntax measures. The aim is to see whether the spatial structure acts as a determinant factor in defining the parameters of shapes. Space Syntax explains the spatial structure as the materialization of the social organization that resides within. Normally, tree-like structures reflect deeper arrangements and a hierarchy in the social organization. Conversely, the provision of interconnected rings of movement in a layout offers choices for movement routes reducing control and depth. The relation between spaces might be 'symmetrical' if for example: A connects to B = B connects to A. Otherwise the relation is 'asymmetrical'. The total amount of asymmetry in a plan from any point relates to its mean depth from that point. This is measured by its 'relative asymmetry' (RA). To normalize the measure, a diamond (D) value is used. With that in use it is possible to compare graphs of different sizes by calculating real relative asymmetry (RRA). Spaces that are, in sum, topologically 'closest' to all spaces (low RRA) are the shallowest (most integrated) in the spatial network. They characteristically have dense movement through them. Those with high RRA are the most segregated.

Convex integration correlates with 'occupational' behavior in a building. Convex spaces are spaces where all contained points are mutually visible. The convex break-up map is formed by identifying the fewest and fattest convex spaces; the fewest to prevail. The greyscale colors of the convex map are distributed in five bands according to their integration values. High integration is marked in a darker color. Within the network of convex spaces we differentiate between different types of spaces based on their proportions. Some are more suitable for occupation and more likely to have a larger degree of convexity in their proportions, others allow for denser through-movement due to their elongated shapes. Some spaces may even contain both movement and occupational functionalities. In the arbitrary decisions about the fewest and fattest convex maps, preference was given to spaces that are recurrent in the evolutionary process of design and exhibit distinct functional particularities such as corridors or hallways in spite of their elongated geometric properties. This sets the rules for recognizing convex break-up map as the first set of shapes in the initial structure defined by the first sketch. By considering a convex break-up map as the set of primary shapes that are ready to transform following certain parameterized rules we expect a relationship of some kind between shapes and socio-spatial behavior.

The general feature that might be noted from the two sequences of sketches (see Fig. 4) is that elongated convex elements are generally well integrated. They are largely defined for movement as circulation pathways and corridors. They seem to be more integrated in CR's case compared to KS's. In general, fatter and larger convex spaces seem to reside in the lower integration part. Small and fat convex spaces seem to present the lowest integration values in both cases. In terms of the evolution of geometric and configurational properties of convex spaces in the sequential sketches, it is observed that elongated large spaces with average integration values are more likely to suffer from fragmentations. We define this property as the *generativity* of convex spaces. It might be important to add that integration values seem to spread more normally in the sketches produced by KS compared to CR's. This is an effect of the symmetry in the designs produced by CR. This symmetry results with an evenly distanced pattern of clusters in the diagrams. By assuming a parametric relationship between shape grammar and the overall values of spatial structure, we might be able to outline three regularities as potential rules. Elongated shapes have higher integration values and tend to preserve their proportions throughout design. Bigger and fatter convex shapes tend to be average in terms of integration but they tend to subdivide into smaller yet relatively fat shapes. Small fat shapes tend to have minimum integration values. They preserve their proportions over the process. These rules in their simplistic form can be inferred from the evolution of the design sketches in these studies. However in order for them to be verified and interpreted into a model, further case studies need to be explored and tested against these observed regularities.

A separate examination of the two design processes leads to further distinctions on the individual level. If we plot the same ranges of integration on the convex maps (see Table 2) we find that the series of layouts produced by CR were more integrated compared to those of KS's. The structure produced by CR evolves into a ring consisting of four segments and connecting the two cores. A different design approach was undertaken by KS who created a central corridor that connects the cores and allows movement to stem from it towards occupational spaces on the

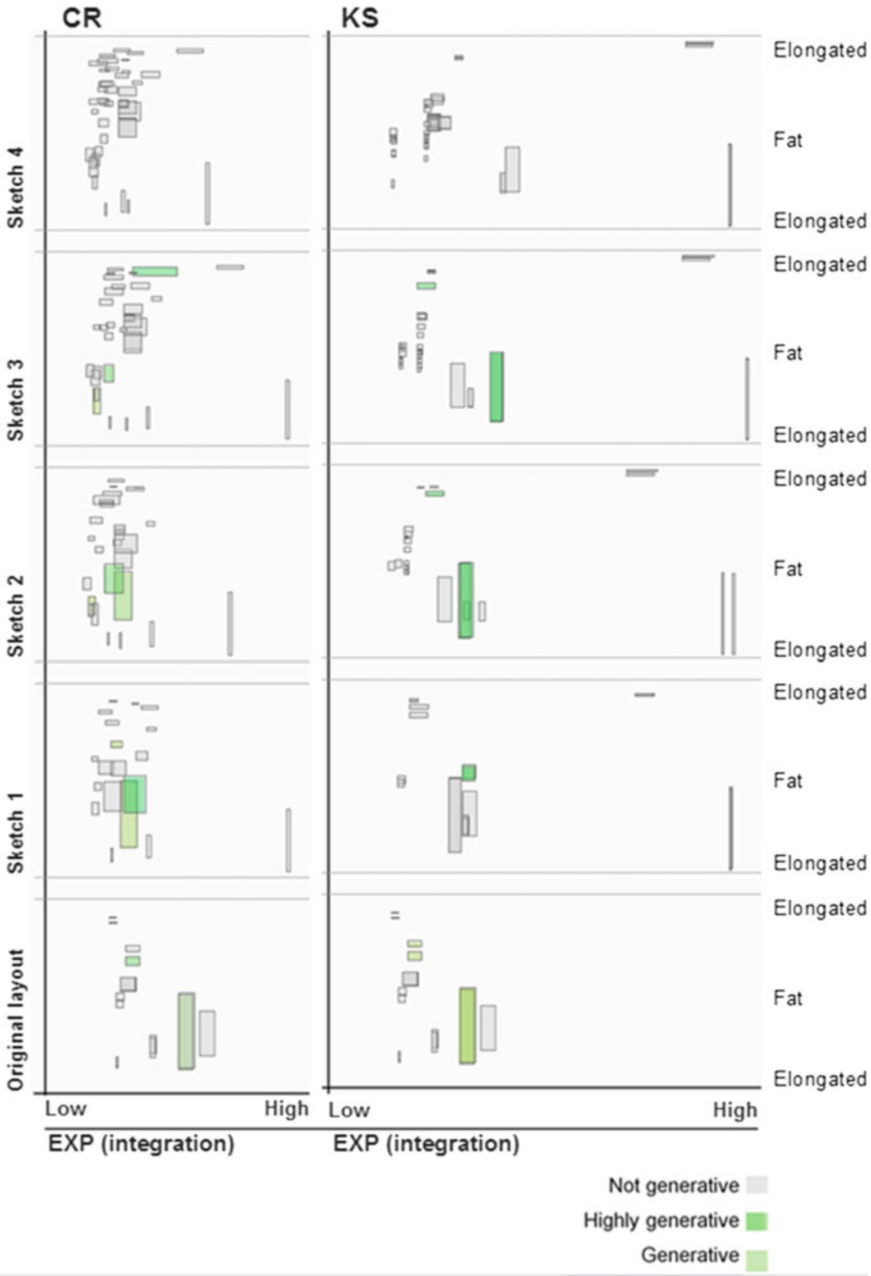
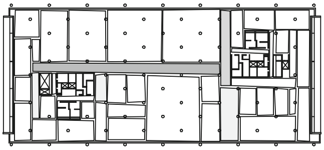
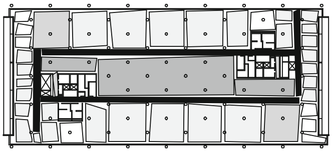
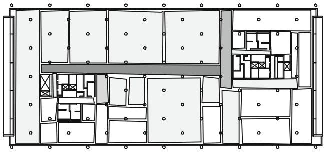
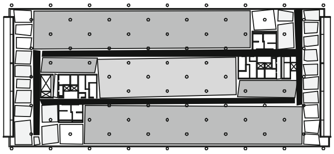
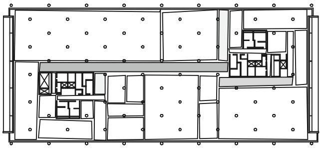
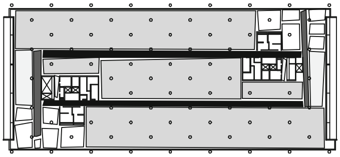
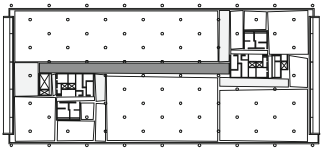
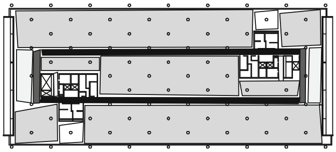
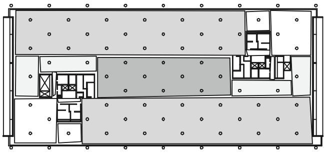
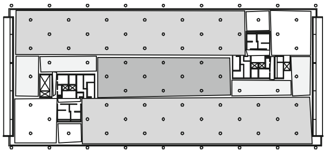
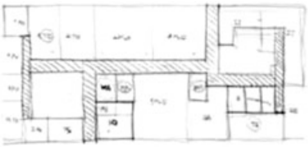
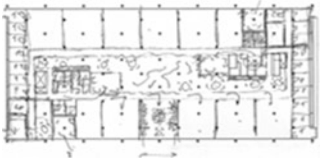


**Fig. 4** The relation between shapes and topological properties in KS sketches

**Table 2** Spatial analysis of the design proposals using Space Syntax tools [16]

| | KS | CR |
|---|---|---|
| Sketch 4 |  |  |
| Sketch 3 |  |  |
| Sketch 2 |  |  |
| Sketch 1 |  |  |
| Sketch 0 |  |  |
| Hand-drawn sketches |  |  |
| |  |  |

edges. In general, the quantitative analysis of structural properties of sketches emphasized the idea that CR was more inclined than KS to minimize depth in her sketches (see Table 2; Fig. 5). Despite these individual differences between the two architects who initially started from divergent preferences in their designs, they both produce structures the gradually become deeper stemming from the circulation routes towards the edges. While integration values spread with wider ranges in CR's case, they tend to spread less evenly in KS's case and compress again within a smaller range in the last sketch.

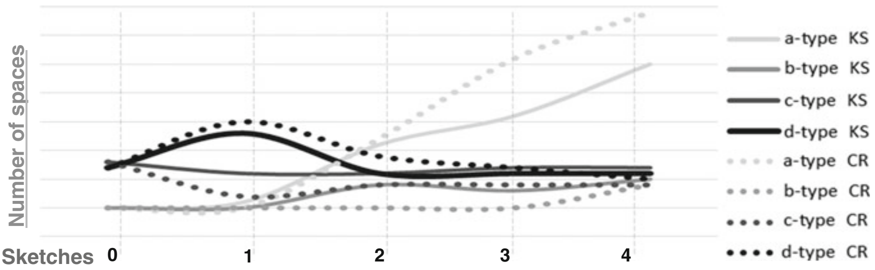


**Fig. 5** Changes in the number of spatial types that took place throughout the evolutionary process of spatial partitioning

Looking into a finer level of differentiation we may identify changes in the counts of different types of convex spaces that are classified by looking at their topological connections in a graph-based representation. This classification is introduced by Hillier [7] who differentiates between four types of spaces: a-types are characterized as dead-end spaces. They connect to no more than one space in the graph. B-types connect to two or more spaces in the graph without being part of any ring. Conversely, c-types are positioned on one ring. D-type spaces must be in a joint location between two or more rings. The positioning of these types of spaces within the local and global configurations of the whole network can determine the maximization and minimization of depth in a spatial complex. The increase of a-type locally and d-type globally minimizes depth creating an integrated system, while the increase of b-type globally and c-type locally maximizes depth resulting with a segregated system.

If we plot the trends of change on the different spatial types in the different stages of design (see Fig. 5), we identify some regularities in the way spaces are partitioned. At the same time as a-type spaces witness an exponential increase in numbers, the count of d-type spaces increases at the early stage and then gradually decreases. The count of c-type spaces witnesses a slight decrease across the different sketches. KS tends to present deeper hierarchies in his sketches where b-types and a-types appear early in the second sketch. C-types and b-types appear to be dominant throughout the process. In CR's designs, b-type of spaces only appears in the last sketch while a high number of d-types and a-types appear throughout the process. This means that CR approaches her design by minimizing depth. She does so through connecting d-type and a-type directly. KS is less likely to do so; he starts by presenting deep hierarchies from the second sketch resulting with a deep structure towards the end. In spite of these observed differences, the number of spatial types appears to plot parallel trends over the course of structural transformation. To confirm that, more data points and cases need to be observed.

## Conclusions

In this study we have adapted the linkograph model to distinguish periods of critical drawing activity. Similarities were detected in the behavior of two architects despite the divergent preferences they have undertaken in their designs. With regards to their sketches, we have also identified relationships between the parameters of shapes and their structural configurations. Moreover, we have seen that the count of different spatial types follow similar trends. For that we still need to define the exact similarity. More importantly, we need to validate these results on a larger population of participants. We also need to refine the experiment. In our current experiment we intentionally chose two divergent design approaches to be able to rule out that any identified similarities are noteworthy. CR's preferences were to prioritize circulation and occupation and how they shape the social organization that resides within the designed space. KS starts with allocating spaces in relation to maximize natural lighting and ends up by configuring circulation. These two approaches while expected to present distinct patterns of design behavior, they counter our expectations by revealing resilient similarities.

Observations suggest that thoughts that are associated with the drawing activity are robustly interlinked throughout design. Both design behaviors exhibit a tendency towards linking to previous design moves rather than future ones. An interpretation for that might be that architects keep referring to previous design moves to reason about their drawing actions. Both architects consumed the first third of the design process producing no depictions as they accumulate understanding of the design problem. There seems to be a strong association between critical design moves and drawing actions in both cases. Critical design moves are spread along drawing actions in the case of KS, while in CR's case critical design moves appear at the beginning of periods when drawing actions are undertaken. This may waive with the idea that architects consume most of their cognitive activity while executing drawing actions.

In terms of their designs, rules that define a parametric relationship between shapes and configurations are named to be three. The first rule is that highly elongated spaces are more likely to be the most integrated. Secondly, fat and small spaces are more likely to be segregated. Both types tend to preserve their proportions and values. As a third rule, large and relatively elongated spaces that have average integration values are more likely to be generative. They subdivide in subsequent stages. This simplified reading of a parametric shape grammar by structural configurations needs to be generalized across a larger sample. Another regularity that might be inferred from the partitioning of space is that rings of movement and circulation tend to be first defined in designs. They are more likely to reduce in numbers on the account of an exponential increase in the number of dead-end spaces. In that, architects are inclined to define circulation first and then progress to allocate occupational spaces. In this process, hierarchies that are subject to the conditions of the organizational structure of the building type contribute to a gain in depth.

The findings are subject to the limitations of the arbitrary segmentation of verbally interpreted design thoughts. Additionally, decisions were made to separate different stages of drawing by means of drawing order have been set by observing stages where certain functionalities were positioned all at once. For that, more systematic procedure is needed. It must also be disclosed that the convex brake-up is not a product of a unique process of reduction. Matching similar spaces was needed to construct a comparative setting across the successive sketches. It might be safe to say, however; that these inadequacies are not likely to have serious implications on the final results particularly with regards to the convex break-up representations. This is given that our purpose is to define similarities not differences between divergent design approaches.

The coupling of thoughts, shapes and configurational structure in both design processes indicates an inherent model in the structure of design thinking and doing. This needs to be confirmed by generalizing the findings on a larger population and different building types. Where differences are present, they tend to serve the idea that everyone is different. These differences do not disqualify the argument that there is a shared way in which designers behave in making decisions and in shaping their design solutions. In essence, we have seen that globally connected decisions have an executive nature in which they translate into major geometric and configurational transformations on the layout. In spite of apparent differences in designers' preferences, designers seem to share a certain way in segmenting their design solutions. This renders into certain associations between shapes and structures over the course of design, where both designers start from circulation and move to allocate occupational spaces following similar trends. These findings need to be backed up by introducing more robust settings and measures to the case study. Future studies should also consider more development on the strand of configurational transformations and their utilization in parametric shape grammar to associate the designed geometry with social meaning.

**Acknowledgments** I wish to thank Sean Hanna, Ruth Dalton, Christoph Hölscher for their support.

## References

1. Cross N (1984) Developments in design methodology. Wiley, Chichester
2. Stiny G (1980) Introduction to shape and shape grammars. Environ Plann B Plann Des 7(3):343–351
3. Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge
4. Ericsson KA, Simon HA (1993) Protocol analysis: verbal reports as data. MIT Press, Cambridge
5. Goldschmidt G (1990) Linkography: assessing design productivity. In: Trappl R (ed) Cyberbetics and system. World Scientific, Singapore, pp 291–298
6. Akin Ö, Lin C (1996) Design protocol data and novel design decisions. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, Chichester, pp 35–64
7. Hillier B (1996) Space is the machine. Cambridge University Press, Cambridge

8. Al Sayed K, Conroy Dalton R, Hölscher C (2010) Discursive design thinking: the role of explicit knowledge in creative architectural design reasoning. AIEDAM 24(Special Issue 2):211–230
9. Mitchell W (1990) The logic of architecture. MIT Press, London
10. Heitor T, Duarte JP, Marques Pinto R (2003) Combining grammars and space syntax: formulating, evaluating and generating designs. In: Proceedings of the fourth space syntax international symposium, London, pp 28.1–28.18
11. Oxman R (1997) Design by re-representation: a model of visual reasoning in design. In Akin O (ed) Design studies: special issue on prescriptive and descriptive models of design 18(4):329–347
12. Goel V (1995) Sketches of thought. MIT Press, Cambridge
13. Goldschmidt G (1992) Criteria for design evaluation: a process-oriented paradigm. In: Kalay YE (ed) Evaluating and predicting design performance. Wiley, New York, pp 67–79
14. Goldschmidt G (1995) The designer as a team of one. Des Stud 16(2):189–209
15. Bertel S, Vrachliotis G, Freksa C (2007) Aspect-oriented building design: toward computer-aided approaches to solving spatial constraints in architecture. In: Allen GL (ed) Applied spatial cognition: from research to cognitive technology, pp 75–102
16. Turner A (2006) UCL Depthmap: spatial network analysis software, version 6.0818b. University College London, VR Centre of the Built Environment, London

# Part VII
# Design Knowledge

# Generalized Design Knowledge and the Higher-Order Singular Value Decomposition

**Andy Dong and Somwrita Sarkar**

**Abstract** The question of what constitutes generalized design knowledge is central to design cognition. It is knowledge of a generic variety about products, the type of knowledge that is not principally related to any one product per se, but to knowledge about a broad class of products or a broad class of operations to produce products. In this paper, we use a complex systems perspective to propose a computational approach toward deriving generalized design knowledge from product and process representations. We present an algorithm that produces a representation of generalized design knowledge based on two-dimensional and multi-dimensional representations of designed objects and design processes, with the objects and processes being represented as a complex network of interactions. Our results show that the method can be used to infer and extract macroscopic, system level organizational information, or generalized design knowledge, from microscopic, primary or secondary representations of objects and process.

## Introduction

'Generalized design knowledge' is knowledge of a generic variety about products, the type of knowledge that is not principally related to any one designed object per se, but to the knowledge about a broad class of designed objects or a broad class of operations to produce those objects. Such knowledge is needed to achieve innovation [23]. The question that this paper raises is how this generalized knowledge can be efficiently calculated and represented from experience gained through interaction with design representations. Stated another way, what is the representation for this generalized design knowledge?

---

A. Dong (✉) · S. Sarkar
University of Sydney, Sydney, NSW, Australia
e-mail: andy.dong@sydney.edu.au

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_23,

James Perner's three-stage model for the cognitive development of representation skills is a useful starting point to address this question [19]. According to Perner, children progress through at least three stages in the development of cognitive skills associated with mental representation. Initially, children have the ability for *primary representations*, which have a semantic fidelity with the external world. Children can represent that an apple is an apple, but not that a computer named Apple is not the same as the apple that can be eaten. In the next stage, children develop the capacity for *secondary representation*. These are models of the referent object, but they may not necessarily have a direct semantic relation to referent object. At this stage, children can represent that an Apple computer is not the apple fruit, but could engage in pretend play to represent any object as an Apple computer or an apple fruit. Secondary representations are fundamental to the cognitive design skill of 'seeing as' because they allow us to entertain multiple models and interpretations of intermediate design representations. They allow us to have a model of what the world could be like, rather than as it is, which is essential to design. As Perner explains, "secondary representations are purposely detached or "decoupled" from reality. They are at the root of our ability to think of the past, the possible future, and even the non-existing, and to reason hypothetically." [19, p. 7].

However, one more stage of development in representational abilities is needed: the *tertiary* ability for *meta-representation*. Meta-representation is, literally, a representation of a representation. We start to understand that there is an ascribed relationship between the referent object and our representation or symbol for it, and we represent this relationship, which Perner names the 'representational relation'. Changing this relation is at the basis of using the same symbol to represent multiple objects and their respective semantic (mental) representations (or equivalently using different symbols to represent the same object). A symbol or the set of symbols in the representational relation can actually be an aggregate over multiple simpler perceptual or conceptual experiences.

As we represent the relation between the referent and its (mental) representation, we gain a capacity to produce alternative interpretations of the referent object by modifying the meta-representation. Stated in a more colloquial way, we can decide to change the way we conceive of an object. We can change our mental representation not by arbitrarily changing it per se, that is, by deciding that an apple is actually an orange, but by updating our representational relation, such that the mental representation now becomes consonant with the way that we represent the relation (e.g., as a fruit or as a computer). This skill allows us to compare alternative design concepts and is at the root of developing alternative representations [18].

Having the capability for meta-representation thus means having the representation of the concept of representation itself. It lies at the core of developing generalized design knowledge as represented in primary representations, because meta-representation entails constructing and having a representation for a model of previous and current design knowledge. The issue that this paper takes up is to develop a formalism to compute this generalized design knowledge. If a system has the capacity to infer or extract generalized design knowledge, then it should be able to group designed products and design processes into plausibly coherent categories. This paper will show how the singular value decomposition (SVD), and its more general form, the higher-order singular value decomposition (HOSVD), provides a formal way to produce generalized design knowledge as it is embodied in primary design representations.

While the literature on finding structure in data is both vast and deep in the machine learning literature, all approaches begin from either a heuristic about the expected clustering of the data or some pre-defined metric of similarity. More recent approaches to find structure from data start from a more principled cognitive science basis. For example, Kemp and Tenenbaum propose an algorithm which starts from a set of known grammars of knowledge structures, and then fits the structure to the data with the highest a priori probability [14]. Our aim is not to find the 'right' data structure for design representations, but rather to identify a way to make sense out of a set of design representations that may, on the surface, appear to be unrelated. Second, while we will not show it in this paper, our further aim is to produce plausibly correct structures, reflecting the observation that designers can identify multiple plausible concepts in design representations rather than a single a priori correct representation. In other words, we are aiming to produce suitable 'concepts' [22] from existing design representations. Design representations may contain both similarity relations, i.e., products that are similar to other products, and relational data, relationships between components in a product. The fundamental problem for designers is what knowledge from one domain can be applied in another domain or what exemplars [6, 27] provide broad knowledge frameworks for another domain [29].

## Modeling Design Representations as Matrices and Tensors

To develop a representation for computing generalized design knowledge from primary and secondary design representations, we will employ a complex networks perspective. Any complex system can be viewed as a complex network using a graph structure, where the nodes are the system elements and the links are structural, behavioral, functional, or any other type of interaction relationship between the nodes. A matrix, where the rows and columns represent nodes and the matrix entries represent these relations, can fully describe a graph. Thus, a graph and its corresponding matrix representation are equivalent representations of the same system.

A primary advantage of adopting this formalism is that it allows us to represent the entire system from the microscopic level (primary and secondary design relations) and simultaneously allows us to analyze, infer, and extract macroscopic level information at the system level (generalized design knowledge). The basic premise in complex systems studies is that interactions at the microscopic level often produce emergent macroscopic effects that cannot be inferred from an analysis of the microscopic relations. Since generalized design knowledge develops over time, over multiple design experiences, and over interactions with multiple designed objects, it will have this emergent quality of complex systems.

The essentials of the mathematical structure are as follows. A designed object, product, family of products, or a set of design processes is represented as matrices (or equivalently, graphs). Linear algebra algorithms, specifically singular value decomposition (SVD) and higher-order SVD (HOSVD), which are matrix decompositions that produce optimal lower-dimensional descriptions of higher order data, are then used to infer generalized design knowledge from the matrices of primary and secondary design information.

We will begin the discussion with an analysis of representations of designed objects rather than design processes, but the mathematical representations of both are entirely homologous. For simplicity, let us begin in a two-dimensional space wherein a designed object or a product has $n$ functions and $m$ behaviors associated with it. This is the type of representation used in the market analysis of products, wherein a product is represented simply as a bundle of attributes. It is a common representation in systems engineering [4], where a product, process, or organization can be represented as a bundle of functions that are fulfilled by performing certain behaviors, or a set of behaviors that are fulfilled by structural components. We can now represent a product as an $n \times m$ matrix $\boldsymbol{X}$ (or equivalently a bi-partite graph) where $n$ is the number of functions and $m$ is the number of behaviors. Thus, each entry $x_{ij} = 1$ if a product fulfills a function $i$, $i = 1$ to $n$ with behavior $j$, $j = 1$ to $m$. The addition of another dimension, the structural one, will produce an order 3 tensor that can represent, for one product, the entire Function-Behavior-Structure (FBS) framework [11]. The tensor element $x_{ijk} = 1$ will signify if a product has a structural component $k$ that takes a part in fulfilling function $i$ through behavior $j$. For now, we will concentrate on the 2D matrix representation, and develop the tensor representation later.

This is a simplified, idealized representation of a designed object, but nothing in this analysis hinges on whether this two dimensional representation is "true". Different designers can come up with different, equally valid matrices for the same object. They need not even be the same size, and different sized matrices, representing the object at different levels of granularity, may be developed. Indeed, this kind of parallel, equally valid, multiple representations lie at the basis of producing generalized design knowledge. The mathematics of analysis, therefore, must necessarily be general and not specifically tied to any particular or typical object.

Thus, our mathematical analysis is general and not tied to a specific model. A similar form of representation has previously been used to transform complex design optimization problems represented in symbolic equation form into a matrix format [21]. Although the matrix $\boldsymbol{X}$ is order 2 (the number of dimensions), the product representations form a vector space of dimension $nm$. This is because each row is a vector in $\Re^n$ and each column is a vector in $\Re^m$. Thus, the dimension of the vector space of all $n \times m$ matrices will be the number of matrices needed to define a basis for this space, which is $nm$.

For example, consider a product with 3 functions and 2 behaviors. Then we have a $3 \times 2$ matrix describing this product. Figure 1 shows the basis for all $3 \times 2$ matrices. Each matrix shows 1 function or 1 behavior. Now consider a product that has functions 1 and 2 but not 3, and behaviors 1 and 2 but not 3. Obviously, just 4 of the basis matrices, and not 6, can represent this product. Further, the resulting matrix will be $\boldsymbol{X} = [1\ 1\ 0;\ 1\ 1\ 0]$. Clearly, for both $n = 2$ and $m = 2$, a lower order, 2D, representation can capture the full design knowledge. Thus, we do not need all the 6 dimensions to describe the knowledge of the product. In many cases, a lower-dimensional representation will be sufficient to capture the full design knowledge. As we will see, a lower-dimensional representation is not only sufficient to capture generalized design knowledge, it is also necessary to bring out the hidden patterns and relationships that are latent in the representation, and are not explicit in the original representation. We now show that the singular value decomposition (SVD) generates an optimal lower order representation by generating an orthonormal basis for an $n \times m$ matrix.

It is well known that a vector space can be best represented by an orthonormal basis. An orthonormal basis is a set of linearly independent, mutually orthogonal vectors of unit length. Any other vector in that space can be expressed as a linear combination of the orthonormal basis. If we could find such an orthonormal basis, this can provide a model for $\boldsymbol{X}$, because the basis allows us to express all possible design representations as linear combinations of the basis vectors. Further, as we show later, we can also decide how many orthogonal vectors, i.e., how many dimensions, are necessary to optimally describe a specific product. There is such a method to find the orthonormal basis satisfying these requirements. For any arbitrary rectangular matrix, we are guaranteed to find the orthonormal basis using the singular value decomposition (SVD) [26].

The singular value decomposition is a matrix factorization technique that calculates a matrix $\boldsymbol{X}$ as a product of three other matrices, such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}$. By definition, $\boldsymbol{U}$ is an $n \times r$ matrix that is an orthonormal basis for $\Re^n$, and $\boldsymbol{V}$ is an $m \times r$ matrix that is an orthonormal basis for $\Re^m$ where $r$ is the rank of $\boldsymbol{X}$. The vectors of $\boldsymbol{U}$ are called the left singular vectors, the vectors of $\boldsymbol{V}$ are called the right singular vectors. The $r \times r$ diagonal matrix $\boldsymbol{S}$ contains the singular values that contain the scaling information on how a vector is transformed when it goes from $\Re^n$ to $\Re^m$ and are usually arranged in a decreasing order of magnitude. The number of singular values is equal to the rank $r$ of the matrix $\boldsymbol{X}$. Thus, SVD re-represents the original matrix in a diagonalized form, and produces a new derived independent set of basis vectors from the original information. Note that the vectors in the original $\boldsymbol{X}$ can now be represented as a linear combination of the vectors in $\boldsymbol{U}$ and $\boldsymbol{V}$, their components multiplied by the singular values in $\boldsymbol{S}$. Since the singular values are arranged in decreasing order, the first $k$ largest singular values are the most important in capturing information. For example, for our earlier example from Fig. 1, performing an SVD for $\boldsymbol{X} = [1\ 1\ 0;\ 1\ 1\ 0]$ will show that only two singular values and the corresponding singular vectors are needed to describe the original matrix in full.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

**Fig. 1** Basis space for a product with 3 attributes and 2 functions

The main outcome of SVD is that the relation between the designed object and its representation $\boldsymbol{X}$ can now be represented in an abstract form as vectors that are linear combinations of derived orthogonal bases and singular values. In summary, the orthonormal bases and the singular values provide a model for $\boldsymbol{X}$, that is, a meta-representation in Perner's model.

This concept scales to higher-order (order of $\boldsymbol{X} > 2$) product representations, since a representation of product knowledge with the FBS ontology [11] or with functional modeling (product, function, and flow) [24] would make $\boldsymbol{X}$ order 4 and 3, respectively. Consider our earlier discussion on representing, for example, each product as a matrix of functions and behaviors. If multiple products, each represented as a matrix, are "stacked" as "slices", then we have a 3D or 4D "cube" to represent a family of products in terms of their functions and behaviors.

Matrices with order greater than 2 are generally called tensors, and, following the accepted notation, tensors are represented with Euler capitals, e.g., $\mathcal{X}$. The decomposition of a tensor into derivational modes (equivalent to the left and right singular vectors in SVD) and a core mode that shows the level of interaction between the components in the derivational modes was first proposed by Tucker in order to handle multi-dimensional psychological data [28]. Tucker solved this composition for an order 3 tensor, and the solution was recently extended it to $N$ modes by De Lathauwer et al. [7], who showed that the Tucker decomposition is a generalization of the singular value decomposition. In so doing, they named the Tucker decomposition in $N$ modes the higher-order singular value decomposition (HOSVD). Like SVD, the HOSVD decomposes a tensor $\mathcal{X}$ into a core tensor $\mathcal{G}$ (equivalent to the matrix of singular values $\mathbf{S}$) and a set of matrices $\mathbf{A}$ (equivalent to the left and right singular vectors $\mathbf{U}$ and $\mathbf{V}$) along each mode of $\mathcal{X}$. For a tensor $\mathcal{X} \in \Re^{I_1 \times I_2 \times I_3 \ldots I_N}$ of order $N$, having $\Re^n$ elements along each mode, the HOSVD is given by [15]:

$$\mathcal{X} = \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 A^{(3)} \times_4 A^{(4)} \ldots \times_{\mathrm{N}} A^{(\mathrm{N})}$$

Each element of $\mathcal{X}$ is thus calculated as:

$$x_{i_1 i_2 i_3 \ldots i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} g_{r_1 r_2 r_3 \ldots r_N} a_{i_1 r_1}^{(1)} a_{i_2 r_2}^{(2)} a_{i_3 r_3}^{(3)} \ldots a_{i_N r_N}^{(N)}$$

$$\text{for } \mathrm{i}_n = 1, \ldots, I_n, n = 1, \ldots, N$$

Note that the calculation for each element of $\mathcal{X}$ depends upon each element of $\mathcal{G}$. Although $\mathcal{G}$ is not diagonal, as with $\boldsymbol{S}$, if we sort the Frobenius norms of $\mathcal{G}$ in descending order, it is possible to obtain a similar ordering of the singular values as in $\mathbf{S}$, which will become important later in interpreting the structure of generalized design knowledge.

The advantage of SVD and the HOSVD is their model for $\boldsymbol{X}$ or $\mathcal{X}$, respectively, permits the calculation of alternative representations of $\boldsymbol{X}$ or $\mathcal{X}$ through truncated forms of the singular vectors. In information retrieval [17] and the latent semantic approach [8], the truncated form is generally used to eliminate noise from the data, although in other work, the truncated form has also been shown to help in the reformulation of complex design optimization problems [21], relax constraints in spatial topology planning problems [9], and identify community structure in complex networks [20]. The truncated vectors play a different role in characterizing generalized knowledge. The claim is that the orthonormal vectors in the truncated approximation of the original higher-order representation embodies generalized knowledge, because these orthonormal vectors are the most appropriate bases for the vectors in the original representation. Since these original vectors encode knowledge about designs, the span of the truncated orthonormal basis vectors must represent sufficient generalized knowledge about the designs so that the original knowledge (design representation) could be reproduced. Mathematically stated, they must represent a sufficient approximation so that the original matrix $\boldsymbol{X}$ or tensor $\mathcal{X}$ can be recalculated as a linear combination of the truncated orthonormal basis vectors/derivational modes.

The calculation of the truncated versions of the original matrix $\mathbf{X}$ or tensor $\mathcal{X}$ retains the $k$ leading singular values of $\mathbf{X}$ or $\mathcal{X}$ as shown below:

$$\boldsymbol{X}_k = \boldsymbol{U}_k\boldsymbol{S}_k\boldsymbol{V}_k^{\mathrm{T}} = \boldsymbol{u}_1 s_1 \boldsymbol{v}_1^{\mathrm{T}} + \boldsymbol{u}_2 s_2 \boldsymbol{v}_2^{\mathrm{T}} + \ldots + \boldsymbol{u}_k s_k \boldsymbol{v}_k^{\mathrm{T}}$$

$$\mathcal{X}_{\mathrm{k}} = \mathcal{G}_{\mathrm{k}} \times_1 A_{\mathrm{k}}^{(1)} \times_2 A_{\mathrm{k}}^{(2)} \times_3 A_{\mathrm{k}}^{(3)} \times_4 A_{\mathrm{k}}^{(4)} \ldots \times_{\mathrm{N}} A_{\mathrm{k}}^{(\mathrm{N})}$$

In the case of the matrix, the approximation is guaranteed to be optimal in the least squares sense, but in the case of the tensor, it can be optimal but not necessarily unique [7, 16]. However, the lack of uniqueness does not alter the basic premise that the truncation provides a generalization of the original knowledge, albeit with a mathematically different orthogonal basis.

In this section, we will study the characteristics or ‘signatures’ of three possible forms of generalized design knowledge: perfectly modular knowledge, hierarchically modular knowledge, and random knowledge, that is, knowledge with no regular structure. The concept of modularity captures the notion that designers generate knowledge by establishing connections between knowledge rather than relying on pre-established and fixed semantic categories [5] and the dynamic situatedness of design cognition [13]. Each of these modules of knowledge brings together objects that share relevant knowledge. In a perfectly modular knowledge structure, what is known about one aspect of a design neither affects nor influences knowledge about another aspect of the design. For example, to know how the LCD screen of a cell phone works, I do not also need to know how the Li-ion battery works, and knowledge of these two aspects of the cell phone are essentially independent. In a hierarchically modular knowledge structure, what is known about one aspect of a design also entails knowing about another aspect; in other words, knowledge about one aspect of the design is subsumed by knowledge of another aspect. Returning to the example of the cell phone, knowledge about a touch screen requires knowledge of both capacitive sensing and liquid crystal display technology. In between a perfectly modular and a perfectly hierarchically modular knowledge structure is a knowledge structure that has some modularity but some overlap as well, which is what we would expect for most real-world designs. Finally, the situation of a random knowledge structure, which we are unlikely to find in real-world examples, implies no known underlying structure for a design. This is impossible, given that any design by definition is 'planned' and therefore has some organization in its knowledge structure.

Let us start then with the simplest example of generalized knowledge structure that is perfectly modular. For illustration purposes, we model this knowledge structure empirically as a 2D network have 64 nodes (for example, 64 functions and 64 behaviors) and 8 modules (8 functions and 8 behaviors cluster together tightly), in other words, 64 units of knowledge and 8 groups of highly inter-related knowledge units with strong intra-module ties to each other but no ties outside of its module. Such a network might also represent a mapping between structural design parameters and functional requirements. Similarly, we can construct an order-3 tensor consisting of 64 units of knowledge along each of the 3 modes, representing for example Gero's FBS ontology of function, structure and behavior [12], or a functional modeling paradigm of product, function and flow [24], or the representation introduced earlier in terms of "stacks" of products from a single family with common attributes and functions. The modularity of the knowledge structure is obtained by taking a truncated approximation of the original knowledge representation and then clustering the nearest neighbors [20]. The darker blocks (in red) show a high level of interaction between knowledge units whereas the lighter colored blocks (in blue) show a weak or no interaction between knowledge units. There will always be a diagonal line of dark (red) blocks since each knowledge unit is related to itself. By using a value of $k = 16$ in the truncation, Fig. 2a reveals the modularity for a 2D knowledge representation and Fig. 2c for 3D knowledge representation, for which the modularity in mode-1 and mode-2 of the tensor are identical and perfectly modular, and in mode-3 of the tensor, there is a regular knowledge structure, that is, partially modular.

We can note a few characteristics from the graphs. The first is that the number of non-zero singular values Fig. 2c for the perfectly modular knowledge structure of Fig. 2a is exactly the same as the number of modules. This holds true for the multidimensional knowledge structure of Fig. 2c for mode-1 and mode-2. In mode-3, the knowledge structure (not shown) is a single block of knowledge because there is only one significant, non-zero singular value, wherein each unit of knowledge is strongly related to all the other units of knowledge. This is reflected

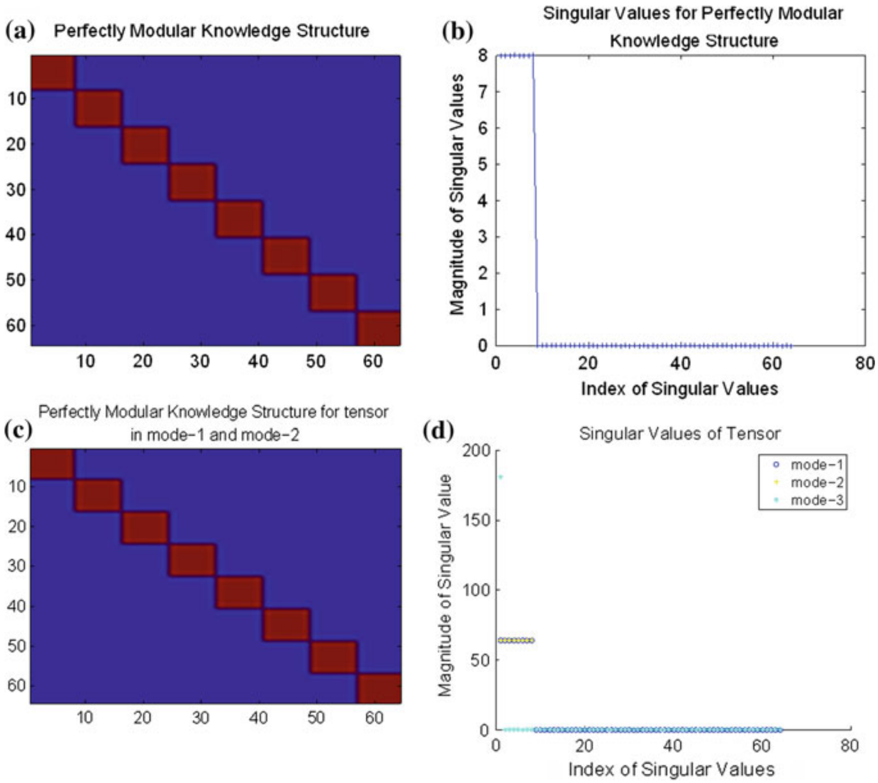


**Fig. 2** Characteristics of modular knowledge structure. **a** Perfectly modular knowledge structure. **b** Singular values for perfectly modular. **c** Perfectly modular knowledge structure for tensor. **d** Singular values of tensor

in Fig. 2d, for which the number of significant non-zero singular values in mode-3 is 1; there is a long trail of singular values of magnitude less than $10^{-12}$. In other words, the number of significant, non-zero singular values gives an indication as to the number of modular knowledge units in the knowledge structure and the degree of inter-relatedness of the knowledge.

We now turn to the case of a perfectly hierarchical modular knowledge structure. To generate the hierarchical modular structure, we first start with a perfectly modular structure and then introduce perturbations in this matrix using a parametric probability $p$ of "rewiring", that is, introduce or remove a relation between two knowledge units. Multiple levels of rewiring, with a decreasing value of $p$ at each stage of rewiring produce a matrix with a hierarchically modular structure. Figure 3a and Fig. 3c show the hierarchical knowledge structure for a two-dimensional and three-dimensional knowledge structure, and Fig. 3b and Fig. 3e show the corresponding singular values. The signature of the singular values differs from the perfectly modular case; rather than a single discrete discontinuity between the significant non-zero singular values and the zero magnitude

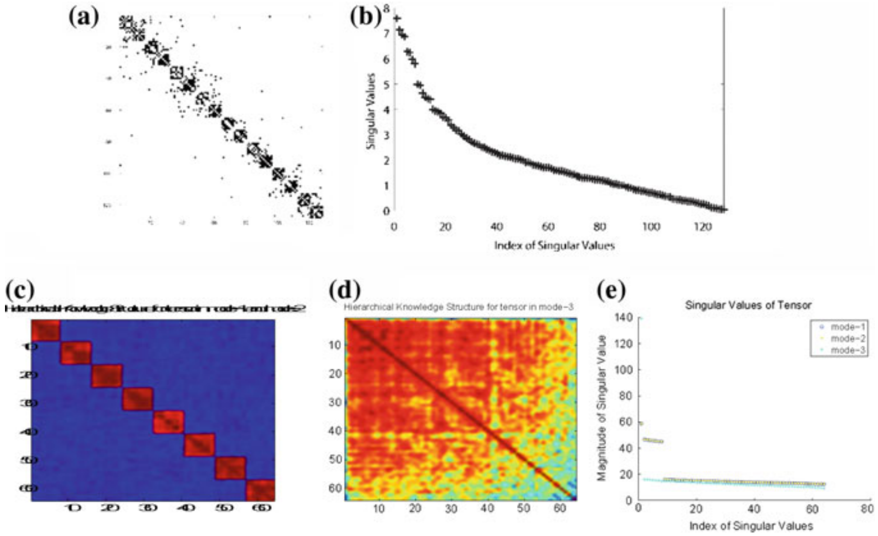


**Fig. 3** Characteristics of hierarchical knowledge structure

values, there is a series of discrete steps. The number of these discrete steps corresponds to the number of hierarchical levels, that is, the number of chunks of knowledge subsumed within larger chunks of knowledge. Figure 3b shows the SVD signature of this knowledge structure as having distinct gaps between the 2nd and 3rd, 4th and 5th, and 8th and 9th singular values, corresponding to three distinct hierarchical levels.

Finally, we turn to the situation of the random knowledge structure. A random knowledge structure is generated by connecting an edge between any two knowledge units if a randomly generated probability is greater than or equal to a probability *p*. There is neither a perceivable structure in the knowledge structure for the 2D case of Fig. 4a nor mode-1 of the 3D case of Fig. 4b, which is equivalent to mode-2 and mode-3 (not shown). Corresponding to this, the singular values follow a trend wherein there is a single significant non-zero singular value followed by a 'long tail' of non-zero singular values. The magnitude of the first singular value is correlated to the degree of randomness; as *p* increases, the magnitude of the first singular value also increases, as shown in Fig. 4d.

The preceding empirical results thus provide a way to characterize the underlying knowledge structure as being modular, hierarchically modular, or random. In particular, there are several characteristics that are particularly useful in identifying the type of knowledge structure, based on the singular value spectrum:

1. If the singular value spectrum shows discrete steps between clusters of singular values, and if the singular values are very similar within the cluster, then the knowledge structure is likely to be modular.

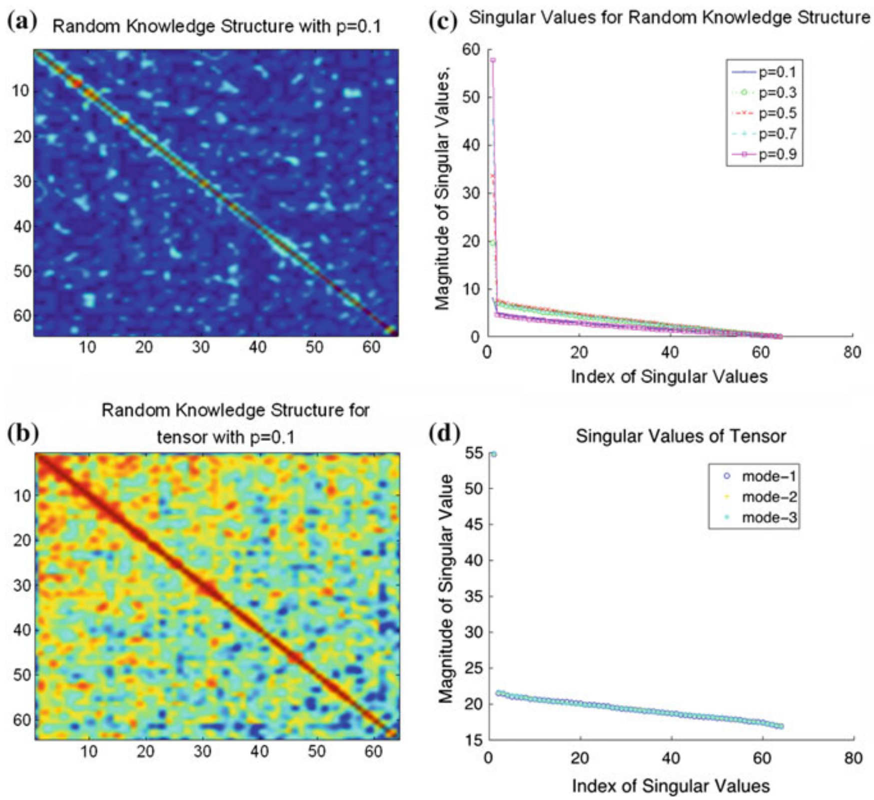


**Fig. 4** Characteristics of random knowledge structure

2. If the singular value spectrum shows discrete steps between clusters of singular values, and if the singular values vary linearly within the cluster, then the knowledge structure is likely to be hierarchically modular.
3. The number of significant singular values in a modular or hierarchically modular knowledge structure is related to the number of knowledge modules.
4. If the singular value spectrum is continuous with a single dominant non-zero singular value, then the knowledge structure is likely to be random.

With the above qualitative characteristics, we now show the underlying generalized knowledge structure of some real-world systems.

## Knowledge Structure of Design Processes

In this section, we analyze the modularity structure of design processes. Though the preceding representations have been shown for representing products, the same matrix or graph representation is easily extended to the analysis of design tasks or processes. In our analysis we consider four product development datasets [2, 3] dealing with vehicle development, operating system software development, pharmaceutical facility development, and a 16-storey hospital development. The authors showed that these networks had a high degree of clustering, a finding that led them to conclude that the network structure of design and problem solving processes is modular in nature. We have obtained all datasets from the New England Complex Systems Institute [1]. In this data, a product development (PD) network is a directed graph with $N$ nodes and $E$ edges, where there is an edge from task $v_i$ to task $v_j$ when $v_i$ feeds information to $v_j$. The 4 datasets are shown in Fig. 5. Each of these networks record design process model diagrams, module-subsystem type design dependencies, structured interviews with designers and engineers in response to questions such as "from which tasks do the inputs to a particular design task come from and to what tasks does it feed?", and various design documents. For example, in the pharmaceutical facility and hospital design projects, examples of tasks can be computations of grid layouts and computations of final pile layouts and spacings based on column positions [2, 3].

Using the SVD based method we propose here, we show that as the size of PD networks scale up, along with product size, their modularity *does not* scale up. We show that large size modules are relatively absent in very large PD networks, although they have many small sized modules. This is counterintuitive because the architecture of the product itself that is being designed using the corresponding process structure could be modular, with large sized modules present at the system level. It also brings out some important constraints that can govern the formation of knowledge structures in design processes, especially ones wherein time plays a critical role.

Figure 5 shows the detailed results for the largest dataset, the 16-storey hospital design project, with 889 tasks. The observations we present on this case, however, are also all valid for the other 3 datasets. First, these matrices are very sparse. The denser the matrix, the more the interaction time spent on design tasks. Thus, the reason for the sparseness is the objective of capturing the most amount of knowledge interaction in the least possible amount of tasks or activities.

Second, note that the singular values plot resembles a hierarchically modular network, with the singular values clustered in groups and large inter-group gaps, but the hierarchies and modules show a much smoother qualitative form as compared to the perfect hierarchical modular form presented earlier. While the first two singular values sit sharply separated from the others, there is a gradual reduction in the values for the others, with successive singular values just a little smaller than the previous ones. This characteristic is much more similar to a regular graph, which describes activities just next to each other as being connected the most, and the farther two activities are from each other, the lower the likelihood of connection between them.

Third, note that the rank of the matrix is 606, much less than the original size 889. This shows a large amount of redundancy in the data. At $k = 606$, the full information is captured. Finally, note the reduced dimension representations,

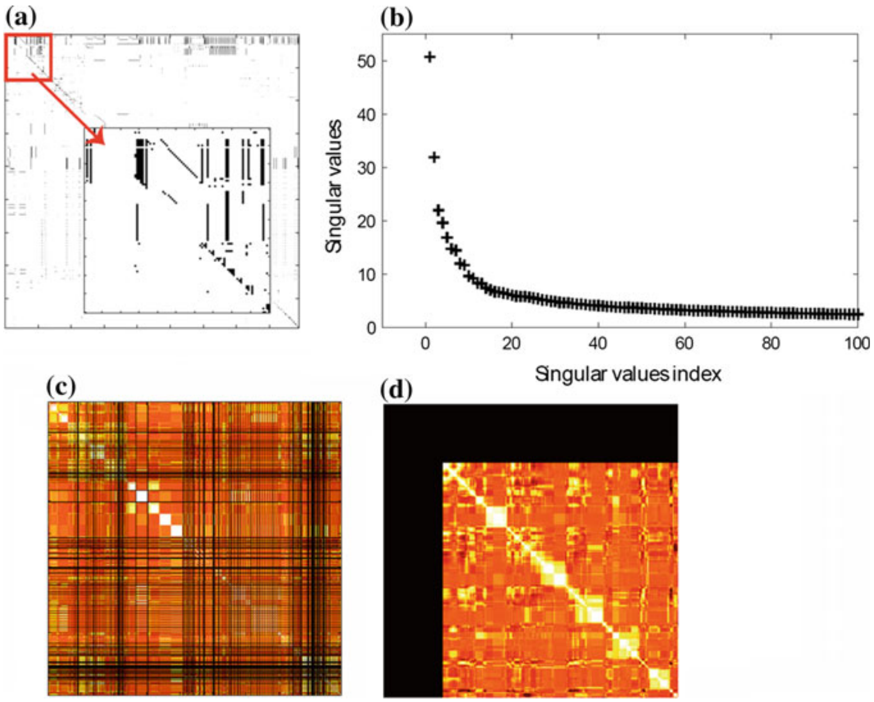


**Fig. 5** SVD analysis for 16-story hospital design task knowledge matrix: (**a**) adjacency matrix, 889; a blow up of the *top left* hand corner of the matrix is shown for clarity (**b**) Singular values plot; (**c**) $k = 16$ lower-dimensional representation; (**d**) reordered $k = 16$ representation, showing modules

Fig. 5c–d. We have preserved the first 16 singular values and vectors, and we analyzed the modularity. It is immediately obvious that there are no large sized modules, and that all modules are much smaller as compared to the full matrix size.

These characteristics suggest that product development/design process networks differ from other social, technological and information networks of human interaction. In social networks, for example, it is well known that the module sizes scale with the network size, to an upper limit [10]. It is clearly visible that from a cognitive perspective, design and problem solving, or indeed any other social goal directed set of activities between humans involving exchange of information, have a distinctly different knowledge structure than networks of informal social interaction.

In the $k = 16$ representation, a large part of the reordered network is black, or blank, signifying that a certain set of activities have no or negligible relationship with other tasks or activities. This shows that in product development networks, a major motivation is to have tasks and activities that are self-contained—their completion does not depend on interaction with many other activities, and hence they will play a significant role in speeding up the completion of the project and play an exemplary role in knowledge modularity. These tasks can be said to be perfectly modular—a single task with defined knowledge that can be completed without knowledge input from any other task.

Based on above observations, we present two hypotheses. Our first hypothesis is that the presence of knowledge or task modularity in process networks is essential to ensure that sets of related tasks being completed need not be done or visited again. For example, the set of tasks involved in laying the foundation of a hospital building must be completely finished before the set of tasks dealing with constructing the higher storeys. This is the "module formation" hypothesis. Our second hypothesis is that large sized knowledge structures in process networks are absent or relatively rare, because they increase the iteration time to task completion within a single module. For example, in a software development module, the larger the size of the module, the more number of iterations needed to finalize it, and the lower its reusability as a generic component. However, given that the authors in the original paper extracted this information from personal interviews and design documentation, we could reasonably conjecture that the practitioners conceptualized their general knowledge of tasks in small chunks of constant size regardless of the size of the overall task. This constant size chunking may be caused by limited short-term memory when being asked to recall relations between tasks. This is the "module size limiting" hypothesis. In other words, we show that product development knowledge structures have the unique form they do because of two opposite formational forces—one forcing the formation of modules, the other putting constraints on the size of modules.

Finally, we note that the modules found in knowledge process networks at various hierarchical levels represent sets of design/management processes that are tightly coupled, and therefore, could be used to inform strategic knowledge in a design domain.

## Knowledge Structure of Designed Objects (Products)

In the following example, a set of products was modeled using the functional basis [24]. The aim of this analysis is to compare the generalized knowledge structure identified by the HOSVD-based analysis and the common products and dominant functions and flows identified by Stone and colleagues. The tensor describing the products has a dimensionality of $i = 70$ products, $j = 34$ functions, and $k = 20$ flows. A non-zero entry in the tensor means that product $i$ implements function $j$ on flow $k$. Consistent with the original data, entries in the tensor represent customer importance of the function-flow for the given product, and ranges from 0 (not important/not implemented) to 22. Where a product implements a function on a combination of flows, e.g., "*mix* liquid and solids" and "*mix* solids", then the importance values were summed. Flows that did not appear in the data set even though they occur in the functional basis, such as "human motion" and "particle velocity", were ignored. Sub-flows that did not appear in the data set were ignored and only the higher-level flow was included, e.g., “energy/electromagnetic /solar” but not “energy/electromagnetic/solar/intensity”. Some minor changes were made to the data set, such as setting the function-flow for the Krups Café Trio to “measure hydraulic pressure”, since the device pressurizes steam to froth the milk, and the Paint Sprayer to “measure pneumatic pressure”, since the device operates on pressurized air to spray the paint out the nozzle. Both devices were recorded in the data set as “measure pressure”, which was ambiguous. In the data set provided, the verb “dissipate” was used as a function. However, this verb does not appear in the functional basis vocabulary. The function dissipate was added to the category “Control Magnitude/Change” since energy dissipation involves controlling the loss of energy in an orderly manner. Figure 6a–c show the non-truncated knowledge structure for the data set, in which reveals no apparent knowledge structure.

To reveal the underlying knowledge structure, a truncated form of the original knowledge structure is generated by retaining the leading singular values in each mode of $k_1 = 12$, $k_2 = 12$, and $k_3 = 13$. These values were empirically obtained by selecting the index of the singular values when the magnitude of the singular value dropped below 20, at which the magnitudes for the singular value for each mode crossed over and dropped significantly. The knowledge structures were then generated, and shown in Fig. 7a–c, for product, function, and flow, respectively.

Some common knowledge structures, i.e., products that appear in the same family of products, dominant functions, or dominant flows, which are consistent with the product module architectures identified by Stone and colleagues [25], include:

- Coffee-maker family of products, appearing as a small module in the middle of Fig. 7a, consisting of the items {‘West Bend Iced Tea Maker-RBS’, ‘Mr. Coffee Coffee Maker-RBS’, ‘KRUPS Café Trio’, ‘Mr. Coffee Iced Tea Maker-RBS’}
- Functional module associated with providing information, consisting of {‘Signal/Indicate’, ‘Signal/Measure, ‘Signal/Display’}
- Human hand flows, consisting of the flows {‘Material/Human’, ‘Energy/Human/Force’}, which are the only human flows appearing with more than one function. The human flow of Human energy alone appears only with the function Import.

In addition, there are overlaps in knowledge structure, implying the ability to transfer the knowledge. For example, the following products share an overlap with at two other product modules: {‘B&D Weed Trimmer’, ‘Durabuilt Hand Vacuum’, ‘Wet/Dry Vacuum’, ‘SKIL Power Screwdriver-RBS’}, which means that the ‘electricity to torque’ module of the SKIL Power Screwdriver and the ‘electricity to pneumatic energy’ of the vacuums could be shared with the Hamilton Electric Mixer and the Pneumatic Air Ratchet. An important overlap identified by Stone and colleagues are the modules supplying electricity and driving “resulting in a mix modularity tool” [24, p. 17], which is explicitly identified in the module {‘Connect/Mix’, ‘Provide/Store’, ‘Convert’}. This modular overlaps with the

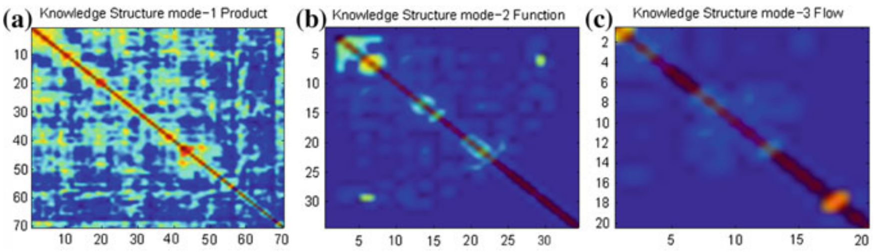


**Fig. 6** Non-truncated knowledge structure for products in functional basis

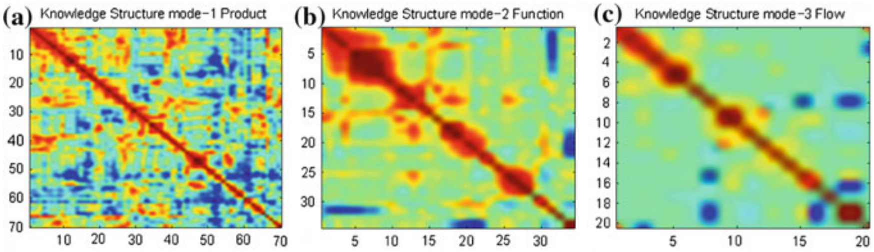


**Fig. 7** Knowledge structure for products in the functional basis

module {‘Provide/Store’, ‘Convert’, ‘Control Magnitude/Change/Condition’, ‘Guide/Translate’, ‘Control Magnitude/Change/Form} and is hierarchically embedded within the larger module {‘Separate’, ‘Connect/Couple’, ‘Transfer/Transport’, ‘Guide/Allow DOF’, ‘Support /Stop’, ‘Refine’, ‘Distribute’, ‘Connect/Mix’, ‘Provide/Store’, ‘Convert’, ‘Control Magnitude/Change/Condition’}.

## Conclusions

This paper proposed a way to describe the underlying generalized knowledge structure of designed objects and design processes. The approach is valid for two-dimensional and multi-dimensional product and process representations. The purpose is to find an approach to obtain broad knowledge frameworks for a set of design representations that may not appear to share any a priori similarity.

The results on synthetic and empirical data sets demonstrate the feasibility of the approach in generating plausible knowledge structures. As the results suggest, knowledge structure discovery over a broad base of design representations provides a way of acquiring experience-driven constraints on the structure of design concepts. Depending upon the type of experiences (or design representations) presented, different concepts can emerge across explicit and implicit relations, and with each subsequent experience having the potential to displace prior concepts.

Different from machine learning systems used to find patterns in design representations based on classifying pre-specified properties, the approach operates on the premise of finding structure in any representation of similarity *or* relations between design parameters or functional requirements of designed products or design processes. Finding these broad knowledge frameworks from experience is crucial for transferring design knowledge across domains, wherein explicit relations are rarely clear or defined, but often discovered.

## References

1. Braha D, Bar-Yam Y (2000) Data for the topology of large-scale engineering problem-solving networks, Available online http://necsi.edu/projects/braha/largescaleengineering.html. Last accessed 02 Jan 2011
2. Braha D, Bar-Yam Y (2004) Topology of large-scale engineering problem-solving networks. Phys Rev E 69(1):016113
3. Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. Manage Sci 53(7):1127–1145
4. Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: a review and new directions. Eng Manag IEEE Trans 48(3):292–306
5. Coyne RD, Newton S, Sudweeks F (1993) A connectionist view of creative design reasoning. In: Gero JS, Maher ML (eds) Modeling creativity and knowledge-based creative design. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, pp 177–209
6. Darke J (1979) The primary generator and the design process. Des Stud 1(1):36–44
7. de Lathauwer L, de Moor B, Vandewalle J (2000) A multilinear singular value decomposition. SIAM J Matrix Anal Appl 21(4):1253–1278
8. Dong A (2005) The latent semantic approach to studying design team communication. Des Stud 26(5):445–461
9. Dong A, Sarkar S (2011) Unfixing design fixation: from cause to computer simulation. J Creative Behav 45(2):147–159
10. Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci 104(1):36–41
11. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36
12. Gero JS (1998) Concept formation in design. Knowl Based Syst 11(7–8):429–435
13. Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Des Stud 25(4):373–391
14. Kemp C, Tenenbaum JB (2008) The discovery of structural form. Proc Natl Acad Sci 105(31):10687–10692
15. Kolda T (2009) Tensor decompositions and applications. SIAM Rev 51(3):455
16. Kroonenberg P, de Leeuw J (1980) Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika 45(1):69–97
17. Landauer TK, Foltz PW, Laham D (1998) Introduction to latent semantic analysis. Discourse Process 25:259–284
18. Leslie AM (1987) Pretense and representation: The origins of “Theory of Mind”. Psychol Rev 94(4):412–426
19. Perner J (1991) Understanding the representational mind. The MIT Press, Cambridge
20. Sarkar S, Dong A (2011) Community detection in graphs using singular value decomposition. Phys Rev E 83(4). doi:10.1115/1.3179148

21. Sarkar S, Dong A, Gero JS (2009) Design optimization problem (re)-formulation using singular value decomposition, J Mech Des 131(8). doi:10.1115/1.3179148
22. Schön DA (1963) Displacement of concepts. Tavistock Publications, London
23. Sinclair G, Klepper S, Cohen W (2000) What's experience got to do with it? Sources of cost reduction in a large specialty chemicals producer. Manag Sci 46(1):28–45
24. Stone RB, Wood KL (2000) Development of a functional basis for design. J Mech Des 122(4):359–370
25. Stone RB, Wood KL, Crawford RH (2000) A heuristic method for identifying modules for product architectures. Des Stud 21(1):5–31
26. Strang G (1988) Linear algebra and its applications. Harcourt, Brace, Jovanovich, Publishers, San Diego
27. Summers JD, Bettig B, Shah JJ (2004) The design exemplar: a new data structure for embodiment design automation. J Mech Des 126(5):775–787
28. Tucker LR (1964) The extension of factor analysis to three-dimensional matrices. In: Frederiksen N, Gulliksen H (eds), Contributions to mathematical psychology, Holt, Rinehart and Winston, Inc., New York, pp 110–127
29. Ward TB (1994) Structured imagination: the role of category structure in exemplar generation. Cogn Psychol 27(1):1–40

# Reformulating CK Theory with an Action Logic

**Filippo A. Salustri**

**Abstract** CK theory is an interesting and unique theory of engineering design. This paper introduces *ALX3d*, a formal descriptive version of CK based on the action logic ALX3, which is able to represent aspects of the actions, preferences, beliefs, and knowledge of collaborating, imperfect agents (such as human designers). It is shown that all the basic notions of CK can be rendered in the logic of ALX3d with only one relatively minor change in how the CK terms *concept* and *knowledge* are defined and related. A case study of CK is used to show how ALX3d can also be used to describe some "real-world" situations. The advantages of ALX3d are that they recast CK in a form more readily understood by those accustomed to expert, knowledge-based, and formal systems; provide a "scientific" vehicle for reasoning about the design activities it can describe; and define a possible basis for the development of new, computer-based designers' aids.

## Introduction

*CK theory* [1, 2] presents an interesting and unique theory of engineering design, but the available literature does not cast CK in a sound logic. Without soundness, there are severe limits to the reliability of results that any "formal" theory can obtain. In this paper, the author will demonstrate that a *sound action logic* can be just as expressive as CK using only conventional notions of logic. That is, the spirit and benefits of CK can be preserved using conventional logic. The author has

---

F. A. Salustri (✉)
Ryerson University, Toronto, ON, Canada
e-mail: salustri@ryerson.ca

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_24,

presented previous related work elsewhere [3]. Having a formal descriptive (not prescriptive) reformulation of CK yields new benefits that will be described.

CK is a high level description of designing based on the intuitive distinction between *concepts* and *knowledge*. Knowledge, in CK, consists of statements that are true. Concepts, on the other hand, are statements the logical statuses of which are *unknown*. They are groups of propositions that together describe a possible design solution. Designing, in CK, proceeds from a base of knowledge, and from which concepts are developed. These concepts are manipulated and eventually verified to become knowledge. Thus, a proposed design is a CK concept, and once the design has been appropriately verified, it becomes *known* as a suitable design solution. CK partitions the domain of all propositions into two spaces; *C-space* contains concepts and *K-space* contains knowledge. *Operators* are used to represent the kinds of activities undertaken by designers that transform propositions. These operators are broadly categorized by whether their inputs and outputs are concepts or knowledge. This leads to four categories of operators.

C→C operators transform one concept into another.

C→K operators transform a concept into knowledge through some validation action that establishes a logical status for the concept.

K→C operators generate concepts by transforming knowledge propositions. For example, given knowledge items $x$ and $y$, but no further knowledge about $x$ and $y$, one might create a concept consisting of $x \wedge y$, or $x \Rightarrow y$. These operators can be thought of as design concept generators.

K→K operators deduce new knowledge from existent knowledge; these are the conventional acts of inference common in analysis.

There have been many papers written about CK, and case studies indicate that CK can describe a variety of design activities. However, as they are all based on a few fundamental ideas, this author uses only those key papers that define CK. Further details about CK will be provided below.

Action logics are formal systems that address the activities undertaken by reasoning agents. ALX3 [4] is especially well suited because it is the only sound and complete action logic of which the author is aware that assumes the agents (i.e. human designers) exhibit *bounded rationality*—they are imperfect reasoners having imperfect/incomplete knowledge (per Simon [5]). CK also assumes bounded rationality.

The rest of this paper is organized as follows. Section "Overview of ALX3" introduces ALX3. Section "Applying ALX3 to Design: ALX3d" then describes how ALX3 covers the scope of CK theory. The author does this by examining each key feature and notion of CK from [1] informally, and then translating it into a formal representation in ALX3, resolving any identified logical problems along the way. We refer to the formal design model thus developed as *ALX3d* (ALX3 for *d*esign). Section "An example" then uses an example of the use of CK from the existent literature to demonstrate the use of ALX3d. Finally, section "Potential Benefits of ALX3d" discusses some of the benefits that a formalization like ALX3d can afford.

## Overview of ALX3

ALX3 is a sound and complete first-order action logic that incorporates knowledge, belief, preference, and action operators to represent the activities of multiple agents working with bounded rationality (per Simon [5]). ALX3 is completely documented in [4]. It assumes the usual apparatus of first order logic: constants, variables, functions, and relations, conjunction ($\ni$), disjunction ((), negation (¬), material implication ($\Rightarrow$), and universal ( ) and existential (□) quantification over variables. We also use the notation *x: y* to indicate that *x* is defined as *y*.

In action logics, an *agent* is an entity that takes actions to achieve certain states. An agent *a* can know ($K_a\psi$) or not know ($\neg K_a\psi$) a proposition $\psi$; the agent may also believe ($B_a\psi$) or not believe ($\neg B_a\psi$) the proposition. ALX3 defines knowledge typically for formal systems as *true, justified belief.* Within the system, knowledge and belief are treated as two separate operators related by a definition of the former with respect to the latter. Other definitions of knowledge are possible without necessarily affecting the soundness of the logic. In the language of ALX3, $K_a\psi$ : $B_a\psi \ni \psi$. That is, for a proposition $\psi$, knowledge is equivalent to true, justified belief. Consider the statement "*It is raining in Tokyo.*" The statement is either true or false, whether or not an agent knows it. The agent may *believe* the statement is true, but will not *know* until the agent takes appropriate actions to verify the belief.

One might question our definition of knowledge with regards to determining truth. Philosophically, a belief is true if it is true in an absolute sense, which means that all we have are beliefs. We would have to have access to some omniscient agent to guarantee that true things are in fact true. However, as engineers and scientists, we are accustomed to assuming truth by invoking other mechanisms. We accept the yield stress of cold-rolled steel as given in standard handbooks, for example, even though the actual yield stress of any specific steel part will not be exactly as the handbook defines. Similarly, we carry out physical testing of products to determine their performance knowing that some products will likely fail in the future—yet we accept the product's performance as true based on those tests. This notion of truth, though philosophically unsatisfying, is sufficient for us to have developed all the engineered products that have ever existed, and with a very high degree of overall success.

Generally, we assume truth within and relative to some specific *context*; so long as the context holds, we can expect a certain measure of truth to be acceptable. This is a typical scientific approach: given evidence that some statement is true, and until incontrovertible evidence is found to the contrary, then accepting the truth of the statement is reasonable.

There is another important difference between knowledge and belief within the convention of formal systems. Knowledge is not "lost"—once you have it, you cannot lose it. Losing knowledge is not the same as just "forgetting" it. Forgetfulness is a function of an imperfect reasoning agent (i.e. an agent with bounded rationality) and not a function of the knowledge that is forgotten. Beliefs, on the other hand, can be *retracted* from a system of beliefs if some action proves that the belief cannot be true. This kind of action is the same as that which, when successful, identifies truth and thus knowledge. In both the author's work and in CK, these kinds of actions are *validation* actions that engineers use to "prove" their concepts—within a reasonable context. This makes beliefs similar to assumptions —statements that we may take to be true for the sake of achieving some goal unless and until they are demonstrated to be false.

ALX3 also uses a *many-worlds interpretation* based on action logic semantics. This means that the current state (or "world") is a collection of propositions, and that alternative (or future, or past, or just *other*) states are accessible from the current state through *actions* that cause propositions to be added or withdrawn. Thus, actions allow one to represent how states of knowledge and belief, and the propositions they include, change. Actions are written in ALX3 as $\langle \mathrm{a} \rangle \psi$, where *a* is the action and $\psi$ is a proposition that is true in any state that can occur because of executing the action. In other words, $\langle \mathrm{a} \rangle \psi$ means that executing action *a* will make $\psi$ true.

It is important to note that the propositions that define a state need not be written in ALX3. State-defining propositions must obey the rules of first order logic in order to maintain the soundness of ALX3, but otherwise, the language of the state propositions may differ. This is especially important with regards to the primitives that are used in those propositions. Any logic built on first order systems is acceptable. Since these propositions capture, among other things, the product being designed—a *product model*—this means that any first order product modeling logic can be used with ALX3 without affecting the soundness of ALX3 itself. This gives us the flexibility to treat the product modeling logic and the design process logic as essentially disjoint entities.

Associated with ALX3 actions is the notion of *accessibility* of states. A state *t* is accessible from another state *s* if there exists an action or a sequence of actions that can be described by propositions that are true at *t*, *s*, and all intermediate states. There are two accessibility relations in ALX3. Direct accessibility (DA) is defined as $\mathrm{DA_i}\psi$: $\exists \mathrm{a}\ \langle \mathrm{a_i} \rangle \psi$; that is, agent *i* can reach a state where $\psi$ holds directly from the current state by executing action *a*. General accessibility (A) is defined as $\mathrm{A_i}\psi :\ \mathrm{DA_i}\psi((\mathrm{DA_i}\varphi \ni (\varphi 4 \psi))$; that is, agent *i* can reach a state where $\psi$ holds either directly through one action or indirectly through a sequence of actions (the causation operator 4 is explained next).

A key feature of ALX3 is a special implication operator for causal relations. Here we write it $\varphi 4 \psi$; that is, in all states (a) that are *closest* to current state and (b) where $\varphi$ holds, $\psi$ also holds. A *closest state* is one that (a) is accessible from current state (i.e. has an action allowing an agent to move to it from the current state) and (b) has the smallest possible changes from the current state. The semantics of ALX3 [4] provides a complete formal description of this operator, but that is beyond the scope of this paper. We note that the ALX3 causal operator is broader than the usual "scientific" sense of "cause and effect." In ALX3, cause can arise from simple preference (see below—i.e. something is caused because an agent prefers it to an alternative). It can also capture the relation between dependent and independent variables—i.e. the values of a set of independent variables *cause* the values of a set of dependent variables. As such, a causal relation in ALX3 can also be regarded as a kind of *explanation*.

Finally, agents may sometimes choose to *prefer* one state to another without necessary explanation. ALX3 supports this with a binary *preference operator*. We write this as $\psi P_a \varphi$; that is, agent *a* prefers closest states where $\psi$ is true to closest states where $\varphi$ is true. This is useful because it decouples what agents prefer from the rationale for that preference.

## Applying ALX3 to Design: ALX3d

In this section, we will explain how ALX3 can be used to represent CK theory. We will consider key concepts in CK one at a time, and show the corresponding element in ALX3d. No new operators are needed for this beyond those already in ALX3. This means that ALX3d is as sound a logic as ALX3.

### *Knowing Versus Believing*

The fundamental departure in ALX3d from CK is the notion and application of *logical status*. In CK, only knowledge has logical status; concepts are not knowledge and have no logical status.

In ALX3d, on the other hand, we distinguish between the logical status of a proposition and whether an agent knows that logical status. That is, we use *belief* (a proposition that has a logical status not *known* to an agent as described in the previous section) as the comparable notion. That is to say, an agent can believe a proposition and work with it even if it is false. While this definition of belief is not as subtle as its formal definition ALX3, it is sufficient for our purposes here.

We therefore make the following equivalences. CK's knowledge space (K-space) contains all true propositions [1]; we interpret this as saying that K-space contains all *known* propositions ($K_a \psi$). Similarly, CK's concept space (C-space) contains all propositions "that have no logical status" [1]; we interpret this as meaning that C-space contains propositions the logical statuses of which are not known to an agent, but to which an agent ascribes a logical status temporarily (until validation is possible) to attain some goal. That is, we are saying that C-space contains only *believed* propositions ($B_a \varphi$), and that ALX3 beliefs subsume CK concepts.

## Initial and Final Conditions

In CK as in ALX3d, a design is complete when its description consists entirely of knowledge (and no beliefs), $K_aD$ (where *D* is the design being developed). This is the final condition. We know the design is sufficient because we know it satisfies a conjunction of requirements *R*; that is, the agent knows that the requirements R imply at least one design D,

$$K_a(R \Rightarrow D). \tag{1}$$

Initially, in ALX3d, the designers know some of the requirements, $K_aR_i$, and may have some beliefs about the design, $B_aD_i$. Since beliefs may be falsified, there is no reason to undertake a design. Thus at least one requirement must be known (i.e. must be true). They would also believe that there are actions they can take that will lead to a state in which some superset of the initial requirements $R_i$ (i.e. R) some design D in that state. We summarize this by saying that the goal of designing is to move from a state of *belief* to one of *knowledge*. This is consistent with CK, where design involves the transformation of concepts into knowledge. In ALX3d, we write this as:

$$K_aR_i \wedge B_aD_i \wedge B_aA_a(\exists R\ \exists D\ ((R \Rightarrow Ri) \wedge (R \Rightarrow D))). \tag{2}$$

## Coarse Description of Design Activity

In CK, K-space and C-space include propositions that capture design information. In ALX3d we impose a little more detail. The requirements, *R*, of a design problem is a conjunction of individual requirement propositions, $R : \ni_i r_i$ . *R* can be a conjunction of different *r*'s at each different state. Similarly, a design *D* is a conjunction of individual design propositions, $D\ : \ni_i d_i$ . *D* can be a conjunction of different *d*'s at each different state.

We identify an appropriate design by going from $B_a(R \Rightarrow D)$ to $K_a(R \Rightarrow D)$. This is done with a validation action by which *D* is evaluated with respect to *R*. (Validation actions are described below.) These actions are identical to C → K operators in CK. Once the designers know that $R \Rightarrow D$, then $K_a(R \Rightarrow D) \Rightarrow K_aD$ (because the knowledge operator in ALX3 distributes over implication) and they know the design too. That is, we know the design *D* is "right" because we have validated it against *R*.

We use causal implication in ALX3 to define the following:

$$K_aR \ni B_aD\, 4\, K_a(R \Rightarrow D)(K_a(\neg(R \Rightarrow D))(\neg K_a(R \Rightarrow D))). \tag{3}$$

This says that if the designers were to *know* the requirements and *believe* to have a valid design, then one of three conditions holds.

$K_a(R \Rightarrow D)$—The validation was successful and we now know that *D* satisfies *R*. We have found a solution.

$K_a(\neg(R \Rightarrow D))$—The validation failed; *D* does not satisfy *R*. *D* and/or *R* will have to be changed to proceed.

$\neg K_a(R \Rightarrow D)$—The validation could not be performed or completed. This would be the case if either *R* or *D* were lacking in sufficient detail.

We will examine these three conditions in further detail below.

## ALX3d Beliefs Versus CK Concepts

CK's first definition of "design" [1] can now be written as: design is the process of expanding *R* and *D* to go from Formula 2 to Formula 1.

The CK notion of *K-relativity* is consistent with ALX3. That is, in CK, determining the logical status of a proposition is done with respect to the available knowledge (K-space). If there is no K-space or if it is empty, nothing can be done. If K-space is not sufficiently rich, C → K, K → K, and K → C operators may be unable to give useful results.

In ALX3d, $R \Rightarrow D$ is evaluated with respect to what is known to the design agents (our equivalent of K-space) in the current "world"—which is similar to K-relativity.

Where ALX3d and CK differ here is that CK does not seem to account for the differences in what each agent knows, whereas ALX3d can (at least to a degree). The ability to distinguish between what two agents know is inherent in ALX3. It is perfectly reasonable to write that $K_a\psi \wedge K_b\psi$; that is, agent *a* knows ψ but agent *b* does not.

One can then use this to describe some aspects of the collaborations between multiple agents. For example, one may say that in the final state all agents must *know* the design, $\Lambda_a K_a D$. This might be reasonable in small projects, but in large and complex projects such as the design of a commercial airliner, this is not feasible or even necessary. In the latter case, one might associate an agent *i* with an element of *D*. The element is the conjunction of only *some* of the propositions that constitute *D*—call it $D_i$; a necessary condition here would be $D \Rightarrow D_i$ in any desirable state. Thus, each agent knows a part of the design, $K_iD_i$. One may then imagine a supervisory agent *s* (a project manager or team leader), who would not need to know the design, but only that the agents in his team know their respective design elements. We can write this as $K_s(\forall i\ K_iD_i)$.

This is only a superficial description of the organizational/social relationships that would occur. The author's goal here is not to exhaustively describe these relationships, but only to suggest that ALX3 has the capacity to represent them. Detailing the organizational/social aspects of engineering teamwork will be the subject of a future paper. In the meantime, one can refer the interested reader to [4], in which some further details of how ALX3 can be used to describe organizations is discussed.

In CK, “the formulation of the requirements is a first concept formulation which is expanded by the designer in a second concept that is called the proposal” [1]. There is a problem, however, if we accept requirements as C-space elements, as implied in [1]. We must *know* at least some of the requirements, i.e. know their logical status, because in CK, one can only reason about things with defined logical status. Without this, the initial requirements, the initial design (the “first concept formulation”), and the design proposal are all extra-logical, which defeats the purpose of formalizations like CK. Other requirements may begin as beliefs (in C-space) and migrate through design actions to K-space, but we cannot create the design without actually *knowing* (some of) the product’s requirements.

In ALX3d, on the other hand, we can reason about beliefs because they are assumed to have a “temporary” logical status as described above. This means that ALX3d is a richer representation than CK, without violating the intention of CK.

CK concepts have “no logical status.” Without this, one cannot identify individual concepts from within sets of concepts because this is done in first order logic by finding a (possibly complex) predicate that is true for only one element of a set, thus identifying it. The treatment of sets is conventionally done in logical systems with *set theory*, and typically with a particular development of set theory called Zermelo-Fraenkel (ZF) set theory [6]. In ZF, there is a specific axiom, the *Axiom of Choice*, which defines how one may select individuals from sets. The axiom is necessary in any development requiring the individuation of specific set elements. In CK, then, the Axiom of Choice is excluded because concepts are without logical status. The set of concepts is defined implicitly by means of propositions that enumerate characteristics of any suitable concept.

This is unwarranted in ALX3d. Beliefs are assumptions regarding the logical status of propositions. The collection of beliefs (the belief structure) allows provisional and conditional reasoning. A belief can be retracted, which would then require validating all inferences that include the retracted belief. Nonetheless, reasoning is possible. Therefore, we do not need to exclude the Axiom of Choice in ALX3d; indeed, at this time, the author finds no conclusive argument either for or against the inclusion of the Axiom of Choice. This gives us a certain greater flexibility of design representation.

The capacity to reason about beliefs is an important element of ALX3d that CK does not provide. What designer would pursue a concept he did not *believe* would be fruitful? What company would pursue a product development project that its members did not believe would be successful? Designers cannot wait for the certainty of knowledge all the time, so they must make assumptions to proceed, and backtrack if those assumptions turn out to be wrong. The belief system support in ALX3d lets us do this.

## Translating CK Operators to ALX3d

In CK, one changes the logical status of concepts by adding or subtracting properties. In logic, we represent properties with propositions. For example, *weight(motor, 5 kg)* asserts a motor has the property of weight with value 5 kg. Description logics [7] can be used to develop ontologies (formal descriptions of bodies of knowledge) based on those properties, but this is beyond the scope of this paper.

In ALX3d, we add and subtract propositions that ascribe properties to *R* and *D*, such that subsequent application of validation actions hopefully turns beliefs into knowledge. From this, all four kinds of CK operators (C → C, C → K, K → C, and K → K) translate easily to ALX3d.

Consider an example in [1] about bicycles with pedals and "effective wings." "Bicycles with Pedals" (denoted by the predicate *bp*) leads to a ALX3d belief $B_a(\exists x \ bp(x))$, while "Bicycles with Effective Wings" (denoted by the predicate *bew*) leads to $B_a(\exists x \ bew(x))$. "Bicycles with pedals and effective wings" is written $B_a(\exists x[bp(x) \ni bew(x)] \ )$. Note that these are beliefs held by an agent—that is, they are design concepts. The real question is not whether such a design is possible but rather whether $R \Rightarrow x$; that is, does there exist a situation wherein a bicycle with pedals and effective wings is appropriate. If there is no such situation, then even considering the concept is pointless.

The answer depends on what is known (the content of CK's K-space). For example, in a dome with an atmosphere on the Moon or some other very low gravity setting, *bew(x)* might be perfectly reasonable. The reason why *bew(x)* seems silly is because of the situation (context) we *assume* in the absence of specific knowledge of the implication. Context logics [8] and work on situated design [9] might also help here, but again this is beyond the scope of the current paper.

Let us now consider the CK operators in more detail.

In CK, K → C operators add or subtract properties, written as propositions in K-space, to or from concepts thus creating disjunctions in C-space [1]. This corresponds to the design activity of *generating alternatives*. These operators expand C-space with elements from K-space. The disjunction arises from considering that adding a property partitions the set of all extant concepts into a set of those that satisfy the property and another of those that do not. Furthermore, C → C operators expand or "flesh out" a concept by adding yet other propositions.

In ALX3d, these two kinds of operators are treated uniformly. We can add a new design proposition $d'$ to $D$. To write this in ALX3d, we let $R_1$ and $R_2$ be any pair conjunctions of subsets of the requirements, such that $R_1 \wedge R_2 \Leftrightarrow R$. Then we can write:

$$B_a(R_1 \Rightarrow d') \, 4 \, B_a(D' : \ D \ni d'), \text{ or} \tag{4a}$$

$$B_a(R_1 \Rightarrow \neg d') 4 \ B_a(D' \ : \ D \ni \neg d'). \tag{4b}$$

Formula 4a reads that if a designer were to believe that some requirements that were theretofore unsatisfied, are satisfied by a new design proposition $d'$, then the agent would believe that the design can be improved by adding the new proposition to the current design. Formula 4b is similar, but for propositions believed to not fulfill any requirements.

In a sense, there is no real distinction in ALX3d between what CK calls calls K → C and C → C operators, because in ALX3d they are simply means to develop new concepts. If it is necessary, for some reason, to distinguish between them, then one only need recognize that K → C operators will involve knowledge operators in ALX3d while C → C operators will involve belief operators in ALX3d. The actions associated with these state transitions (implied by the $\therefore$ operator) are actions by which a designer proposes new aspects of a design. This partitions states into those where $d'$ holds and those where $\urcorner d'$ holds.

We can see how this reasoning can be represented with ALX3d in the following hypothetical sequence of activities.

1. An initial state is assumed of $\mathrm{K_aR} \ni \mathrm{B_aD}$. The design agent knows the (initial) requirements $R$ and believes the (initial) design $D$.
2. The agent attempts to perform a validation action, but finds that validation cannot be done: $\mathrm{K_aR} \ni \mathrm{B_aD}\ 4\ \neg\mathrm{K_a}(\mathrm{R} \Rightarrow \mathrm{D})$.
3. Since the validation cannot be done, either $D$ or $R$ must be explicated further. The agent chooses to expand $D$ with a new proposition $d'$: $\mathrm{B_a}(\mathrm{R_1} \Rightarrow \mathrm{d}')\ \mathrm{D4K_aR} \ni \mathrm{B_a}(\mathrm{D} \ni \mathrm{d}')$.
4. The agent attempts to validate the design again. This time, validation fails: $\mathrm{K_aR} \ni \mathrm{B_a}(\mathrm{D} \ni \mathrm{d}')4\mathrm{K_a}(\neg(\mathrm{R} \Rightarrow (\mathrm{D} \ni \mathrm{d}')))$.
   Since the first validation (step 2) could not be done, but including $d'$ causes validation to fail, the only alternative is $\urcorner d'$.
5. The error is corrected: $\mathrm{K_a}(\neg(\mathrm{R} \Rightarrow (\mathrm{D} \ni \mathrm{d}')))4\mathrm{K_aR} \ni \mathrm{B_a}(\mathrm{D} \ni \neg\mathrm{d}')$.

If validation of the design in step 5, cannot be performed, we know that neither $D \ni d'$ nor $D \ni \neg d'$ is a sufficient solution and that more expansion must be done to the design (and possibly the requirements).

If the validation in step 4 yielded $\neg\mathrm{K_a}(\mathrm{R} \Rightarrow (\mathrm{D} \ni \mathrm{d}'))$ , we would not be able to choose between $d'$ and $\urcorner d'$ because neither led to a suitable design. Some alternative courses of action here include the following.

- The agent could try a different validation action since it might be the validation action itself that cannot operate on the available information in $R$ and $D \ni \neg d'$.
- The agent could pursue both $D \ni d'$ and $D \ni \neg d'$ as design alternatives until validation does give a distinct answer.
- The agent could change $R$ and try to validate again.
- The agent could choose $d'$ or˜$d'$ based on the agent's own preferences (e.g. $d'$ $P^{\sim}_{a}d'$).

Changing $R$ is done just as changing $D$, by adding $r'$ or˜$r'$ to $R$.

If, on the other hand, the validation in step 5 *fails*, then neither $d'$ nor˜$d'$ is a suitable solution. This means that there is an error in $R$, since one of $d'$ or˜$d'$ must be true (whether the agent knows which one is true is not the point). In this case, one must use some sort of strategy to backtrack to earlier states until one finds a state in the history of changes to $R$ where either $d'$ or $d'$ does hold.

The preceding discussion applies equally to CK's *expanding partitions* (operations that increase the number of possible design concepts) and *restricting partitions* (operations that decrease the number of possible design concepts). Expanding partitions are those for which the designer only believes $d'$, $B_a d'$; restricting partitions are those for which the designer knows $d'$, $K_a d'$.

In CK, the current state can be "backtracked" by returning to a previous state, but the theory itself does not formally describe this (for example, by some appropriate operator). In ALX3d, however, we can use *belief retraction* to formalize backtracking directly. We can write this as $d': D \backslash d'$—that is, $d'$ is like $D$ but without $d'$. Backtracking in this manner applies only to beliefs and not to knowledge, because it is a principle of action logics that knowledge cannot become unknown once it is known. (Note: this is *not* the same as backtracking in logic programming languages like Prolog.)

Details of this strategy constitute future work; here it is sufficient to recognize that such representations are possible in ALX3d.

Let us now consider C → K operators, which turn a concept into knowledge in CK. In ALX3d, these are *validation actions*. Once a (design) concept becomes knowledge in CK, it is a sufficient design solution. In ALX3d, validation actions validate a belief, turning it to knowledge. As in CK, such actions include conducting mathematical analyses, experimental tests, etc. The only substantive difference is that in ALX3d the key belief that must be validated is the implication $R \Rightarrow D$, rather than the design $D$ itself. In ALX3d, knowing $D$ follows from knowing (validating) this implication.

Finally, CK's K → K operators expand knowledge space through logical and scientific reasoning. Any such operator is available within the first-order logic underlying ALX3d.

## An example

In Hatchuel et al. [10], present examples of the application of CK theory. In this section, the author will discuss how ALX3d can achieve at least the same level of description as CK. We will use one of the examples in [10]: the design of a new chemical ($Mg\text{-}CO_2$) rocket motor for use in Mars exploration missions. In [10], the case study is divided into four phases; we will follow the same layout here.

The initial state (Phase 0) is the proposal that a $Mg\text{-}CO_2$ engine would be "better" than the conventional solution. Per [10], we label this proposal $C_0$. In CK, the proposal is a *concept* because it has no logical status. In ALX3d, the proposal is a *belief*, a statement that we assume to be true and then reason with it until we can either prove or disprove it. We write it in ALX3d as $B_a C_0$.

In Phase 1 of the case study, an attempt was made to use the Mg-$CO_2$ concept for a *sample return mission* to Mars (labeled $A_1$ in [10]). We can write this in ALX3d as $B_a(C_0 \ni A_1)$. An "evaluation" was then carried out by comparing the new motor to existent ones with respect to the key criterion of minimum landed mass on Mars for a sample return mission. This constitutes a validation action in ALX3d. It was found that the new motor failed the validation; that is, the new motor is not as good as a conventional motor. In ALX3d as in CK, this only means that $K_a \neg(C_0 \ni A_1)$. As in CK, and by the fundamental properties of first order logic, this does not necessarily imply that $C_0$. So we can preserve our core belief, $B_a C_0$, by contending that $B_a A_1$, which would account for the validation result. The designer's new belief is then $B_a(C_0 \ni \neg A_1)$ . We can write all this as:

$$\begin{array}{l}(C_0 P_a \neg C_0 \ \wedge (B_a(C_0 \ni A_1) \quad \therefore K_a \neg(C_0 \ni A_1)) \Rightarrow K_a(\neg C_0 \vee \neg A_1)) \\ \quad \therefore B_a(C_0 \wedge \neg A_1).\end{array} \tag{11}$$

Note that in CK, it is assumed at this point that $A_1$ is false; i.e. that the new motor will not work for a sample return mission. This is in fact incorrect. All that the agents can infer logically is that they *believe* the Mg-$CO_2$ motor will not work in this kind of mission; they only *know* that the combination $C_0 \ni A_1$ will not work. At this point, CK would have us accept the validity of our main proposal $C_0$, but the whole point of the exercise is to determine if the concept has any merit at all. We see then that ALX3d is more expressive of the actual state of affairs in this case.

In Phase 2 of the case study, it is reported that a study conducted of mission profiles *excluding* sample return missions $\big($i.e. $B_a(C_0 \ni \neg A_1)\big)$ yielded no positive results, but that this was due to an excessive number of attributes placed on the problem during evaluation. It is also suggested that CK provides a key insight here—that those excess attributes must be removed in order to discover other possible solutions. However, the current author has been unable to find a clear indication of how CK itself accommodates this. Indeed, the current author contends that this is a feature of an ontological representation of design problems as a composition of facts. This is how logic works in general, and is not a feature particular to CK. There is an old adage: *always question premises*. In this case, the premises are the attributes. Questioning them involves determining whether they are necessary or simply accepted by fiat, convention, or error.

In the case of the Mg-$CO_2$ motor, it is evident that all scenarios had at least one attribute in common: that the motor would be used in transit to Mars. This is the premise that is questioned in [10]. In fact, then, the belief (the CK concept) $B_a(C_0 \ni \neg A_1)$ was interpreted *incorrectly* because the premise of using the motor in transit is not part of the concept; that is, $A_1$ (use for sample return missions) does not necessarily imply use in transit. It would appear then that human error is the root cause of this situation.

Let us assume, lacking other information from [10], that we should have distinguished transit as a key element of the mission profile. Let us further assume for the sake of simplicity that the mission profile can be exhaustively divided into two main segments: the transit to Mars (both going and coming) and the mission on Mars. We can rewrite the original belief $B_a(C_0 \ni A_1)$ now as $B_a(C_0 \ni A_{1t} \wedge A_{1m})$ where $A_{1t}$ stands for "using the motor for transit" and $A_{1m}$ stands for "using the motor on Mars".

Given this, the failure of the validation action noted above then tells us that if we wish to maintain $C_0$, then either $A_{1t}$ or $A_{1m}$ must be wrong. We would have then had the belief: $B_a(C_0 \ni \sqsupset [A_{1t} \vee A_{1m}])$ . That is, the Mg-CO2 motor might be suitable for either transit to Mars or operation on Mars, but not both.

Since all the investigated scenarios involved $A_{1t}$, the logical alternative here, regardless of the use of CK or ALX3d, is to use the Mg-CO2 motor for purposes other than the transit to Mars. Practically, this is equivalent to using the motor only on Mars, labeled $A_2$ in [10], which we can represent in ALX3d as $B_a(C_0 \ni A_2)$, so long as we also accept that $A_2 \Rightarrow \neg A_1$.

Continuing through the case study, [10] then identifies four other attributes that constitute possible uses of the $Mg\text{-}CO_2$ motor on Mars: $A_3$—"used for mobility," $A_4$—"unplanned mobility," $A_5$—"emergency lift-off," and $A_5$'—"additional distance". The systematic appearance of these alternatives follows from the use of CK only insofar as CK implies the use of breadth-first searches, which is our only logical course of action. A new concept is then specified in [10], which can be written in ALX3d as $B_a(C_0 \ni A_2 \ni A_3 \ni A_4 \ni A_5)$ .

There are, however, two problems here. First, both this belief and its CK variant mean that the agent believes an appropriate design is a $Mg\text{-}CO_2$ motor used for *unplanned emergency lift-off mobility on Mars*; that is, the mission involves the *simultaneous* occurrence of $A_2$ through $A_5$, because of the logical conjunctions.

The current author believes this was not the actual intention. Rather, it makes more sense that the intention was for the new motor to be used in *any combination* of the situations denoted by $A_2$–$A_5$. That is, a disjunction should have been used, i.e. $B_a(C_0 \ni (A_2(A_3(A_4(A_5)))$ , to correctly represent that any of $A_2$–$A_5$ could constitute an appropriate use of the $Mg\text{-}CO_2$ motor.

The second problem arises from considering the nature of propositions $A_2$–$A_5$. Specifically, three important facts are missing. First, $A_2$ is a generalization of $A_3$–$A_5$; that is, "use on Mars" includes "use for mobility," "use for unplanned mobility," and "use for emergency lift-off". Second, $A_3$ is a generalization of $A_4$ and $A_5$; that is "use for mobility" covers both emergency and unplanned mobility. Third, some design activities must have occurred to get from $A_2$ to $A_3$ and then to $A_4$ and $A_5$; that is, for example, in moving from "use on Mars" to "use for unplanned mobility" implies some design action that identifies the required mobility as *unplanned*.

The current author therefore suggests the following ALX3d representation for situation reported in [10]:

$$\left(\left(B_a\left(C_0 \ni A_2\right) 4B_aA_3\right) 4B_aA_4\right) 4B_a\left(A_5\left(A'_5\right)\right). \tag{12}$$

Formula 12 captures a great deal about the situation:

- Initially, the agent believes the Mg-$CO_2$ motor is a viable alternative for use on Mars ($C_0 \ni A_2$).
- There is a causal relation between *use on Mars* and *mobility on Mars* ($A_3$), so some design action must occur for the relation to hold.
- To achieve $A_3$, there must exist some design action (a "conceptual expansion" in CK) that moves the agent there. This is a human cognitive act connecting a means (the motor) to a desirable capability (mobility).
- Similarly, once the agent believes $A_3$, there is an action that will lead the agent to a new state where the mobility is *unplanned* ($A_4$).
- Finally, once the agent is in a state of believing $A_4$, there is an action that will lead the agent to believe either *emergency lift-off* or *additional distance* ($A_5$ or $A_5$') as alternate suitable situations.

We note that once we reach a state where $A_5$ or $A_5$' is true (and only in such states), then we can also say that $B_a(A_5(A_5')) \Rightarrow A_4 \Rightarrow A_3 \Rightarrow A_2$, which gives a causal chain back to the original propositions. Again, this demonstrates that ALX3d provides a richer representation of designing than CK, while remaining consistent with the intent and general principles of CK.

Finally, in Phase 3 of the case study, a comparison of the Mg-$CO_2$ concept and an alternative design, the ExoMars Rover, is reported. The concept used is that of Mg-$CO_2$ combustion for *unplanned mobility on Mars*; that is, $A_5$ and $A_5$' are not used. The ExoMars performance constraints are given as (a) motor weighing less than 60 kg, (b) mission life of no more than 180 days, (c) maximum power consumption of 200 W, and (d) minimum 10 km range. These constraints are used to limit a performance domain that the Mg-$CO_2$ concept must satisfy. Based on existent knowledge (e.g. principles of rocket propulsion), two key design parameters for the Mg-$CO_2$ concept are discovered—motor mass ($m_m$), and mass of the $CO_2$ acquisition plant ($m_p$)—that can be used to calculate values for performance characteristic of lifetime (t), power (p), and range (r).

The values of the parameters exist within a bounded domain; any value set within the domain constitutes a possible solution, i.e. where the Mg-$CO_2$ motor concept can compete against the ExoMars alternative. The authors argue [10] that this opens up new possibilities for mission concepts and design alternatives that would not have been noticed otherwise.

Phase 3 is described in [10] using text and diagrams, and it is not necessarily clear which activities are derived from CK and which arise simply from the use of rational, logical reasoning in general, or innovative thinking about the problem. No matter which is actually the case, the current author will show that stages of development that occurred in Phase 3 can be represented directly in the language of ALX3d and consistently with CK. We recall that the goal is not to have ALX3d lead designers through the process, but rather to capture descriptively the reported design activities.

First, let $\Delta D$ be true only for any design concept $D$; that is, $\Delta D$ is a predicate that identifies design concepts. We would therefore assert $\Delta C_0$ to mean that $C_0$ is a design concept. The set of all known design alternatives satisfying some propositions $\psi$ is given in ALX3d by

$$\mathrm{D_a}(\psi): \quad \{\mathrm{x} \ : \ \mathrm{K_a}\mathit{\Delta}\mathrm{x} \Rightarrow \mathrm{B_a}\psi\}. \tag{13}$$

The agent knows that $\Delta x$ because the agent asserted it. Note that $\mathrm{K_a}\Delta\mathrm{x}$ (knowing that $x$ is a design concept) does not imply $\mathrm{K_a}(\mathrm{R} \Rightarrow \mathrm{x})$ (knowing that $x$ is an acceptable design concept for a given problem). For example, consider the previous example of bicycles with pedals and effective wings. Let $B_0$ be "bicycle," $P_1$ be "with pedals," and $P_2$ be "with effective wings". Furthermore, assume we were interested in finding alternatives that have wings ($P_2$) to "bicycles with pedals" ($\mathrm{B_0} \ni \mathrm{P_1}$). The set of alternatives is given by $\mathrm{D_a(P_2)}$: $\{\mathrm{x}$: $\mathrm{K_a}\Delta\mathrm{x} \Rightarrow \mathrm{B_aP_2}\}$, which would include bicycles with pedals and any other design concept satisfying "with effective wings". Similarly, $\mathrm{D_a(P_1)}$: $\{\mathrm{y}$: $\mathrm{K_a}\Delta\mathrm{y} \Rightarrow \mathrm{B_aP_1}\}$ would contain the alternatives to bicycles with effective wings that also satisfy "with pedals".

Now, returning to the Mars case study, let $\mathrm{C_1} : \ \mathrm{C_0} \ni \mathrm{A_4}$; i.e. $\mathrm{C_1}$ is the concept of using Mg-$\mathrm{CO_2}$ motors for unplanned mobility on Mars. The designer can assert $\Delta\mathrm{C_1}$ as a possible design. The designers' state thus includes $\mathrm{B_aC_1}$. To look for alternative concepts, we need to identify concepts that involve $\mathrm{A_4}$—all cases of unplanned mobility on Mars—but without $\mathrm{C_0}$. We can write this as $\mathrm{C_1 \backslash C_0}$. Now the set of all design alternatives is just:

$$\mathrm{D_a}(\mathrm{C_1 \backslash C_0}): \quad \{\mathrm{x} \ : \quad \mathrm{K_a}\Delta\mathrm{x} \Rightarrow \mathrm{B_aA_4}\}. \tag{14}$$

$\mathrm{D_a(C_1 \backslash C_0)}$ includes all the design concept alternatives to $\mathrm{C_1}$. To gather these alternatives, the designers began with a belief $\mathrm{C_1}$, and did the appropriate research (a C → K operator in CK) to find the design alternatives $\mathrm{D_a(C_1 \backslash C_0)}$. We can represent this as a causal relation in ALX3d:

$$\mathrm{B_aC_1} \ 4\mathrm{D_a}(\mathrm{C_1 \backslash C_0})(\neg \ \mathrm{D_a}(\mathrm{C_1 \backslash C_0}). \tag{15}$$

That is, every subsequent state following the search for design alternatives is one that either definitely does or does not have such alternatives. Obviously, to continue the case study, we must assume that $\mathrm{D_a(C_1 \backslash C_0)}$ is in fact the case.

One might ask: is there some feature of a state where $\mathrm{B_aC_1}$ that draws the agent to look for alternatives? The original case study [10] only states "…the prototype should overcome the rover solution for the next known missions…." At this time, the current author can only propose that setting a goal of comparing concepts to alternatives is an extra-logical design principle. This activity may be a part of a validation action; that is, it could be one way to determine if a design concept has merit. This might suggest an axiom (a statement accepted as true but not provable within a logic) for ALX3d, but setting out exactly what this axiom might be remains an item for future study.

We can now describe this phase of the case study in ALX3d. Let the design parameters $P$ for the Mg-$\mathrm{CO_2}$ motor be $m_m$ and $m_p$. Let the values of the design parameters be written as functions mapping a design parameter to a value: $m_m(d)$, $m_p(d)$. Let the performance metrics of any designs be $M: \{p, t, r\}$ (power, lifetime, range). The values of metrics can be written as functions $p(d)$, $t(d)$, and $r(d)$ for a design $d$.

The metric values lead to (or "cause" in ALX3d) the parameter values. That is, the case study indicates that $p$, $t$, and $r$ were dependent values, and $m_m$ and $m_p$ were the independent values. In ALX3d, this is written:

$$\forall \mathrm{d}\left[\left[\mathrm{m_m(d)} \ni \mathrm{m_p(d)}\right]4[\mathrm{p(d)} \ni \mathrm{t(d)} \ni \mathrm{r(d)}]\right]. \tag{16}$$

Furthermore, the values can be partially ordered, e.g. $p(x)O_p p(y)$, for different designs, where $O_p$ is a generalized ordering operator on $p$.

Constraints were defined in the case study based on knowledge of rocketry and physics. Let the constraints be written as: maximum power consumption $p = 200$ W, expected operating life $t = 180$ days, and minimum range of operation $r = 10$ km. A condition for a satisficing design [5] is given by: $p(d) < p \ni t(d) < t \ni r(d) > r$. We can now write a satisficing goal for the Mg-$CO_2$ concept $C_1$ as a belief in a causal relation. Since the design is only a concept, we cannot *know* this satisficing relation, but only believe it. In ALX3d, we can write a satisficing goal for this case as:

$$\mathrm{G^s[C_1]:\ B_a\left[\left[m_m(C_1) \ni m_p(C_1)\right]4[p(C_1) < p \ni t(C_1) < t \ni r(C_1) > r]\right]}. \tag{17}$$

This statement essentially captures the domain of possible values for the identified design parameters such that any design that satisfies this statement is a possible solution. This kind of *formal* representation of the goal of design activity is not available in CK theory.

We can also go beyond the case study somewhat by considering a way to find the best design within the domain of satisficing solutions given by $G^s[C_1]$. Let there be two designs based on $C_1$, defined by $C_2 : C_1 \ni x$ and $C_3 : C_1 \ni y$ and that both satisfy $G^s[C_1]$. We can use the formalism of trade-off goals [3] to capture the agent's preference for one satisficing design over another. Given $C_2$ and $C_3$ as defined above, and letting $u$, $v$, and $z$ be other satisficing designs in $G^s[C_1]$, and letting $\varphi$ and $\psi$ stand for any two of the metrics, we can write the following.

$$\begin{aligned}\mathrm{G^t}[\varphi(\mathrm{C_2})]:\ & [\varphi(\mathrm{C_2})\mathrm{P_a}\varphi(\mathrm{C_3})] \ni [\varphi(\mathrm{C_2})4\psi(\mathrm{u})] \\ & \ni \mathrm{B_a}[\neg\exists \mathrm{z}\ (\varphi(\mathrm{z})\mathrm{P_a}\varphi(\mathrm{C_2}) \ni \varphi(\mathrm{z})4\psi(\mathrm{v}) \ni \psi(\mathrm{u})\mathrm{P_a}\psi(\mathrm{v})],\end{aligned} \tag{18a}$$

$$\mathrm{C_2P_aC_3} \Rightarrow \mathrm{B_a}\left[\varphi(\mathrm{C_2})\ \mathrm{O}_\varphi\varphi(\mathrm{C_3})\right]. \tag{18b}$$

This says that $C_2$ is preferred to $C_3$ if $C_2$ attains a "better" value of one of the metrics ($\varphi$) than does $C_3$, and doing so will not limit finding a more preferred value for one of the other metrics ($\psi$).

We have now developed a new model of the case study in [10] that is grounded far better in a formalism of design activities than CK can provide.

## Potential Benefits of ALX3d

The author has introduced ALX3d, a formal theory of design activities built upon the action logic ALX3, and designed to account for the key features and intent of CK theory. A case study from the existent CK literature was reworked in ALX3d to demonstrate its representational richness.

The purpose of any formal model, including ALX3d, is only to provide a reasoning tool, a mechanism to allow one to reason in as rational and structured a manner about a domain. All models are by definition incomplete; else they would be indistinguishable from the thing being modeled. Models like ALX3d provide one perspective on a thing. There are other equally meaningful ways to think about design processes. ALX3d in no way discounts them; it only provides an alternative.

Furthermore, this work in no way invalidates CK. Rather, it demonstrates that the fundamental premises of CK are reasonable premises regarding the act of designing; namely, that there is an important difference between knowledge and concepts, and that a rational (logical) process can describe (but not necessarily explain) at least some parts of the act of designing. ALX3d also demonstrates the power of logical systems to capture essential aspects of design processes, especially the decisions that designers must make based not only on knowledge but also on their beliefs and preferences.

ALX3d is a research tool, not something to be used by practicing designers. However, continued development of theories in mathematics and the sciences have often let eventually to practical benefits for designers. It is reasonable to assume the same could happen with logical theories like ALX3d and CK. As ALX3d matures, it will be possible to use it for several purposes in this regard, some of which include the following.

**Appeal to formal systems researchers.** CK theory, which has distinct benefits as a design research tool, is somewhat hindered because it does not conform to conventions of formal systems. ALX3d maintains the intent and basic principles of CK while casting it in a form more readily understood by those with grounding in formal systems. As such, ALX3d makes CK theory more appealing to the community of design researchers who understand and use formal systems, including researchers in artificial intelligence, computer science, and cognitive science.

**Reasoning about documented design processes.** Assuming a complete description of a design process as documented either in industry or the literature can be constructed (and the author currently believes this is entirely possible), then the description can be reasoned about using the inference rules that are built into ALX3 to study the process, and find and address its problematic aspects. This would significantly advance our understanding of the nature of engineering design.

**Construction of new design processes.** It may well be that in the natural course of analyzing design processes, new process descriptions may arise that could significantly improve the design capability of a group of designers. The description of rules for establishing trade-off goals, and for switching between different types of design tasks—as outlined in the preceding sections—are examples of this. It may be that new, “industry-strength” methods can be developed by considering different ways that such activities can be described in ALX3d.

**Construction of new computer-based design aids.** Logical systems are well suited to implementation in computer tools. It should be possible to use ALX3d to develop new design applications of artificial intelligence and knowledge-based systems. Such systems may also yield significant advantages for practicing designers. For example, it may be possible to develop tools that will suggest sequences of design activities that are more likely to lead quickly to better (or at least satisficing) designs. One may also envision case-based reasoning engines that use sequences of actions as cases.

## Conclusion

Beyond what is currently possible with CK, ALX3d leverages ALX3 to provide a richer framework for describing design activities in formal terms. While adding support to the CK approach, ALX3d also demonstrates the potential benefits of using formal systems in design research. Although ALX3d is still being developed, there are strong indications, as demonstrated in this paper that it may be a useful tool for design research.

## References

1. Hatchuel A, Weil B (2009) CK design theory: an advanced formulation. Res Eng Design 19 (4):181–192
2. Kazakci AO, Tsoukias A (2004) Extending the CK design theory to provide theoretical background for personal design assistants. In: Marjanovic D (ed), Design 2004, pp 45–52
3. Salustri FA (2003) Towards an action logic for design processes. In: Proceedings of international conference on engineering design, paper #1051
4. Huang Z (1994) Logics for agents with bounded rationality. Ph.D. thesis, University of Amsterdam
5. Simon HA (1981) The Sciences of the Artificial. The MIT Press, Cambridge, Mass
6. Copi IM (1979) Symbolic logic, 5/e. Prentice-Hall, Englewood Cliffs, NJ
7. Brachman RJ, McGuinness DL, Patel-Schneider PF, Resnick LA (1991) Living with a classic: when and how to use a KL-ONE-like language. In: Sowa JF (ed) Principles of semantic networks: explorations in the representation of knowledge. Morgan Kaufmann Publishers Inc, San Mateo, pp 401–456
8. Akman V, Surav M (1996) Steps toward formalizing context. AI Magazine 17(3):55–72
9. Gero JS (2004) Situated design computing: Introduction and implications. In: Marjanovic D (ed), Design 2004, pp 27–36
10. Hatchuel A, Le Masson P, Weil B (2004) CK theory in practice: lessons from industrial applications. In: Marjanovic D (ed), Design 2004, pp 245–257

# On an Integrated Analytical Approach to Describe Quality Design Process in Light of Deterministic Information Theory

**Tamer El-Khouly and Alan Penn**

**Abstract** This paper introduces a methodology to analyse design linkographs by quantifying entropy at each single move throughout the design process. The method adopts the *deterministic information theory* proposed by Titchener to develop a quantitative model aiming to highlight the significant nodes by coding the dependency relations (*backlinks* and *forelinks*) into character strings of information. Two computational methods are suggested to quantify *T-code sets* on a micro-level at every single utterance. This proposition is intended to capture repetition of patterns and hierarchy in the linkograph pattern. This quantitative approach is integrated with a qualitative model of judging sketching episodes and evaluating the relations between the instantaneously evolved products during the design process such as the interim sketches. The results point at significant correlations between quantitative and qualitative models on the key nodes to occur in the process to identify the emergence of novel ideas and describe design creativity.

## Introduction

A *linkograph* is a representation that traces the associations of every single move (utterance), the design process can be looked at as a linkograph pattern that displays the structure of the design reasoning. The venues of dense interrelations are overtly highlighted on the graph as well as shallow ones and can be further interpreted through the emerging artefacts (interim sketches etc.) along the process. To reveal the structure of design reasoning, the proposed analytical model builds on the hypothesis that the emergence of ideas is an outcome of the interplay between *reflection in the mind* and the *alternative solutions generated.*

---

T. El-Khouly (✉) · A. Penn
University College London, London, UK
e-mail: t.el-khouly@ucl.ac.uk

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_25,

The aim of this proposition is to discover how to correlate a quantitative method to signify pivotal moments or key venues in the development of ideas with the corresponding cognitive activities, foremost of which are externalised sketches and drawings.

The prime goal of an objective tool to quantify the dependency relations between design units in a linkograph is to describe the characteristics of a design idea and its hierarchical structure; and to develop a qualitative model that describes a designing episode and identifies starting and ending positions of reflections between the mind and the media of representation.

The multi-level complexity of cognitive process and syntheses of designing are the primary motivations for the development of an integrated analytical approach to understand the nature of the design process and its products.

Relations in linkography are looked at from two perspectives, *information* and *entropy*. We propose that relations can be transcribed into a *character string* of information coding dependencies into binary symbols.

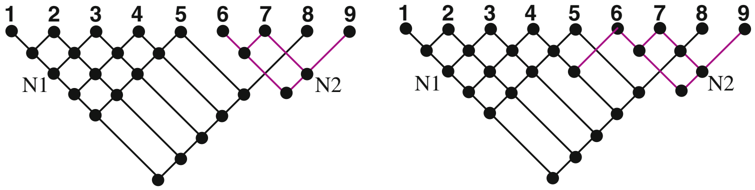
*Deterministic information theory* proposes a set of codes where each code measures a parameter on the *character string of information*. Titchener developed this theory in stages. T-code sets comprise two primary algorithms: *T-decomposition* decomposes the complexity of the character string into its possible primary level; in contrast, *T-augmentation* augments the primary units' "codewords" to reproduce the full character string. By using this theory we can compute the T-code measures such as *T-entropy*, *T-complexity* and *T-information* for any string [1–3].

## The Design Process as a Multi-level Complex System

This study begins by highlighting the characteristics of linkographs, which have been inferred through a series of analyses on different patterns of case studies. The following structure forms the hierarchy of linkograph: nodes, clusters and networks.

*Segregation* or *integration* of networks varies from case to case: the pattern is sometimes coherent and parts are connected despite the diversity of the cognitive activities undertaken, but this cannot be postulated as a general rule because sometimes a total separation occurs between two or more subsets. Based on this, the structure of linkograph varies between *fully connected and saturated* or *totally random and disordered,* Fig. 1. Both are extreme situations in design thinking. Thus three prototypes of linkographic patterns are categorised: *highly ordered*, *structured* and *disordered*, reflecting *integration*, *coherence* and *diversification* respectively.

The linkograph reflects a state of design that changes through time. This change might underlie an entire state with properties that cannot be identified through the outward appearance of the pattern. Schön [4] suggests that a design transforms its state according to the change of repertoires in the mind. The challenge of this paper is to understand how the transformation of ideas from one state to another



Two “Separate” networks:
N1 (black lines) & N2 (magenta lines)

Two “connected” networks “linked”
In nodes: (**4**⟷**6**) & (**7**⟷**8**)

**Fig. 1** The relations between networks in a linkograph; *separated* or *connected*

can be captured. In addition, the *reflective practice* (a methodological design paradigm developed by Schön [4]) plays a vital role in the manipulation process of the artefacts of design. Why do certain interim artefacts not reflect exactly what was happening in the designer’s mind at a particular moment? Although the sketches result from the mind, there will be instances when the sketch will reflect back to some (buried) insight in the mind.

Goldschmidt [5] revealed that a designer does not represent images held in the mind, as is often the case in sketching by non-designers, but creates *visual displays* that help induce images of the entity that is still being designed. This is considered an *intermediate* medium of representation to mediate between mental manifestations and the design outcomes.

In relating the cognitive processes with the linkograph, various patterns of mental representation can be inferred from studying the relations that can be made between the units of design. Figure 2 demonstrates how a linkograph can be configured from *ordered* to *disordered* patterns.

What deserves attention is how the design process is built up from the parts to the whole in order to look at the venues of high creativity within the structure. The proposition is therefore to investigate the synthesis of relations in every action and globally to understand the structure and describe the design process. In this context, it is vital to distinguish between information and entropy since most of the current publications on protocol analysis adopt entropy as a central element to describe the design process.

## Information and Entropy

*Information* and *entropy* are two angles from which to look at linkography. While the information theorist looks at the *probability* that can be created for a sequence of relations for a single *item*, the entropy theorist considers the *set* (which is made up of items) a microstate on its own for the system. The two theories are in opposition. Entropy grows with *probability*, while information increases with *improbability*. The less likely an event is to happen, the more information its occurrence provides.

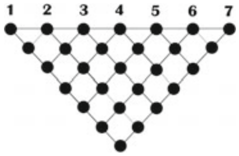
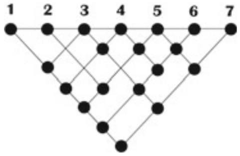
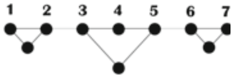
| **Pattern Type** | **Description of the State** | ***Linkographic* Configuration** |
|---|---|---|
| **Order** | ▪ A highly ordered pattern states an ongoing identical probability to move from one episode to another<br>▪ It reflects a state that a designer keeps performing the same actions ever then<br>▪ A pre-mature fixation effect of a certain idea may occur |  |
| **Structure** | ▪ A structured pattern delivers variable chances to develop an idea from one single utterance to another<br>▪ A diversification of various design ideas is experienced in the process |  |
| **Disorder** | ▪ A chaotic design situation reflects unrelated thoughts on the design situation<br>▪ It might cause total confusion and loss on the track of development. |  |

**Fig. 2** Different states of design reflect different patterns

Entropy is a measure of the state of disorder for any system. The aim of estimating entropy in information theory is to predict the probability of an event occurring. The objective of information theory is to investigate probability by establishing the number of possible sequences that can be created per single item. The sequence of an item is not taken into account in entropy theory but is necessary in information theory. Information theory is adopted to develop a quantitative approach to quantify the possible relations that are likely to occur at each item in the linkograph. Table 1 highlights the differences between information and entropy.

## Towards an Integrated Model to Describe the Design Process

There are various models that aim to study the design process. This paper explores two challenges: how to develop a quantitative tool to quantify the dependency relations in the linkograph and how to judge the dependencies in a qualitative way. This bi-modal methodology obtains insight from the application of deterministic information theory [1–3] and the paradigm of reflective practice process [4]. The aim is to provide an integrated approach to identify the formation of design ideas.

**Table 1** The differences between *information* and *entropy*

| | Information theory | Entropy theory |
|---|---|---|
| Structure | Nothing is better than those certain "sequences" of items that can be expected to occur | Sets constitute the main characteristics of the structure |
| Principles | Is about "sequences" and "arrangements" of items | Is about the "overall distribution" of kinds of item in a given arrangement |
| | The less predictable the sequence, the more information the sequence will yield, and the more remote its representation from order | The more remote the arrangement of sets is from a random distribution, the lower its entropy, and the higher its order representation |
| | A highly randomised sequence will be said to carry much information by the information theorist because information in this sense is concerned with the probability of this particular sequence | A randomised distribution will be called by the entropy theorist "highly probable" and therefore of low order because innumerable distributions of this kind can occur |

## *The Quantitative Model*

### Methods to Quantify Linkograph via Shannon Entropy

*Shannon entropy* aims to measure information associated with a communication source. Shannon suggested that the amount of information carried by a message is based on the probability of its occurrence [6]. The application of Shannon entropy to linkography, first proposed by Kan and Gero [7], aims to measure the degree of "probability" and "surprise" of the emergence of ideas in the design process, with a hypothesis that high uncertainty motivates the designer to explore the design space for more solutions (i.e. to become more creative).

The application of classical entropy to quantify linkograph has been argued by Shyan-Bin Chou [8]. To rectify the estimation process of Shannon entropy, his method adjusts entropy value with a *pattern–matching* factor to pick up the frequency of appearance of patterns into the estimation.

We argue that the application of Shannon entropy treats the linkograph in terms of *sets* and *networks* regardless of the sequential arrangement of occurrence of microelements that constitute the set according to *time* (the order of relations is according to the emergence of nodes). It looks at the overall distribution of sets in a global manner. Figure 3 shows two different linkographs that contain the same number of nodes (same n size of the system) but have different distribution. Both graphs sum up the same value of *linked* and *unlinked* relations giving an identical single entropy value:

As linkographs with different arrangements of relations reflect different design processes, the classical application of entropy is only associated with the quantity of links regardless of variation in distribution. The paradox still exists if we refer entropy into a single node—*can we address the right distribution of the created*

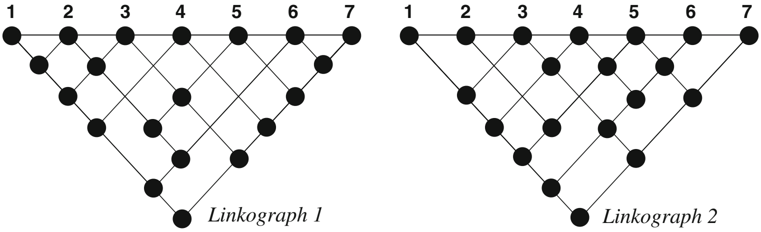


**Fig. 3** Two different linkographs give identical values of Shannon entropy. Processing Shannon Entropy: [H = $-$ ($p_{\text{linked}}$. $\log_2 p_{\text{linked}}$) + ($p_{\text{unlinked}}$. $\log_2 p_{\text{unlinked}}$)], The total number of possible relationships = n[(n-1)/2] = 7(6/2) = 21, Where n is the total size of the linkography (the number of nodes), The total number of "linked" relations in both graphs is = 13 → ≃ 61.9 %, The total number of "unlinked" relations in both graphs is = 8 → ≃ 38.1 %, **H** = $-$ [(13/21) × ($\log_2$(13/21)] + [(8/21)x($\log_2$(8/21)] = **0.2 bit/bits**

*relations on each node in the linkograph?* And, *can we reduce such a multi-level complex system to one single value?*

## A Synopsis on the Deterministic Information Theory

Titchener first presented the *DI* theory in 1998. A series of developments followed to obtain more accurate computation on *variable-length* strings of codes [1–3]. This theory presents *T-code sets* to encode the frequently occurring symbols of an information source, in which *messages* from any source can be coded into *alphabets* or *symbols* of *codewords* forming *a character string of information.*

Two coding alphabets can be characterised when coding the dependency relations in the linkograph: '1' for *linked* and '0' for *unlinked*. In this method, the linkograph can be transcribed into an alphabetical string of the two symbols and with a string length of all the relations that can be coded from a node to the others (length = n−1; where n is the system's size; the number of all moves in the linkograph). To code information at each design move, we look at its *preceding* and *following* relations and thus we can extract a character string and inspect its properties. As a rule of thumb, prior to DIT, however, characters cannot be processed directly without forming what is often called in the theory, "codewords".

T-code sets *form* or *decode* the codewords to reach an unambiguous serial transmission of information. A T-code set has two properties: it *augments* symbols to produce the codewords to construct the string, or *decomposes* the string to the microelements required to build it. In the augmentation process, *T-augmentation* can be seen as a string production algorithm, while *T-decomposition* may be seen as an algorithm that removes codeword boundaries in a hierarchical order. T-codes process the string by parsing the codewords in "forward" and "reverse" directions to compute its *complexity*, *information content*, and *entropy* providing three measures: *T-complexity*, *T-information* and *T-entropy*. The advantage of this method distinguishes different patterns of codewords and accounts for the position of each symbol in the string. The repeating appearance of codes, such as 000000000 or 111111111 is considered an extreme measure of the least uncertainty of serially communicated information.

Consider the following bits strings: 00000000001111111111 and 01001101100010101110. Despite containing a balanced number of 1 s and 0 s (giving the same Shannon entropy value), both strings deliver different T-code sets. Yet to the casual observer, it is clear that the second string contains a little more information than the first due to the complexity added by the differentiation of arrangement of symbols. In this case, a clear difference is shown if T-code method is processed on each string.

At the onset, DIT is introduced to this research area to emphasise the difference between classical entropy and T-codes. Shannon entropy assumes that a string is made up of *n-grams* (n-bit substrings) while T-codes assume that the string is a *codeword* from some sets and measures the number of weighted steps required to build that code [9, 10]. T-entropy looks at *codewords* while Shannon entropy looks at *n-grams* or *n-bits*. This method delivers accurate estimation of entropy that accounts for the arrangement of patterns of codewords at the level of micro items rather than classical entropy in the linkograph. It considers the position of any symbol in the string in the computation process which is crucially relevant to the context of sequential emergence and occurrence of moves in the design process. The following equations are principal to DIT:

$$\underset{i=1}{\text{T-complexity}}\ \text{is}:\ \mathrm{C_T(x)} = \sum \log_2(\mathrm{k_i} + 1)\ldots \tag{1}$$

*where k gives the number of adjacent copies of the T-prefix found at the position in the string when the T-prefix is identified for the first time.*

$$\text{T-information is}:\ \mathrm{I_T(x)} = \mathrm{li}^{-1}(\mathrm{C_T(x)})\ldots \tag{2}$$

*where* $li^{-1}$ *is the inverse logarithmic integral*

$$\text{T-entropy is}:\ \mathrm{H_T(x)} = \mathrm{I_T(x)}/||\mathrm{x}||\ldots \tag{3}$$

Generally, the richer the variety of symbolic arrangements, the more information the string carries. T-entropy predicts the patterns occurring in the string, where a high value indicates minimal repetition of patterns and means new patterns are appearing in the string with high unpredictability. A low T-entropy indicates a high repetition in the string—highly predictable information [2, 9–11].

*T-complexity* and *T-entropy* tend to “converge” as the string gets longer; the effect will be seen mostly with longer strings (hundreds or thousands of bits). Lack of accuracy increases with strings of less than 25 characters.

## Quantifying the Linkograph via T-Code Sets

This section introduces a computational tool to process the linkograph. Two methods are proposed: one operates on the level of individual nodes; the other subdivides the graph into subsets or sub-linkographs.

Method 1: Processing T-codes for individual moves. Our prime target is to compute *T-complexity* and *T-entropy* at each node in the linkograph, where both measures fluctuate throughout. In this method, the process to compute extracted strings can be carried out via one of three ways, which differ according to the direction of reading links (backward or forward):

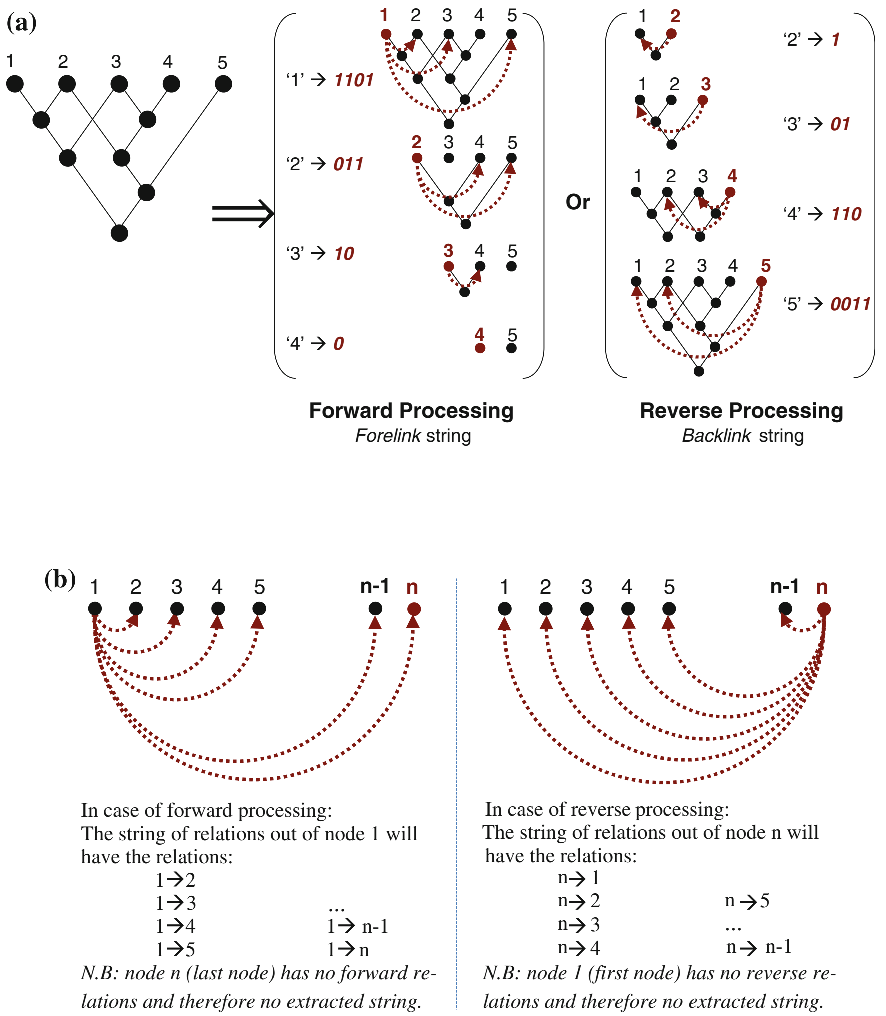
1. Extracting only *backlinks* string per each node: In this method, all relations are extracted in a reverse way to the emergence (from end to start). For example: node 5 has relations (*linked* or *unlinked*) with 4, 3, 2 and 1, but 1 has no back relation since it is the starting point.
2. Extracting only *forelinks* string per each node: In this method, all relations are extracted in a forward way, like the direction of growth in the linkograph (from start to end). For example: node 1 has forward relations (*linked* or *unlinked*) with 2, 3, 4, 5, … n, but n has no forward relations since it is the end point. Methods 1 and 2 are both synchronous to the emergence of links. See Figs. 4a and b to check the method per each reading direction.
3. Concatenating *backlinks* and *forelinks* together per each node: This is a third method based on concatenating both strings (*backlinks* and *forelinks*) per each node together in order to process one longer string at once (see Fig. 5).

Many methods can be suggested to extract a character string of information. The proposed method for a linkograph is to undertake the synchronous occurrence of nodes and to consider the direction of reading the relations. Figure 4a illustrates an application of “reverse” or “forward” methods of processing strings on a linkograph where the direction of reading makes a significant difference to the final results.

For example, in a linkograph with 5 nodes, if strings are extracted in reverse, node ‘1’ has no preceding relations, ‘2’ might have a relation with ‘1’, ‘3’ has two probabilities with ‘2’ and ‘1’, ‘5’ has four with all the preceding nodes. Generally the string out of ‘n’ has (n-1) string size, but in forward processing, the string out of ‘1’ will have four probabilities while the last point ‘5’ will have no probabilities with any following nodes.

In this hypothetical graph, node (15) has the following coded relations: *Backlink* string is 01111011011111 and *Forelink* string is 101110 and the concatenation is 01111011011111.101110.

Method 2: Processing T-codes for subsets or sub-linkographs. In this method, the linkograph is subdivided into a series of subgraphs. The subdivision can be made in two ways: *time rate* or *amount of nodes*. In each, *back* and *fore* strings can be computed similarly to method 1 (*individual* or *concatenation*). However, it should be noted that measures per frame must be normalised to the n-size of the sub-system in order to “relativise” the results of the subgraphs together. This is necessary to



In case of forward processing:
The string of relations out of node 1 will have the relations:

1→2
1→3
1→4
1→5
...
1→ n-1
1→ n

*N.B: node n (last node) has no forward relations and therefore no extracted string.*

In case of reverse processing:
The string of relations out of node n will have the relations:

n→ 1
n→ 2
n→ 3
n→ 4
n →5
...
n→ n-1

*N.B: node 1 (first node) has no reverse relations and therefore no extracted string.*

**Fig. 4** **a** An application of extracting the strings: forward or reverse. **b** Extracting the string in two directions: forward or reverse

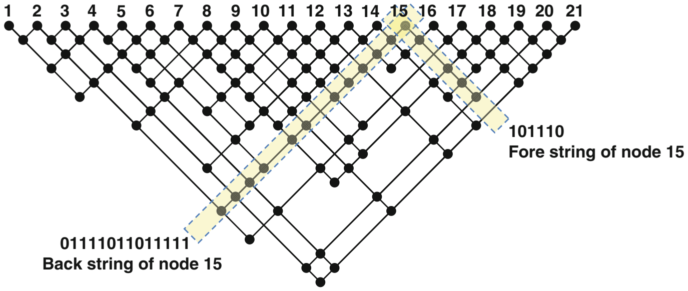


**Fig. 5** Processing T-code measures over a linkograph by concatenation

achieve the required accuracy but the result is conditional on being divided by the logarithm of the n- total number of nodes in each subset ($\log_2 n$). The calculation process starts by setting up the number of nodes (or occurrence rate) in a hypothetical window that slides across the baseline of the linkograph. The more a window displaces, the more nodes are included in the estimation process.

This method is built on the basis of two factors—*time* and *activity*—that must be described in the design process. We can inspect a certain activity that is relevant to time of emergence; such as looking at a certain sketch (design medium) that has specific start and end points. The emergence of action and the formation of concept are illuminated to find which actions are pivotally responsible for the emergence of a novel idea. The application of T-codes to linkography is looked at in a design case study in the earlier section "Towards an Integrated Model to Describe the Design Process" (Fig. 6).

## *Qualitative Judgements on Sketching Episodes*

The purpose of this approach is twofold. First, it sets up the *starting* and *ending* of what is often called a design *move* or *utterance*. Second, it aims to unveil cognitive mutual reflections with instantaneously externalised design artefacts all through various modes of representation (artefacts are the interim products such as sketches, 3D models etc.). As such, we aim to illuminate *stimuli* responses with respect to the sketching episodes; how they help the designer to break away from a *frame of reference* (which may lead to fixation) to proceed to a new one. The linkograph can then be drawn according to the judgements of dependency relations.

While *cross reflection* is an imperative key to understand the sketching interoperability with the mind, *instantaneous perception* is also a design process included in judging the sketching episodes. Tschimmel [12] suggests design as a *perception-in-action* process has five *nonlinear* intersected procedures: *the perception of the task, the perception of a new perspective, the perception of new semantic combinations, the perception in prototyping,* and *the perception of users' reactions*.

Hence, we define a *sketching episode* as a transformation in perception from one state to another while marking out the drawings prior to the design situation and to the interim reflection with the sketch still in progress.

Any sign that the designer has perceived a notion to break out of a frame of reference and shifted to another is considered an insight according to Akin and Akin's description [13]. A *creative insight* moves the perception completely to a different state that is independent from the current design situation. The design moves are hence coded based on two sets of contribution: actions preserve continuous reflections in the mind and actions defy continuous reflections.

Preserving reflection proceeds on the initial concept. It takes various forms of activity, such as *replication*, *redefinition* or *advanced incrementation*, in the same design state. Defying reflection introduces a new item to the current state. It has a

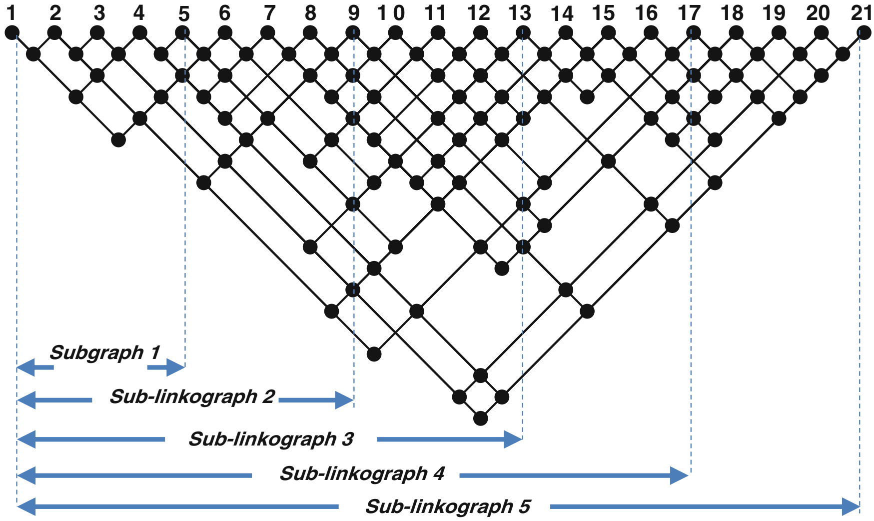


**Fig. 6** Processing the linkograph as a series of subgraphs

different taxonomy of actions that operate to change the design situation, e.g. *divergence, synthesis* and *reconstruction*.

*Sketching* is an act to perceive and reflect cognitive actions since it plays a central role in transferring notions in the mind into a design configuration. Goldschmidt [14] stated two types of sketching: type (1) aims to transform imagery into new forms of combinations and is considered as rational mode of reasoning; type (2) is sketching to generate new imagery of forms in the mind and is a non-rational form of design thinking. Our proposition primarily adopts all the preceding elements into developing a qualitative model to assess sketching episodes (see Fig. 7).

## An Empirical Case Study on Architectural Design Process [1]

We look first at the brief given to the designer, a chartered architect with 12 years' experience, in a design experiment at Bartlett School of Graduate Studies. The compatibility between the quantitative and qualitative results is in our attention to assess the adequacy of this model.

---

[1] The Phase of Earlier Initiation of Conceptual Ideas.

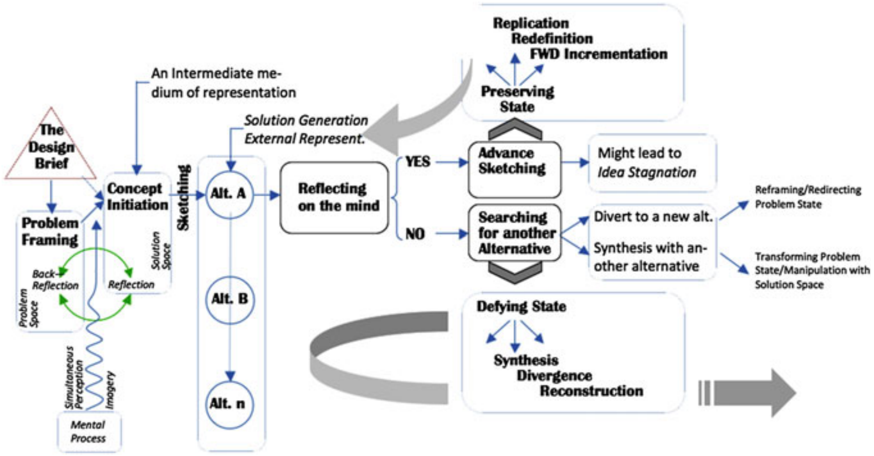


**Fig. 7** Qualitative judgements model to describe sketching episodes

## *The Design Brief*

The designer was asked to design a pavilion at *Expo-Shanghai*, 2010, presenting an image of her country from her own perception. The brief was left open-ended with no specific requirements or constraints to give the designer free rein. The conceptual idea was to be presented via any means of representation without any specific drawings or projections being requested and with no intrusion from the researcher. The process was video-recorded and the designer was asked to retrospectively explain the design idea for the serial order of three A2-size sketches produced in this session (Fig. 8).

## *Describing the Process, Products, and Significant Moves*

The process began by setting up elements to initiate the concept. These elements were considered to reflect the nature of Greece, where the designer's country. Eight elements were drawn in a diagrammatic form: (1) *Sea ripples: Circulation in and around lakes*; (2) *Built environment: boxes and light*; (3) *Complexity: steps, organic or orthogonal forms* (4) *Sun and sky;* (5) *Olive and lemon trees*; (6) *Colour scheme (blue, white, turquoise)* (7) *Rock and water*; *and* (8) *Strong shadows*.

Five independent pavilions were grouped in one site plan, each reflecting an element from the preliminary set (sometimes a synthesis of some elements together). The aim was to reflect impressions and spatial experiences to the users. Pavilions 1, 2 and 5 strongly relate to the preliminary set of design elements; the 'Stepped Forms', 'Olive Tree', and 'Rocks and Water'. Pavilions 3 and 4, 'The

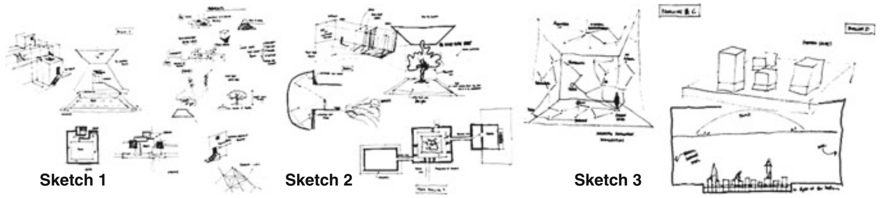


**Fig. 8** Snapshots of sketches from the design experiment

Tunnel' and 'Links and Cities', are considered as "creative leaps" on the flow of idea development. They have no strong links with the predecessors or even with the last pavilion (Rocks and Water). In spite of being produced serially pavilions 3 and 4 have no relations and their concepts are totally independent. In retrospective comments, the designer described those products came up while designing.

## *Transcription and Analyses*

The dependencies between design utterances are examined and coded according to the order of occurrence of interim artefacts (sketches). The designer wanted to implement *natural lighting* as a central concept. Artefacts 2 and 3 and the last slide (pavilions 1, 2, and 5) adopt roof skylights. Tracking this idea from one pavilion to another helps us to perceive the *lateral transformation* beyond the interim artefacts. The investigation was made on an overview of the whole but also on pairwise comparisons. The sudden "*absence*" and "*appearance*" of the *skylight* element encourages further investigation of the reasons beyond this and search into the possibility that creative insights might occur in parallel to the discourse.

Activities at points (52), (71) and (79) in the linkograph need consideration since no direct reflection is perceived on these particular moves. The sudden occurrence of node (55) is a change of state, when the designer diverted her train of thought from designing the olive tree pavilion to designing a new one. The tunnel pavilion is a spectroscopy on various colours and a gradual diffusion of artificial light. However, with *lighting* as a prime concept, the outcome is obviously different since the form and installation drawn for this pavilion are for various kinds of artificial lighting.

Point (71) is a paradigm shift, probably divergent and more than a mere change of state as in action (55). At this point, the designer suddenly diverted to draw a new element on the dispersion of Greek community around the world. The illustration of this idea, in pavilion 4, shows names of cities to which Greeks used to immigrate: Melbourne, London, Istanbul, etc. The designer called this pavilion "*Interactive Installation Immigration*". The names of cities are symbolised by dispersed nodes in a cubic form and connected with links. Colours and lights are embedded in ways that are different from in the preceding pavilions. The idea has intruded into the prevailing flow in this way.

Point (24) is a back-reflection (sketching back) to reframe the idea of *composition* and *complexity* in the preliminary set of elements. The designer drew a diagram of overlapping terraces and masses around a central atrium to assemble a *parti* (conceptual artwork) of the independent pavilions after designing the *stepped form* pavilion (nodes 17–23). Point (52) is also a back-reflection about irrational openings and balconies. The designer went back over this diagram to add to the first set of elements after designing the 2D-plan of the *Olive Tree* pavilion (nodes 31–51).

To summarise this proposition, coding relations in a linkograph is mainly about tracking reflections with respect to their order of occurrence in the context of the interim artefacts to construct an adequate linkograph. Table 2 presents the procedure and order of analysis in the proposed (Fig. 9).

## Results and Discussion

Starting with the quantitative model (T-code sets), the results of the two proposed methods to compute strings' entropy show the following:

1. The two methods of computation, *directional string* or *concatenation*, help to exemplify the significant nodes in two different situations. The former (directional *back* string) explains dependency relations on the instantaneous design situation in progress (before completion) and the latter (concatenated string *back* and *fore*) explains the design process after the completion of the whole design situation.
2. Correlations and comparisons between results can be achieved from these computational methods because all values are relativised to the "n-size" of the system. This is one of the main characteristics owing to deterministic information theory and T-code sets.
3. The numeric results provide measures that can be compared in correlation with the *interim design artefacts* or *cognitive activities* associated with the relevant moments.

In the whole process, node (53) achieves the highest degree of integration with a value of 6.09. The lowest degree of integration is delivered by node (25) and (71) with a zero value. At node (53), the designer redrew a diagram, titled "*irrational openings and balconies*", as a new element of design to generate new syntheses of form in her mind (that might be obtained again afterwards in the designing discourse).

At node (25), the designer put down the pen and glanced at the design brief. This happened between two sketching episodes: (24) in which she drew a new form into the set of elements "composition of terraces and overlapped masses"; and (26) when she shifted to design the "Olive Tree" pavilion by scribbling a diagram of shrubs, passages and green landscape. Node (25) is a disconnection on the train of thought.

**Table 2** The integration of qualitative and quantitative approaches in one model

| # | Phase | Detail |
|---|---|---|
| 1 | Transcription | Transcribing the design activities accordingly to the time of occurrence |
| 2 | Interim artefacts | Setting out the interim outcomes in the order of emergence (design artefacts, sketches, etc.) |
| 3 | Identification of design episodes | Processing the qualitative/cognitive model by identifying each design move according to the notion of "reflection-in-action" |
| 4 | Coding process | Coding the externalised drawings in relation to the pertained cognitive activities. The dependency relation is looked at through two angles<br>(1) The relation between each artefact and the preliminary set of design elements (the interim artefacts and initial conceptual elements)<br>(2) Pair-wise comparison between sequential pairs of drawings. This is to investigate the lateral transformation and search for any sudden insight that might occur in the prevailing flow |
| 5 | Reflection-in-action | Finding the "reflections" and "back-reflections" amongst the sketches to classify a hierarchy of the products |
| 6 | Linkography | Drawing the linkograph |
| 7 | String computation | Processing the quantitative model (T-code measures)<br>- Setting out a matrix of relations<br>- Processing the T-code algorithm |
| 8 | Archiography | Drawing the archiograph (another representation of the linkograph but it reflects the relations in a clearer way) |
| 9 | Comparisons | Comparing the archiograph with the interim artefacts |
| 10 | Correlations | Checking out if correlations exist between the quantitative model and qualitative judgements |

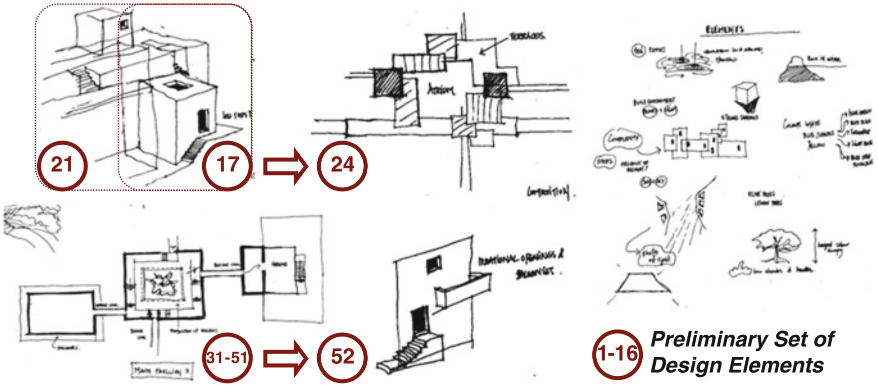


**Fig. 9** Snapshots of examples of sketching episodes and idea transformation

Likewise, node (71) is a *rupture* in the running activity—sketching episode (70) is about adding quotations on 2D-section-the *tunnel* pavilion—where she drew suddenly a diagram of a new element entitled *links* on a different sheet. Despite delivering zero connections with preceding actions, point (71) has strong linking relations with following actions until the completion of pavilion 4.

Subsequently, three levels are denoted to inspect the structure of design hierarchy: (1) the preliminary level of concept initiation (an intermediate medium); (2) a level of continuous forward sketching and externalisation of ideas; and (3) a level of back-reflections to generate imagery of forms in the mind (represented by adding new design elements to the preliminary set of concept initiation). Figure 10 illustrates the distribution of sketching artefacts (snapshots) across these three levels.

Point (31) achieves the highest T-complexity and T-entropy values via the concatenation method with values of 17.98 and 0.8. The designer shifted the sketching episode from two different projections (3D perspective to 2D plan). This happened while designing the "olive tree" pavilion in order to further enhance and describe the concept in detail. Exchanging thoughts "*back and forth*" and switching the idea between different media (drawing projections) is a clear example of the mutual reflection between the design media and the mind (between two cognitive structures—internal and external). The integration value for point (31) is 2.98, which is the lowest value relative to the whole linkograph. However, (31) reflects a high degree of understanding and developing the idea through different media of representation. In this discourse, integration is inversely correlated with string measures (T-complexity and T-entropy). See the numeric values of some significant nodes in Table 3.

The lowest degree of T-complexity and T-entropy is delivered by node (49) with values 7.25 and 0.21. At this node, the designer traced over an existing sketch but just with a thicker pen to highlight a plane cutting through the tree pavilion. The utterance is not crucial to the development but an emphasis on the activity.

Figures 11a, b overlay methods of computing strings on the *archiograph*. A change of state is often observed in *integration*, *T-complexity* and *T-entropy* values on the switches between different media (sketches) on how the idea utterances are synthesised in the process. *Archiography* is a process to represent dependencies amongst nodes. It was developed to illustrate relations in a way to avoid dense clusters not on the baseline. It looks at graph connectivity rather than clusters of nodes. *Archiographer*© software was developed to build up the relations and draw the archiograph.

Pavilions 3 and 4 are semi-disconnected from the whole linkograph and the relations are similarly distributed amongst nodes. Therefore, no significant fluctuation is seen with T-codes compared to integration values.

The correlation between syntactical measure (integration) and T-code sets (complexity and entropy) are examined in detail in a previous study by the author. El-Khouly and Penn [15] study the correlations on various systemic graphs that share multi-level complex properties such as '*linkographs*' and describe the

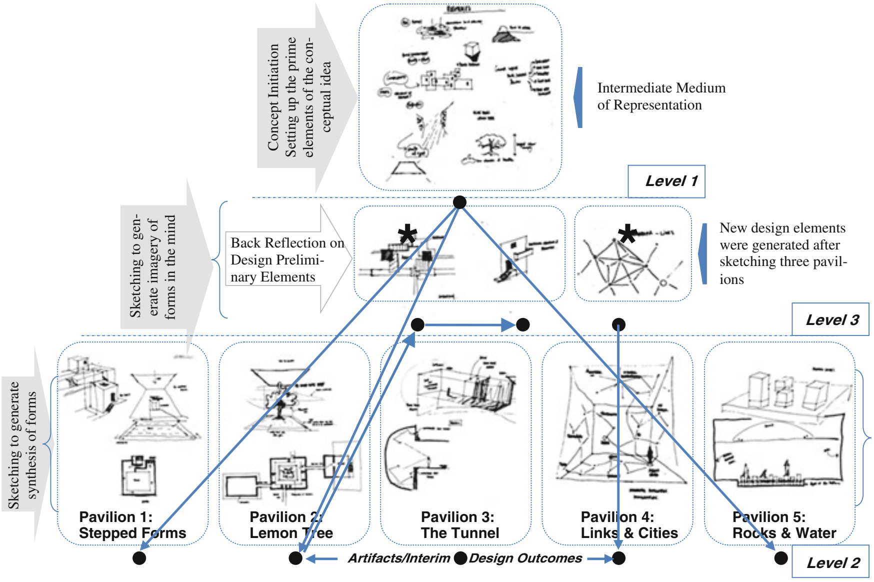


**Fig. 10** A hierarchical classification of sketches based on qualitative judgements: (reflections) and (back-reflections)

structure of state via different spectrum of patterns; *orderliness* versus *disorderliness*.

## In Conclusion

This study outlines a qualitative approach for describing design hierarchy. In this discourse, it discusses the compatibility of merging quantitative and qualitative models. Forming non-rational syntheses serves to introduce new boundaries to the design programme and also encourages the exploration of new functions that have not been explored before. Throughout the proposed model, we can detect a multi-levelled concept that has been conducted through a design process, to see how the design serves it goals and how it sets new goals. A multi-level design concept is evolved through:

1. An intermediate medium of representation for the concept initiation.
2. An execution process of the idea.
3. A retrospective reflection on earlier thoughts.

**Table 3** Various quantitative measures of some significant nodes

| # | Extracted concatenated strings | Integration values | Dynamic backlink entropy | String computation | | |
|---|---|---|---|---|---|---|
| | | | | $C_T$ | $I_T$ | $H_T$ |
| 25 | 000000000000000000000000.1000000000000000000000000000000000000000000000000000000000000 | 0 | 2.45* | 7.93 | 20.77 | 0.24 |
| 31 | 001010000000001100100000011101.1111111111101100000110000000000000100000000000000000000 | 2.98 | 3.45* | 17.98 | 69.38 | 0.80 |
| 49 | 000000000000000000000000000100000000000000000000.0000000000000000000000000000000000000 | 4.33 | 0.72* | 7.25 | 18.21 | 0.21 |
| 52 | 000100000000000010001001000000100000000000100000001.1100000000000000000000000000000000 | 3.02 | 2.73* | 13.63 | 46.29 | 0.53 |
| 53 | 0000000000000000000000000000000000000000000000000001.000000000000000000000000000000000 | 6.09 | 2.64* | 8.09 | 21.38 | 0.25 |
| 71 | 0000000000000000000000000000000000000000000000000000000000000000000000.100001000000000 | 0 | 2.70* | 9.32 | 26.37 | 0.30 |
| 79 | 000000000010000000000000000000000000000000000000000000000000000000000000000000.0010001 | 3.5 | 1.78* | 10.11 | 29.74 | 0.34 |

* As has been explained earlier in this paper, *Dynamic Entropy* values do not reflects the actual entropy of a certain single node; rather it is a logarithmic summation of a group of nodes

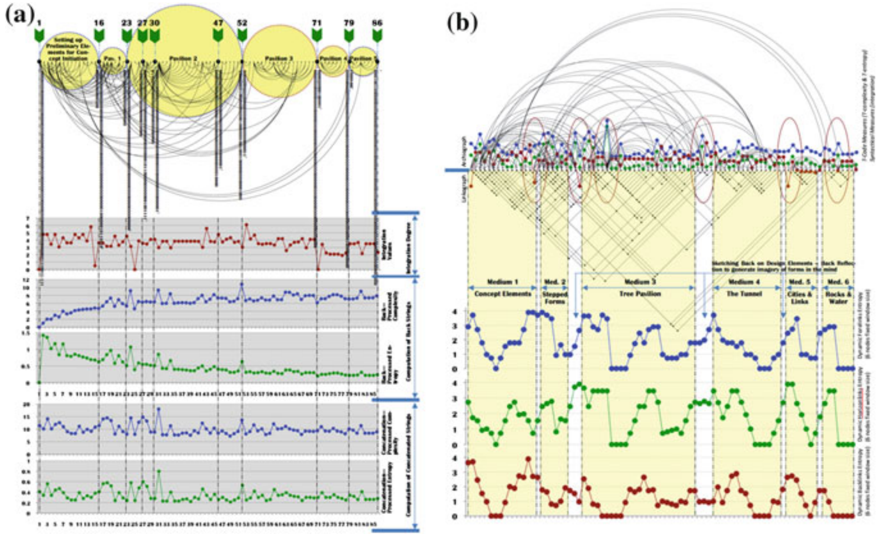


**Fig. 11** **a** Overlaying string computation graphs on the archiograph (*left*) and **b** Overlaying T-code sets, integration, and dynamic entropy (*right*)

The observed complexity of idea transformation is owing to the fact that there are no isolated concepts in ordinary thoughts except those rare but remarkable breakthroughs.

Despite the meaningful interpretations that can be made by implementing the Shannon method, its disadvantage is that it estimates entropy in a global manner since the process is done *layer-by-layer*. The aspiration of this study is to append another measure to acquire information from a non-linear system such as linkography. T-code sets provide an objective tool to inspect entropy in a multi complex design process, working in integration with qualitative judgements on the design outcomes.

**Acknowledgments** I wish to thank *Dr Alkistis-Zoi Skarlatou* for her participation in the design experiment providing this study with valuable insights. I also acknowledge the contribution of *Dr Ulrich Speidel* (Department of Computer Science, University of Auckland, New Zealand) who has provided clear directions on the *deterministic information theory* and *T-code sets*. I am grateful for his generous advice and comments. I am also indebted to *Consultant Engineer Mohamed Abdallah* (Thrupoint, USA) for his thoughtful suggestions and technical support to reach an accurate computational model for this proposition. Finally, my gratitude goes to *Prof. Alan Penn* (Bartlett School of Graduate Studies, UCL) for his constructive supervision, critical and inspiring throughout and to the school for providing the facilities to establish this research. I take full responsibility for any shortcomings in this paper.

## References

1. Titchener M (1998) Deterministic computation of string complexity, information and entropy. International symposium on information theory, MIT, Boston, pp 16–21 Aug 1998
2. Titchener M (2004) The practical measurement of information and entropy, Online tutorial: http://tcode.auckland.ac.nz/tutorial.html
3. Titchener M, Nicolescu R, Staiger L, Gulliver A, Speidel U (2005) Deterministic complexity and entropy. Fundamenta Informaticae 64(1–4):443–461
4. Schön DA (1983) The reflective practitioner: how professionals think in action. Harper Collins, Basic Books, New York
5. Goldschmidt G (1991) The dialectics of sketching. Creativity Res J 4(2):123–143
6. Weaver W, Shannon CE (1949) The mathematical theory of communication. University of Illinois Press, Urbana, Illinois
7. Kan JWT, Gero JS (2005) Can entropy indicate the richness of idea generation in team designing? In: Bhatt A (ed), CAADRIA ’05, New Delhi: TVB, Vol 1, pp 451–57
8. Chou SB (2007) A method of evaluating creativity in linkography, 10th QMOD conference of quality management and organizational development. Our Dreams of Excellence, 26, Helsingborg, Sweden, pp 18–20
9. Titchener M (2008) An introduction to T-codes. Online tutorial: http://tcode.auckland.ac.nz/∼mark/T-codes%3A%20Intro.html
10. Speidel U, Titchener MR, Yang J (2006) How well do practical information measures estimate the Shannon Entropy, CSNDSP, Patras, Greece, pp 19–21 July 2006
11. Speidel U (2008) T-code and T-information theory, Online tutorial: http://citr.auckland.ac.nz/∼ulrich/research.html
12. Tschimmel K (2010) Design as a perception-in-action process. In: Taura T, Nagai Y (eds) Design creativity. Verlag, London, pp 223–230
13. Akin Ö, Akin C (1996) Frames of reference in architectural design: analysing the hyper-acclamation (Aha!). Des Stud 17(4):341–361
14. Goldschmidt G (1994) On visual design thinking: the vis kids of architecture. Des Stud 15(2):158–174
15. El-Khouly T, Penn A (2012) Order, structure and disorder in space syntax and linkography: intelligibility, entropy, and complexity measures. In: Green M, Reyes J, Castro A (eds) Eighth international space syntax symposium (SSS8). Pontifica Universidad Católica, Santiago De Chile, pp 1–22

# A Representational Scheme for the Extraction of Urban Genotypes

**Sean Hanna**

**Abstract** A representational scheme is described for cities, which uses the spectrum of the graph derived from a network of streets. This is of a sufficiently high dimensionality both to capture information of the city structure and to allow different representations of urban types to be extracted from it. It is proposed that a machine can extract the 'genotype' description that classifies a given group of cities. Results demonstrate that these capture morphological relationships between cities, and reveal correlations between these and a city's geographical location. This has implications for our understanding of design processes and the modelling of creativity, in that the final representation can be made autonomously by the computer, rather than predefined by a priori standards.

## Introduction

Our folk psychological concept of creativity places nearly all emphasis on output—the act of producing something, or at least conceiving of its coming into being. While this is undoubtedly essential, it is not the whole picture, and may even be less important than a creative agent's ability to interpret what already exists in the world. Accounts of creative insight [1, 2] emphasize 'seeing as', and models of the wider social structure [3] account for creative inspiration as a recognition and interpretation of a domain of works produced by others. Under the assertion that a an understanding of mechanisms for the interpretation of existing artefacts is essential for the understanding of how design and creative thought occur, this paper demonstrates a mechanism by which this is feasible by machine, by allowing the computer to derive an autonomous feature description via

S. Hanna (✉)
University College London, London, UK
e-mail: s.hanna@ucl.ac.uk

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_26,

dimensionality reduction of a much higher dimensional representation. The format for this is described for cities, using the graph spectra of large urban plans.

The case may be made for the importance of interpretation within the computational environment used to aid design thinking. The dominant stream of thought in the development of computer-aided design (CAD), building information modelling (BIM) and related computational design tools rests strongly on a basis of standardized, symbolic representation.

Historically, this has been embedded since the earliest CAD programs [4] in a division between the internal, symbolic processing to manipulate the model, and the visible graphic display on the surface. New computational environments, such as current parametric tools, are often touted as flexible [5] for the superficial ease in manipulation of geometry within well defined parameters, but their use require the specification of a highly structured set of associative relations between model elements, which actually increases the importance of the symbolic level. In storing and transferring CAD data, the strategy is to develop common file formats and standards by which known geometrical primitives and data structures can be encoded.

While these may be effective in ensuring clarity of communication and efficiency of information storage, they are at odds with much of what is important in the modes of thought essential in design. In the first instance, there is the issue of cognition in general. The physical symbol system hypothesis [6], that intelligence itself is a manipulation of discrete symbolic tokens, underlies much of classical artificial intelligence (AI), but it is precisely the arbitrary assignation of tokens to phenomena that is revealed as problematic in criticisms of AI from the varied 'embodied' Heideggerian [7], computational [8] and dynamical systems [9] camps. Indeed, it would seem that the act of autonomously assigning a meaningful representation to an observation is a basic requirement for the definition of a cognitive act. In the second instance, the particular nature of the creative process that occurs specifically in design involves an essential flexibility of representation. The useful ambiguity of the sketch is one obvious and common instance, but the phenomenon of creative discovery in the form of the paradigm shift [10], the punch line [2] or common insight [11] can almost invariably be described as a point of reinterpretation—of seeing 'something' as 'something else' for the first time. This is an act that explicitly runs counter to the basic principles of standard representations, and something not currently well supported by existing tools.

Originally in the context of urban morphology, Hillier and Hanson [12] described the ability of a culture to propagate the features of its built environment from generation to generation with a biological analogy. The particular form of a given city type can be considered as a genotype, but while in nature this is embodied and transmitted by the fixed code of DNA, in human cultural evolution there is no such mechanism. Instead, the cultural equivalent is an 'inverted genotype' [12, p. 43], which is identified and copied from the persisting examples of built form by each subsequent generation. This violates the 'central dogma' of molecular biology [13] which allows information to flow only from the code to the resulting product (in this case, an organism), but in stressing the act of independent interpretation of existing artefacts and the absence of a standard code, it is a far better match to theories of creative thought. The requirement for such a process, in lieu of a coding mechanism, is a mechanism for extraction of this genotype that can be performed independently by each agent. This mechanism must be able to determine such a genotype by recognizing the similarities and common features across cities that are only evident when these are seen as a group.

## Aims

This paper demonstrates a means by which the urban plan can be represented numerically in such a way that genotypes may be extracted from the data. To do so, it has several aims:

1. It is essential that the data be of sufficiently high dimensionality to allow the possibility extraction of a lower dimensional subspace. It will show that graph spectra provide a representational scheme that satisfies this condition, both for entire cities and for local neighbourhood areas.
2. The assertion that genotypes are an element of cultural propagation suggests that geographical proximity should be revealed in cultural similarities expressed in morphology, and therefore in the representation. While factors such as climate are also potentially relevant to urban form, particularly in the comparison of cities north to south, some correlation between east-west distances and differences in the representation should be demonstrable.
3. In proposing such a mechanism specifically as an alternative to standardized representation, the paper will aim to show that different genotypes can be extracted depending on the data set (domain). While corresponding to different feature sets, each of these should still be useful in the plausible classification of cities.

## Method: Extracting the Spectra of Segment Graphs

Axial maps [12] of cities are used in this work, but for all practical purposes these may be considered as maps of road centre lines [14], with each straight road or road segment given by a single linear element. Each map may be considered as a graph, with each node corresponding to one of these lines and connections to each road line it intersects (see Fig. 1).

The spectrum is given by the ordered set of eigenvalues of this graph's connectivity matrix. Graph spectra have been shown to reliably capture differences between the overall structure of compared graphs, in that they are nearly unique to a particular graph [15], and the measured distance between spectral vectors increases with progressive changes to a given graph [16]. The standard method for deriving the spectral representation, as given by e.g. Luo et al. [17], Robles-Kelly

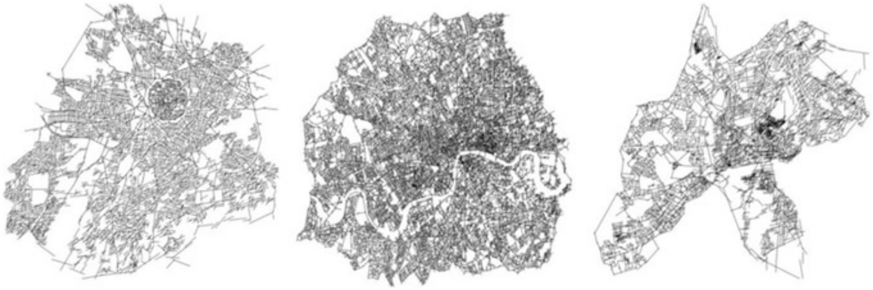


**Fig. 1** Axial maps display street networks. Graphs in the data set range in size from 46 nodes (Rio de Contas) to 79,740 nodes (Sao Paulo). Displayed above: Nicosia (16,932 nodes), London (58,697 nodes) and Nottingham (13,728 nodes)

and Hancock [18], and others, begins with an unlabelled, unweighted graph given by a set of nodes $V$ and edges $E$. The spectrum may be derived from an adjacency matrix representing all connections between nodes such that:

$$A(i,j) = \begin{cases} 1 & \text{if}(i,j) \in E \\ \text{or} & \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Eigenvalues $\lambda$ and eigenvectors $\varphi$ for $A$ are found by solving for

$$A = \Phi \Lambda \Phi^{\mathrm{T}} \tag{2}$$

where the matrix $\Phi = [\varphi^1 \mid \varphi^2 \mid \ldots \mid \varphi^{|V|}]$ contains the eigenvectors as columns and the matrix $\Lambda = \mathrm{diag}(\lambda^1, \lambda^2, \ldots, \lambda^{|V|})$ contains the eigenvalues as diagonal elements. The spectrum is then defined as the set of ordered eigenvalues

$$S = \left\{\lambda^1, \lambda^2, \ldots, \lambda^{|V|}\right\}. \tag{3}$$

To form a feature vector, eigenvalues are sorted in an order that must be constant across all data. This has been done in practice either by absolute magnitude such that $|\lambda^1| > |\lambda^2| > \ldots > |\lambda^{|V|}|$ [17, 18], or by actual value such that $\lambda^1 > \lambda^2 > \ldots > \lambda^{|V|}$. As the former method can lead to ambiguity in cases where identical magnitude eigenvalues appear under opposite signs, the latter method is used here.

## Angular Weighting in the Spectrum

The above procedure captures graph topology, and is similar to methods shown as effective in image retrieval and other applications. In recent Space Syntax work, particularly in the local configuration of urban sub-regions, the geometry of the actual plan and angle of street segment intersections has also been found to be relevant, as given by measures such as choice [14] and angular radius. These employ a weighting of graph connections in proportion to the degree of turn between joining streets, such that a greater angle between street segments implies a greater effort of turning when moving through the space and therefore incurs a greater cost. This cost of direction change can range continuously from zero for no change in direction to a maximum at a 180 degree about face.

For the representation of individual neighbourhood regions within the city, a similar approach is used here to weight the adjacency matrix with an inverse of this cost, such that nodes with the strongest geometric connection—segments of a straight line—are weighted highest, and values decrease with greater angle of turn. Whereas the binary matrix in (1) represented any connection with a value of 1 and all unconnected nodes with 0, an alternative value for connected nodes is derived from the angle between the unit vectors of the street segments:

$$A(i,j) = \begin{cases} \mathrm{a}\cos(i_{vect} \cdot j_{vect})/\pi \\ \text{or} & . \\ 0 \qquad \text{otherwise} \end{cases} \tag{4}$$

This results in a maximum $A(i, j) = 1$ when $i$ and $j$ are part of a single, straight road line, and decreases toward zero as the turn angle becomes more acute. The algorithm that determines whether $(i, j) \in E$ does so by searching for common segment end points, and therefore finds segments to be self-connected, however for $i = j$ (the diagonal of $A$), these links represent a 180° turn back onto the same segment, are calculated by (4) as $A(i, j) = 0$.

## Qualities of the Spectral Representation

To test the effectiveness of the graph spectrum as a basic representation in capturing relevant features of city form, a data set was used consisting of 152 cities around the world. This dataset is a compilation of a number of separate collections of axial maps previously drawn by Space Syntax researchers following identical principles as in [12], and consists of a representation of each street network as a set of linked line segments. As a union of cities originally chosen for prior, unrelated research, their locations are not uniformly distributed, but sample most major regions with notable exceptions for large parts of Africa and Asia for which no such maps were available. The size of the sampled cities, and therefore their graphs, vary greatly from approximately 50 to 80,000 streets. Each city is labelled by geographical location, both by latitude and longitude coordinates, and by a tag indicating one of five broad geographical/cultural regions: NOR (English speaking North America: all samples from the USA), LAT (Latin America, including Mexico and South America), EUR (Europe), ARA (Arabic speaking countries:

actually extending from Africa through Western Asia including Iran), and ASP (Asia-Pacific, also including New Zealand).

In line with aim (2) in section "Aims", the following sections examine the degree to which regional properties can be determined by comparative analysis of the spectra both of entire cities and by individual neighbourhood areas. To do so, it is necessary to ensure spectral vectors of the same dimensionality. For whole cities, the unweighted adjacency graph of full axial lines is used, and spectra are maintained at a constant dimensionality of $d = 100$ by 'cropping' all but the 100 highest magnitude eigenvalues (after [17]). For neighbourhood sub-graphs within a city, samples are taken of the 200 nodes closest in distance to a given location in the city, to maintain a constant dimensionality of $d = 200$. Street segments (the line segment between any two intersections) are used for finer granularity and the angular weighted matrices are used. Further details of the method and following analyses can also be found in different contexts [19, 20].

## Whole Cities and Their Geographic Location

Each city is initially represented by a vector in 100 dimensional space. While difficult to visualise in its entirety, a rough approximation of the relationship between the spectra in this space can be achieved by basic dimensionality reduction to a visible subspace. Principal components analysis (PCA) was used first to do so. Figure 2 plots all 152 cities in the set by their first and second principal components, with the shade of each point indicating its longitude. Although there is a considerable degree of overlap between regions, the identities of the regional subgroups NOR, LAT, EUR and ARA are indicated as general clusters in the diagram. Only the ASP group, with the widest geographical and morphological variance, is unidentifiable with an approximate region in the two dimensional plot.

The first principal component in Fig. 2 appears to describe much of the variation in longitude of the data set and can be used in isolation to estimate a lower bound on the degree to which individual city spectra are determined by geographic location. Figure 3 plots the first principal component of all city spectra (vertical) against their longitude (horizontal), revealing a low but not insignificant degree of correlation ($R = 0.32$) between longitude and spectral component.

The assumption of a single, linear component to capture the geographical variance of the set is problematic, however, due to the geographical distribution of the cities themselves. Most of the regional subgroups consist of cities in a band through the temperate regions of the northern hemisphere, except for LAT, which is largely in the south. At this point no effort has been made to separate out features potentially affected by climate or other cultural factors within this group, and the resulting north-south axis of the Americas potentially complicates the east-west distinction we are looking for. When northern hemisphere cities only are used (see Fig. 3, right) correlations improve to a moderate $R = 0.46$ with linear

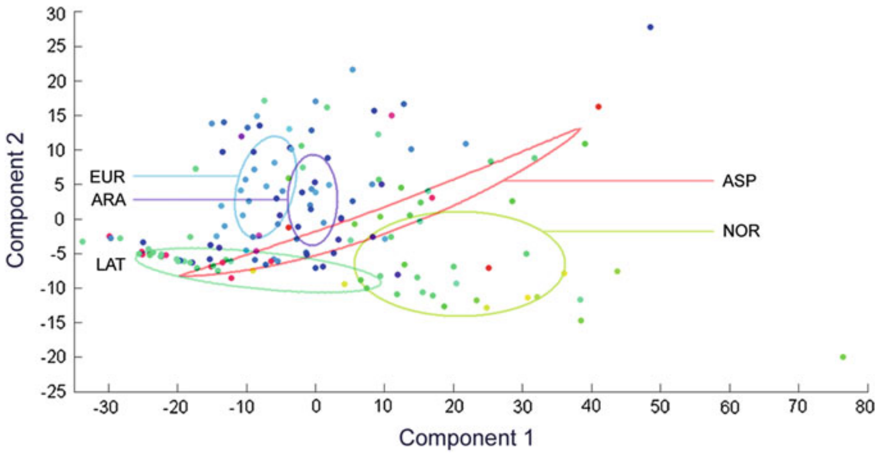


**Fig. 2** First two principal components of city spectra

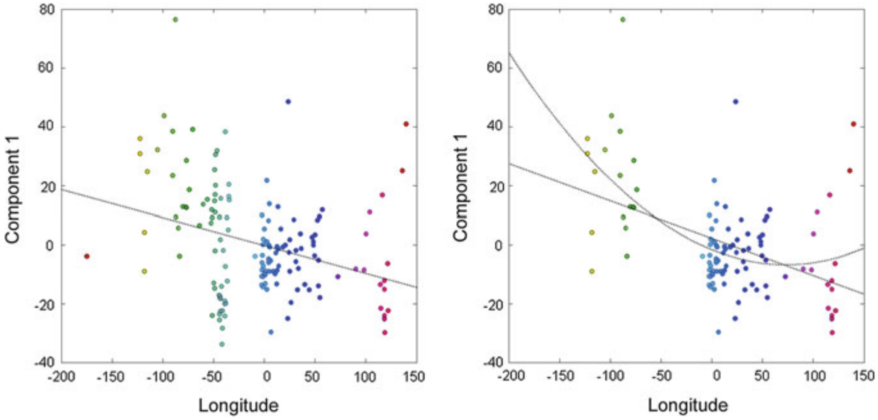


**Fig. 3** First principal component of city spectra (*vertical*) against longitude (*horizontal*). Correlation is improved for the northern hemisphere only (*right*)

regression, or R = 0.55 if a quadratic function is used. Within the American continents, latitude is revealed as more significant than longitude, as displayed in Fig. 4. Here, the first component for LAT and NOR cities is plotted against their latitudinal location, with a moderate correlation of R = 0.44.

To further disaggregate the clash between north-south and east-west distinctions, the principal components may be plotted against a two-dimensional world map (see Fig. 5). The method identical to a mapping method in biological genetics, in the work of Cavalli-Sforza, Menozzi and Piazza [21] to plot the principal components of gene frequency distributions, which have been used to

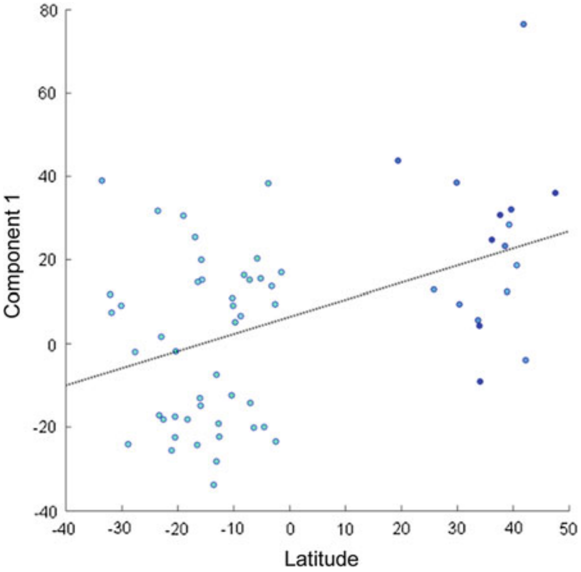


**Fig. 4** Moderate correlation (R = 0.44) between the first principal component of city spectra (*vertical*) against latitude (*horizontal*)

infer patterns of prehistoric settlement and migration. Here, it is the underlying data from which the cultural genotype will be formed that are being mapped. For each component, a surface is fitted to the distribution of points as an approximation of how the city spectra vary continuously across the world, using a method described in Shepard [22]. Like genetic data, the variance of city samples in a local region rarely allows for smooth interpolation, so instead the 'expected' value of the surface is calculated at each point as a weighted average of the data, with the weight based on relative distance to observed data points [21, p. 45, 22]. This gives a clearer picture of the locally dominant values in the regions with the highest density of samples and a better indication of the variance of cities by their location.

With some correction of seven cities identified as outliers, the resulting surface displays a marked correlation between expected values and actual spectra. The first principal component correlates with a coefficient of R = 0.61 across all regions of the globe, nearly doubling from the original linear longitudinal correlation of 0.32. The values of the map surface may also indicate broader regional relationships between city forms, for example the similarity between Western Europe and the eastern cities of South America, or the similarity between Mediterranean and Iranian cities. More research would be required to determine whether these are significant.

## *Distinguishing Individual Neighbourhoods Within the City*

The relationship between the spectral representations of smaller scale samples of urban plans was investigated under two working hypotheses: that (a) the large scale geographical distinctions seen in section "Whole Cities and Their

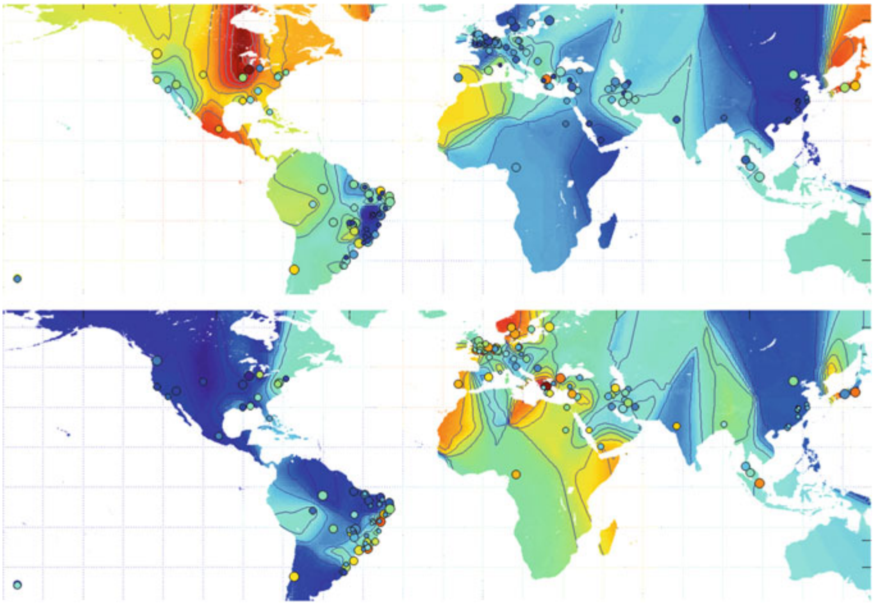


**Fig. 5** First (*top*) and second (*bottom*) principal components of city spectra fitted to the world map

Geographic Location" would be evident in the general variance of the smaller scale samples, with neighbourhoods within a city expected to be more alike than neighbourhoods between distant cities, and (b) the samples within a particular city would display a more local pattern of variance, correlating with e.g. distance from the centre. Two cities of similar size but differing character, London and Seattle, were fully sampled into spectral neighbourhoods of $d = 200$, by constructing local area graphs of the closest 200 street segments to a particular point in a city.

Principal component analysis is again used to visualise the relative placement of the spectral representation of neighbourhood samples (see Fig. 6), revealing the variance both within and between the two cities. For legibility, only 10 % of all segment neighbourhoods are shown, providing 5,870 samples from London and 6,390 from Seattle. London (upper cluster) and Seattle (lower) each occupy a relatively continuous cloud of points, with gradual distinctions between neighbourhood spectra revealed by the position of each. Both cities can be seen to occupy clear and nearly distinct zones in the feature space—far more so than the broad geographical regions in Fig. 2. Primarily separated by their second component, it is possible to identify entirely from the spectrum the city from which nearly all of the local samples are drawn; using first two components as plotted, 89.5 % of samples are clearly classified in their correct city by a simple linear discriminant function. Treated in the full feature space of 200 dimensions, a linear discriminant performs marginally better in correctly partitioning 91.8 % of the samples.

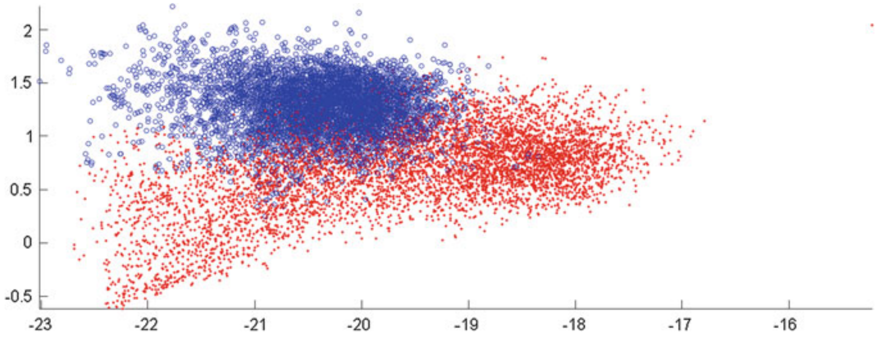


**Fig. 6** Sampled spectra from London (*top cluster*) and Seattle (*bottom cluster*) plotted against the first two principal components of the combined set. Each city occupies a distinct region of the feature space

In addition to global scale geographical distinctions between cities, it was hypothesised that more local differences within cities would be captured by the spectral representation, such as the change in urban morphology over time. The data for the age of each street was not available in the data set, but a very crude approximation may be made by assuming a generally radial growth over time from an original central city core. In London, where geography allows growth in all directions north and south of the Thames, the geographical mean of all segments was taken to be this core, while in Seattle, where its coastal location only permits eastern expansion, the centre is taken to be the westernmost point of the mean east-west axis. The position of each street segment's local spectrum on the city's first principle component is indicated by the shade of that segment in Fig. 7. Both cities reveal a general cluster of patches at the extreme end of the component in the city centre, with that of London in the geographical centre of the city, and that of Seattle at the shoreline to the left of the map.

The first component of each city is shown (see Fig. 8) plotted against distance from these geographical centres. Although only a very crude approximation, a reasonable to strong correlation can be seen, with a coefficient of $R = 0.521$ ($R^2 = 0.271$) for London, and $R = 0.752$ ($R^2 = 0.566$) for Seattle. It is difficult to place a realistic bound on the effectiveness of this method to reveal the actual time of construction of each neighbourhood, but with properly labelled data, including approximate dates or other factors, this might be more clearly determined. The principal component at present is based entirely on internal variance, and it is almost certain that an approach using supervised learning to derive a more effective function would yield better results. This will be used in the following section to explore how the machine might independently extract a plausible urban genotype from labelled sample spectra.

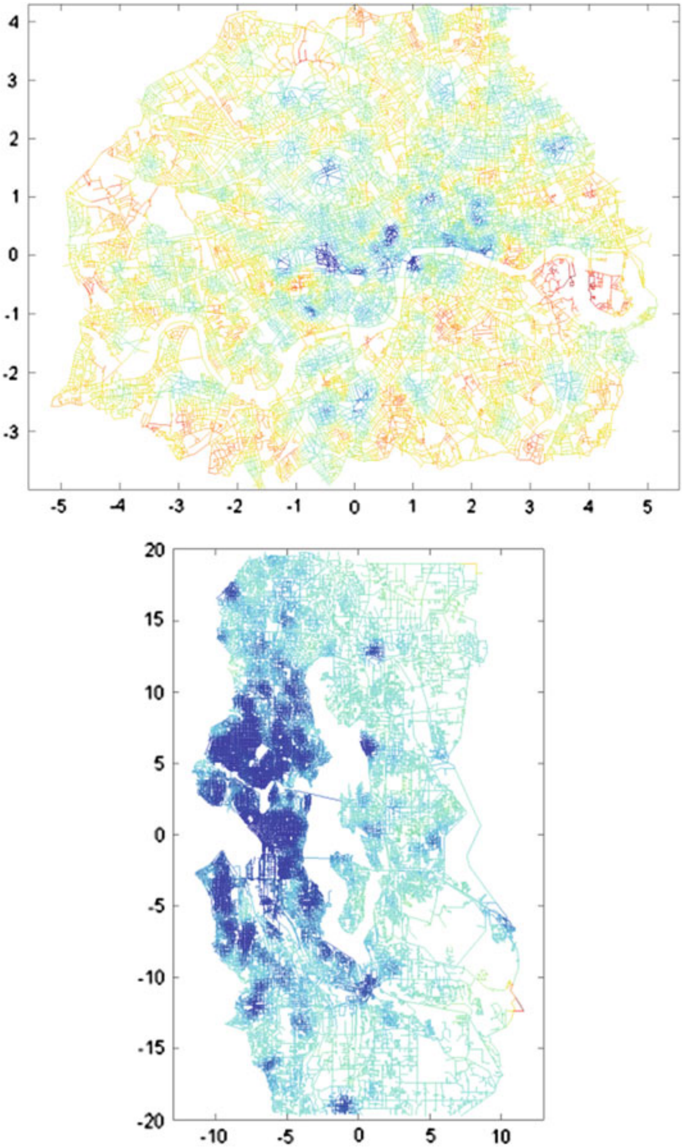


**Fig. 7** The first principal components of $d = 200$ subgraphs of London (*left*) and Seattle (*right*) plotted on the city map

## Extraction of the Type from Classes of Examples

Beginning with the raw data supplied by the graph spectra above, we are concerned with the possibility of extracting a genotype, or description of a particular set of morphological features that uniquely identify a particular city or class of

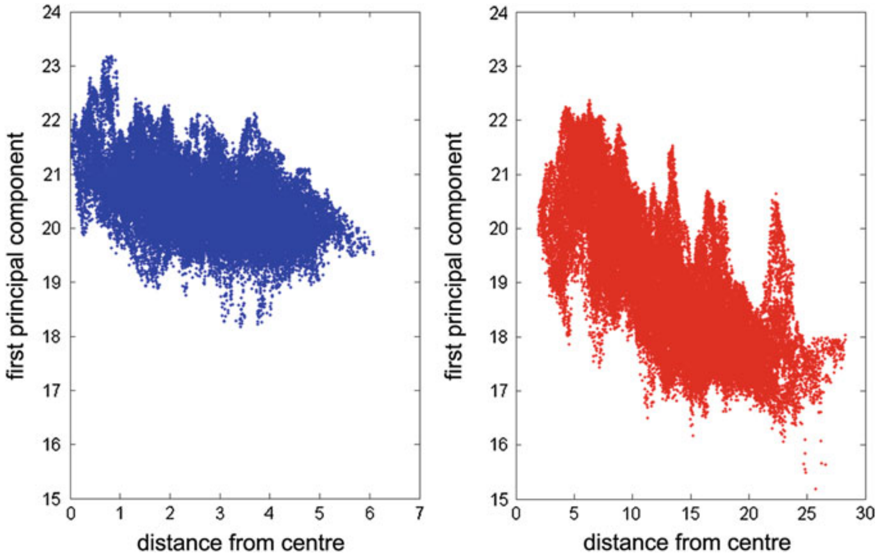


**Fig. 8** First principal components of $d = 200$ subgraphs plotted against radial distance from city centres. London (*left*) has a correlation of R = 0.521 ($R^2 = 0.271$); Seattle (*right*) has R = 0.752 ($R^2 = 0.566$)

cities as distinct from others. This can be viewed as a classification problem, for the sake of convenience selecting as the genotype particular features defined by some group of the geographical regions NOR, LAT, EUR, ARA and ASP. To the extent that autonomous interpretation by the machine is important, it must be possible to extract different, yet valid, genotypes from the data, depending on context, and to this end two different binary classifications were made (see also [19]). The first divides the globe through the Atlantic Ocean and the Eurasian land mass, distinguishing cities in NOR, LAT and ASP from those in EUR and ARA; the second splits the two main land masses by the Atlantic and Pacific oceans, dividing NOR and LAT from EUR, ARA and ASP.

A support vector machine (SVM) [23] was trained with the aim of mapping inputs of individual neighbourhood spectra to outputs of class labels (e.g. $[-1, 1]$). A radial basis function was used for the SVM, without optimising the parameters specifically to the problem but maintaining a standard setting of ($\sigma^2 = 30$ and $\gamma = 30$). Two different methods were used to produce the city spectral data: the whole city axial graph as in section "Whole Cities and Their Geographic Location," and the neighbourhood segment sub-graph as in section "Distinguishing Individual Neighbourhoods Within the City." In the latter case, each city is represented in the data set by 100 individual spectra, of which there is considerable variance within each city. To classify the city as a whole from these, classification is performed separately on each of the 100 spectra, and the mean value of these taken as the city class. In performing this classification, the normal binary distinction $[-1, 1]$ is not used, but instead is replaced by the real valued latent output corresponding to the degree to which each spectrum is a member of a given class. This effectively weights the most clearly classified spectra much higher than those on the borderline.

In both cases, the predicted classifications are the result leave-one-out cross validation, the least biased and most effective method of estimating true classification error [24], in which a single test example is held apart from the set while all others used to train, then used to validate, and a new training period undertaken for each subsequent example. In the case of the local neighbourhood representation, the spectra used represent subgraphs of 200 segments that partially overlap with several other samples in the same city, thus potentially contaminating the validation process if these samples cause partial information to be placed in both training set and test example. Leave-one-*city*-out, rather than leave-one-*spectrum*-out, was used, in which the training set for city $n$ included only spectra from all other cities. This has the effect of providing a true estimation of the predicted class of a new and unknown city, as no data from that city is made available to the algorithm until after training is complete.

The results are displayed in Table 1, which lists all misclassified cities for each case and notes the overall classification error. Both classifications and both representations perform well, with accurate classification of approximately 75–80 %. The two representations perform approximately equally, with the spectra of entire cities generally resulting in similar overall performance to that of classifying by the weighted vote of local neighbourhoods. When neighbourhood spectra are used, there is some difference in accuracy with the classification grouping East Asia with the Americas (20 % error for whole cities vs. 27 % for smaller spectra), but classification by land mass is an identical error of 24 %.

In the context of the machine's ability to interpret autonomously, the genotype should conform to two, superficially contradictory, criteria. It should be expected to capture properties of the cities that might be considered to be inherent, such as the geographical distinctions in the successful classification above, but to avoid the inflexibility associated with a priori standardised representations it should also allow for some differences each time it is extracted. These are not actually at odds, as the latter property is dependent on how the data set is labelled to determine the classes, and expressed in the specific cities that show up as correctly or incorrectly classified in each run. This is evident in the difference between the sets of misclassified cities in each column of Table 1. While the majority of cities were correctly classified in each run, a change in the labelling of the set, or the representation itself, produces a different division of classes, not only with the group that is moved between the classes (ASP), but in other regions as well. There would appear to be limits to the machine's ability to learn arbitrary divisions, as indicated by the largely disjoint sets of errors in the ASP region, and cities persistently misclassified (Ann Arbor, Rio de Janeiro) in the other regions. This indicates that despite the high dimensionality of the spectral input features, some inherent properties of the cities themselves are not easily be overruled by training.

Table 1 Errors in classification by SVM, and misclassified cities

| (NOR + LAT + ASP) versus (EUR + ARA) | | (NOR + LAT) versus (EUR + ARA + ASP) | |
|---|---|---|---|
| (Whole city) Error: 20 % | (by neighb'hoods) Error: 27 % | (Whole city) Error: 24 % | (by neighb'hoods) Error: 24 % |
| *ASP* | | | |
| Ahmedabad | Ahmedabad | – | – |
| Auckland | – | – | – |
| Dhaka | Dhaka | – | – |
| – | – | Chengkan | – |
| – | – | Johor Bahru | – |
| Hong Kong | Hong Kong | Hong Kong | – |
| – | Kyoto | Kyoto | – |
| – | Penang Island | – | – |
| – | Pequim | Pequim | – |
| Phuket | Phuket | – | – |
| – | Pingshan | – | – |
| – | – | – | Tokyo |
| Shanghai | Shanghai | – | – |
| – | – | Xidi | – |
| – | Yuliang | Yuliang | – |
| – | – | Zhanqi | – |
| *ARA* | | | |
| Kerman | – | Kerman | – |
| Gurgan | – | Gurgan | – |
| – | Adaban | – | – |
| – | Nain | – | – |
| *EUR* | | | |
| – | Aachen | – | – |
| Athens | Athens | – | Athens |
| Barcelona | – | Barcelona | – |
| Gassin | – | Gassin | – |
| Konya | Konya | – | – |
| Lisbon | – | Lisbon | – |
| – | – | London | – |
| Mytilini | – | Mytilini | – |
| Nauplion | Nauplion | Nauplion | – |
| Nicosia | Nicosia | – | – |
| Prague | Prague | – | – |
| – | Belgrade | – | Belgrade |
| – | Gotemburg | – | – |
| – | Helsinki | – | Helsinki |
| – | Manchester | – | – |
| – | Rome | – | – |
| – | Wolverhampton | – | – |
| *LAT* | | | |
| – | – | Alcantara | – |

(continued)

**Table 1** (continued)

| (NOR + LAT + ASP) versus (EUR + ARA) | | (NOR + LAT) versus (EUR + ARA + ASP) | |
|---|---|---|---|
| (Whole city) Error: 20 % | (by neighb'hoods) Error: 27 % | (Whole city) Error: 24 % | (by neighb'hoods) Error: 24 % |
| Aracaju | – | Aracaju | – |
| – | – | Brasilia | – |
| Cidade de Goias | Cidade de Goias | Cidade de Goias | Cidade de Goias |
| – | – | Diamantina | – |
| Florianopolis | – | Florianopolis | Florianopolis |
| Maceio | – | Maceio | – |
| – | Mucuge | Mucuge | Mucuge |
| Ouro Preto | Ouro Preto | Ouro Preto | – |
| – | Penedo | Penedo | Penedo |
| Petropolis | – | Petropolis | Petropolis |
| – | Pirenopolis | Pirenopolis | Pirenopolis |
| Porto Alegre | Porto Alegre | Porto Alegre | Porto Alegre |
| Rio de Janeiro | Rio de Janeiro | Rio de Janeiro | Rio de Janeiro |
| Salvador | Salvador | Salvador | Salvador |
| Sao Luis | – | Sao Luis | Sao Luis |
| – | Vitoria | Vitoria | Vitoria |
| – | – | – | Belem |
| – | – | – | Ico |
| – | – | – | Manaus |
| – | Parati | – | Parati |
| – | Recife | – | Recife |
| – | – | – | Cachoeira |
| – | – | – | Cuiaba |
| – | – | – | Goifnia and Aparecida |
| – | – | – | Joao Pessoa |
| – | – | – | Lencois |
| – | – | – | Mariana |
| – | – | – | Porto Seguro |
| – | – | – | Santiago |
| – | – | – | Sao Paulo |
| – | – | – | Tiradentes |
| *NOR* | | | |
| Ann Arbor | Ann Arbor | Ann Arbor | Ann Arbor |
| Washington | Washington | Washington | Washington |
| Atlanta | – | Atlanta | Atlanta |
| – | Baltimore | – | Baltimore |
| – | Las Vegas | – | Las Vegas |
| – | Seattle | – | Seattle |
| – | Pensacola | – | Pensacola |

## Conclusion

A method was presented for representing essential features of large urban maps such that a subset can be extracted to describe only those features common to a particular class. This description of the class is considered to meet the criteria of the inverted genotype [12] for urban areas. It differs from the principle of the biological genotypes in that it violates the 'central dogma' of molecular biology [13] which states that information flows in only one direction: from code (DNA) to structure (phenotype) in the form of a set of generative instructions. Instead, information flow is in the other direction, and the code is extracted from the structures themselves. This results in a description of the type that can only exist a posteriori.

A potential disadvantage of such an 'inverted' description is that it lacks an inherent mechanism for the production of new examples, as is expected in biological genetic descriptions. A separate mechanism would be required to do so, the details of which are beyond the scope of this paper, but may involve either one of many possible 'generate and test' procedures, or the modification of the general method of genotypic description to procedural rules themselves—i.e. the machine may learn from the process rather than the product.

The advantage of such a description, however, lies in the fact that it can be derived autonomously by the computer from a set of examples of the type, rather than relying on pre-defined, a priori definitions. This offers scope for our modelling of design processes and understanding of creativity in several ways:

First, the level of explanation offered is at a higher level, not on the description itself, but on the mechanism that produces it. Many of the assumptions implicit in classical artificial intelligence, via the physical symbol system hypothesis [6], seemed to suggest that describing the symbolic relationships 'correctly' would be equivalent to explaining intelligence. While successful mimicry of some aspects of intelligence resulted, this has turned out not to be a satisfying explanation, and there is no reason to assume that a description of similar (human written, static) rules for generating plausible design artefacts should be any more successful. Understanding the means by which such rules might be made, changed and communicated between individuals, may help. Extraction of descriptions from actual artefacts, as shown here, is one such possibility.

The second possible contribution to understanding lies in the fact that such a description naturally and unproblematically allows for changes. It may be recreated anew as the set of examples changes, and is entirely based on the set of examples presented. This therefore offers potential for the investigation of relationships between types, changes of style, and the creative process of interpretation and 'seeing as' [1].

In practice, a priori standards typical in CAD do not apply here, and a further understanding of how such descriptions may be used computationally may lead to useful forms of human-computer interaction as yet unexplored.

Objectives were presented in the form of three criteria for the representational scheme proposed here. It was to be (1) of sufficiently high dimensionality that subsets of features could be reliably extracted from it, and (2) capable of displaying morphological properties presumed to correlate with geographical location, both globally and locally. Section "Qualities of the Spectral Representation" demonstrated that the raw data supplied by the graph spectra sufficiently captured these morphological distinctions corresponding to geographical (and presumably cultural) distances. To fulfil requirements for non-standard, independent interpretation as demanded in the creative process of design, it was to be (3) capable of allowing different subsets of features to be described, depending on the labelling and structure of data. Section "Extraction of the Type from Classes of Examples" showed that the base representation was of a dimensionality high enough that different genotypic representations can be extracted.

It is unlikely that our human perception of city morphology is closely related in process to the graph spectral representations described here, but to the extent that we are able to draw distinctions between cities in the same geographical regions, the results of the two processes are similar. The spectral representation may thus be useful in exploring various morphological relationships in a quantitative manner, where traditional, qualitative methods lack required precision, and may thereby be used to investigate cultural communication in a manner similar to Cavalli-Sforsa et al. [21] or provide comparative evidence for design decisions. In the wider context, this and similar approaches to the extraction of descriptive information from artefacts themselves, indicate the feasibility for a computer to read a CAD file similar to the way we might, to interpret autonomously, and thus provide another experimental tool for research into the nature of processes in design.

**Acknowledgments** Many of the axial maps of Brazilian cities were originally compiled by Valerio Augusto Soares de Medeiros. The better part of the remainder used are unfortunately uncredited, so I would like to thank the unnamed authors as well as Shinichi Iida and Bill Hillier who have maintained and provided access to the bulk of the data set. This research has been supported in part by an EPSRC fellowship (EP/E500706/1).

## References

1. Schön DA (1963) Displacement of concepts. Tavistock, London
2. Koestler A (1964) The act of creation. Hutchinson, London
3. Czikszentmihalyi M (1988) Society, culture, and person: A systems view of creativity. In: Sternberg RJ (ed) The nature of creativity: contemporary psychological perspectives. Cambridge University Press, Cambridge, pp 325–339
4. Sutherland IE (1963) Sketchpad, a man-machine graphical communication system. Ph.D. thesis, MIT. Reprinted: Kuhn M (ed.) technical report UCAM-CL-TR-574, University of Cambridge Computer Laboratory. ISSN pp 1476–2986
5. Schumacher P (2009) Parametricism—A new global style for architecture and urban design. In: Leach N, Castle H (eds) AD architectural design: digital cities 79(4):14–23

6. Newell A, Simon H (1976) Computer science as empirical enquiry: symbols and search. Commun Assoc Comput Mach 19:32–105
7. Dreyfus HL, Dreyfus SE (1988) Making a mind versus modeling the brain: artificial intelligence back at a branchpoint. Artificial Intelligence. Reprinted in Boden MA (ed.) The Philosophy of Artificial Intelligence, Oxford, 117(1):309–333
8. Searle J (1980) Minds, brains and programs. Behav Brain Sci 3(3):417–457
9. Van Gelder T, Port RF (1995) It's about time: an overview of the dynamical approach to cognition. In: Port RF, Van Gelder T (eds.) Mind as Motion, MIT Press, pp 1–43
10. Kuhn TS (1962) The Structure of Scientific Revolutions, 3rd edn, University of Chicago Press, Chicago
11. Akin Ö, Akin C (1996) Frames of reference in architectural design: analysing the hyperacclamation (A-h-a-!). Des Stud 17(4):341–361
12. Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge
13. Crick F (1970) Central dogma of molecular biology. Nature 227:561–563
14. Turner A (2007) From axial to road-centre lines: a new representation for space syntax and a new model of route choice for transport network analysis. Environ Plan 34(3):539–555
15. Van Dam ER, Haemers WH (2002) Spectral characterizations of some distance-regular graphs. J Algebraic Combin 15:189–202
16. Zhu P, Wilson RC (2005) A study of graph spectra for comparing graphs. British machine vision conference 2005
17. Luo B, Wilson RC, Hancock ER (2003) Spectral embedding of graphs. Pattern Recogn 36:2213–2233
18. Robles-Kelly A, Hancock ER (2003) Edit distance from graph spectra. In: Proceedings of the 9th IEEE international conference on computer vision (ICCV 2003)
19. Hanna S (2009) Spectral comparison of large urban graphs. In: Koch D, Marcus LL, Steen J (eds.) Proceedings of the 7th international space syntax symposium. royal institute of technology (KTH), Stockholm, Sweden
20. Hanna S (2011) Comparative analysis of neighbourhoods using local graph spectra. In: Proceedings of the 8th international space syntax symposium. (Forthcoming)
21. Cavalli-Sforza LL, Menozzi P, Piazza A (1994) The history and geography of human genes. Princeton University Press, Princeton
22. Shepard D (1968) A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM national conference, New York, NY, pp 517–524, 27–29 Aug 1968
23. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
24. Reich Y, Barai SV (1999) Evaluating machine learning models for engineering problems. Artif Intell Eng 13(1999):257–272

# Part VIII
# Design Function

# Function–Behavior–Structure Representation of the Grids in Graphic Design

**Prasad Bokil and Shilpa Ranade**

**Abstract** Grid in graphic design is a well-known tool. It is used for planning and creating graphical layouts. Though lot of empirical literature on grids is available through various case studies and visual samples, there is hardly any articulation of the design process involving grids. This paper tries to present the formalism for design knowledge in application of grids in graphic design. The function-behaviour-structure framework is applied to grids for the understanding of design process. This framework is used to create a conceptual model for a grid in action and to define its variables. To demonstrate its significance, the potential advantages of this new approach are discussed.

## Introduction

Graphic design as a knowledge domain is deeply rooted in arts, crafts, and the cultural history of human civilization. However, the research in graphic design is a recent phenomenon and it has been maturing slowly [1, 2]. The significance of research in graphic design has been debated since long. Design research is either considered an oxymoron and trivialized or neglected [3]. In the last few decades, the research in graphic design has been taking shape, but the methods and methodologies are yet to be established. The efforts are now made to go beyond the practical aspects of graphic design to understand the design process itself. One such approach is function-behaviour-structure framework, which is established to articulate the design description. It is proposed for a general design theory

---

P. Bokil (✉) · S. Ranade
IDC, Indian Institute of Technology Bombay, Mumbai, India
e-mail: prasad_idc@iitb.ac.in

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_27,

independent of specific design domain but it is mainly established in the fields like CAD, product design, and engineering design. Such a theoretical approach is generally neglected by the practicing graphic designers. Hence, in the field of graphic design, function-behaviour-structure framework is hardly ever used for knowledge representation and knowledge transfer.

In this paper, we extend the scope of FBS framework by applying it to grids, a common tool in graphic design. The aim of this paper is to validate the applicability of this framework, in the process of layout design, and to discuss the emerging possibilities. FBS framework has long been proved to give a satisfactory basis for the development of process models in routine design. It will help to understand the graphic design process and will establish the platform to assist in pedagogy, regular practice, and innovation.

## *Grids in Graphic Design*

The grid is a well-known tool in graphic design mainly in typography. It generally consists of a fixed set of guidelines. It can be minimalistic like points, lines, or basic shapes like triangle, square, or it can be complicated like second or third degree curves in three dimensions. In abstraction, it is nothing but the natural urge to seek for order or regularity.

The use of grid for visual layouts is not at all new. In visual arts, we can find the evidences of effective use of grid from the pre-medieval period [4, 5] up to the works of Escher [6]. In Indian art, the grid always had a large variety of form and application [7]. It is still in use for image-making [8]. In the last few centuries, the grid has relocated its role in the perception by the rational cognition [9].

The modern contemporary grid was fully developed in its current form and use, by second decade of the twentieth century [10]. In the early twentieth century, a new design philosophy emerged out of constructivism, which was inclined towards rationalism and minimalism. Obviously, such a design identified the rectangular grid with vertical and horizontal guidelines as a very essential tool and became popular in the group of theoreticians in academia.

Controlling the space, as it leads to achieve the balance, structure, and unity, is always the main concern in a graphic layout. Grid can be used to create a hierarchy through a careful control of measured space [11]. From a designer's point of view, the advantages of grid can be categorised under two types: (1) visual decision-making (2) managerial decision-making [12]. The advantages for visual decisions are with respect to content management, space management, consistency, and visual identity. In addition, the advantages for managerial decisions are in the area of work distribution, workflow management, optimum time, and resource management.

### Changing Scenario of Grid

The use of grid in modern design many times is limited to the static applications in typography and hardly stretched beyond for the dynamic applications. In last few decades, the medium, tools and subject matter have changed and; so have the requirements.

New digital technology has created enormous virtual space and brought in lot of challenges to manage it. Surely, the use of grid is in demand for web designing but it is more for the convenience of the fast-paced readymade computer graphics than for better design solutions. The two foremost opportunities created by new technology are in the field of dynamic media and interactive media. In the layouts, where the viewer is allowed to interact and as a response, the interface is changing, the distribution of space is no more static.

The advent of new technology has led to the emergence of several new needs: demand for high pace production, continuous thrust for change in content and layout, image heavy information panels, dynamic visuals, and easy customization. This indicates the great demand for more flexible, information heavy, customizable, ever changing but still homogeneous graphic solutions within least possible time.

## Knowledge Representation of Grid in Graphic Design

Graphic design as profession is highly practice based and application oriented. It can be observed that the practice and pedagogy of graphic design does not give much stress on articulation and theorization. Although in the last few decades the research in design has started to reveal the mystery of creative thinking and designerly way of looking, there are many areas in design practice, which are limited to thumb-rules and intuitions. Grid is one of such tool, which is assumed as a trivial and intuitive. There is enough literature available to share the practicalities but proper knowledge representation and scientific articulation is almost missing.

### *Representation of Grid: Review*

Grid is very common tool in graphic design. The history has preserved and documented some grid based architectures [13] and designs [14]. However, unfortunately, there was hardly any attempt of articulation or knowledge representation. In the modern times, graphic design was redefined with design schools like Bauhaus and Ulm. With the advancement of printing technology grid has gained the importance in print and publication design. Grid as design tool is extensively exploited in the domain of type design and typography. However, the first extensive writing on grids was by Joseph Müller-Brockman in the form of a book-Grid Systems [15]. It explains the role of grid in design, its advantages and demonstrates formulation of grid through design steps. It is followed by lots of examples of design layouts. It is expected from the reader to understand the process of design from the examples and design descriptions. Almost all literature published on grids [11, 16–20] and about typography [21–23] followed the same model for knowledge transfer.

This model of knowledge transfer is based on direct internalization of knowledge through observations. Available literature discusses the grid philosophically and then for practicalities focuses on the examples; finally the readers are left to decode the structure and the involved thought process. The procedural knowledge is not externalized. The same methodology is generally implemented in pedagogy of graphic design [24].

The lack of articulation of cognitive steps in designing a visual using grid is never got enough attention since the routine design is limited to current model of knowledge transfer. But this model with direct internalization through empirical methods demands lots of practice and time. Current status of design literature shows the incomplete knowledge externalization, lack of knowledge representation and inefficient model of knowledge transfer.

Many designers and design experts mention the fear of being trapped in grids. It is unanimously accepted in design community that the grid, if not used creatively, may lead to uninteresting and ineffective visual. However, we believe that this problem is rooted in the lack of proper knowledge representation. Without profound representation, it is difficult to provide the means for ‘creative use’ of grid. Design research has separated creative and innovative design from routine one [25]. Applying same principles for grid, if it is defined with appropriate variables, the change in the value of such variables will result in the innovative use of grid; and change in the variables will give creative use of grid.

## *Constructing the Definition and Functional Model*

The ambiguity in articulation of grid starts with the definition of grid. Many design books just avoid defining grid; they use this term as lay term. The literature does not give any common definition. There are many descriptions and definitions but whatever description is available does not cover the total concept of grid. For the concept of grid, expanded beyond the application of typography, the definition should also be more inclusive.

One can describe grid with working experience or it can be explained to connect with epistemology. Grid etymologically means the mesh of horizontal and vertical lines. However, the practice has shown abundance of variations in the form and usage of grids. It is no more possible to restrain the term to the set of vertical and horizontal lines.

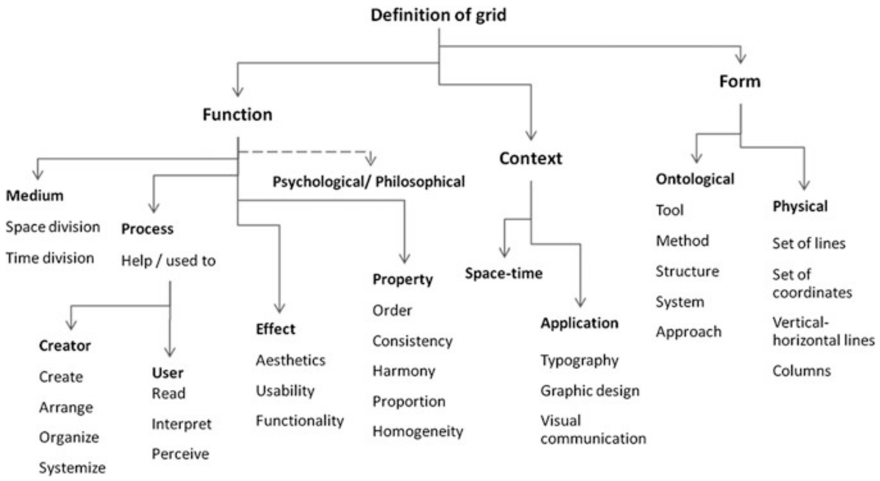


**Fig. 1** Relationship analysis of definition of grid

To formulate the definition of grid the methodology proposed by Sarkar and Chakrabarti [26] is used as a reference. The method of Relationship analysis is borrowed and modified for this case. We started with the collection of 61 definitions and descriptions, from literature and from the field survey, which then analyzed to formulate a definition of grid.

After analyzing all the definitions, it is found that each definition consisted of few basic semantic units. They necessarily describe function of grid, its form and/or locate it in the design application context. The relationship diagram thus created for grid is shown in Fig. 1. From this analysis, the definition of grid can be generated by equation below.

$$\begin{aligned}\text{Grid} = {} & (\text{Ontological form} + \text{Physical form}) + (\text{Medium} + \text{Process}[(\text{Person})\ \text{Action}] \\ & + \text{Property} + \text{Effect} + \text{Psycho - Philosophical}) \\ & + (\text{Application context} + \text{Socio - cultural context})\end{aligned} \tag{1}$$

Hence, the grid can be defined as ‘a pre-determined understructure in the geometric form that creates the consciousness of the space and guides the designer to bring order and consistency in the visual representation and the arrangement the visual information in the demarcated space for better aesthetics and functionality’.

Designers use grid to create the visuals. The grid may or may not be perceived by the viewer. Grid contains the graphic elements within the page. Apart from arranging the elements within the space, grid sometimes helps to create the graphic elements like in case of type design. Grid helps designer to manage the space within the given boundary. Thus, the functional model of grid is represented by Fig. 2.

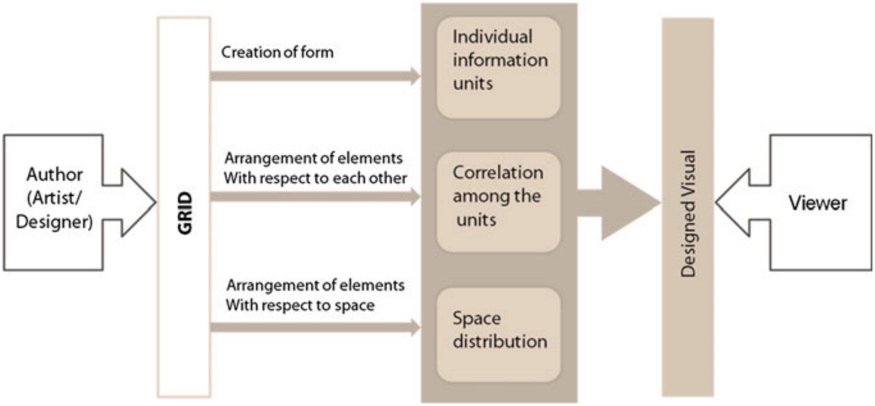


**Fig. 2** Functional model of grid in visual communication

## *Considering Grid as Design*

While defining grid, designers use terms such as tool, method, structure, approach, etc. The concept of grid is not limited to a structure or tool which is used in design but it is equally important the way it is made and the way it is used in design.

Throughout the design literature, the grid is treated as a tool or method. Treating grid as a design tool includes the practical aspect of its use in design. As a method, designers struggle to create appropriate grid and use it for designing a visual. Although the process of creating grid is a part of design process, the flow of the process is not clear. This makes the discussion fuzzy. The form and application of grid and visual are all mixed up and under focus at the same time. A sound theoretical foundation is required to make the representation more rigorous.

Creating a visual is a goal-oriented, constrained, decision-making, exploration and learning activity that depends on the designer's perception of context. In addition, so is the making of grid. According to the definition of design by Gero [27], the activity of making grid can be considered as a design activity with a goal of creating grid to assist designer in decision making while creating visuals. 'Considering the grid as a designed object rather than a tool or method' may look like a trivial step but it is a crucial one as it opens the door of applying design theories to understand and articulate grids in graphic design. The designer is treated here as user which is consistent with Gero and Kannengiesser's approach for Representational affordances in design [28].

Making the conceptual model of grid (Fig. 2) more explicit gives us two levels as shown in Fig. 3. Here, the grid is considered as a designed object at the first level and designer is the user. Designer, then at the second level, uses the grid to design visual for expected viewer. Treating grid as a design gives more focus on grid itself and use of grid is separated from the effect of designed visual. The main advantage of this two-level model is the applicability of design research methodology like FBS model in the study of grids.

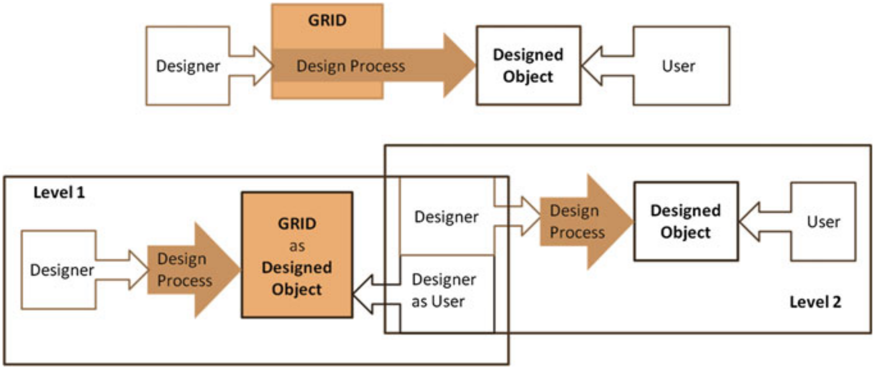


**Fig. 3** A two-level model for graphic design process with grids

## FBS Framework for Grids in Graphic Design

The discussion of grids in graphic design is characterized by lots of subjectivity, which results in the ambiguity of representation of design process. Treating grid as a design, rather than a design tool, allows us to implement various established frameworks to explain the process better. FBS framework s proved effective in the articulation of design process. Although FBS framework is claimed to be generic representation of all design domains including graphic design, it is rarely evaluated for actual graphic design process in practice. In this paper, FBS framework is extended for this practical tool in graphic design.

### *FBS: Background*

The FBS ontology [29, 30] is first developed to represent designed objects in terms of their structure, working and purpose to cater a design requirement. It has proved effective to represent the design at different stages of development. Structure (S) of a designed object is consists of elements it is composed of, their attributes and relationship among them. Behaviour (B) is inferred by attributes derived from its structure, by which it achieves function. Moreover, Function (F) in simple terms is the purpose of designed object. It is then extended from representation of design objects to representation of various processes [30].

FBS framework [27] describes the design with causal relationship between function, behaviour, structure and it is defined by three classes of variables: function variables, structure variables and behaviour variables. These variables are linked together by eight processes as shown in Fig. 4. These processes, which are claimed to be common for all designing, are formulation (1), synthesis (2), analysis (3), evaluation (4), documentation (5) and three reformulation processes (6, 7, 8)

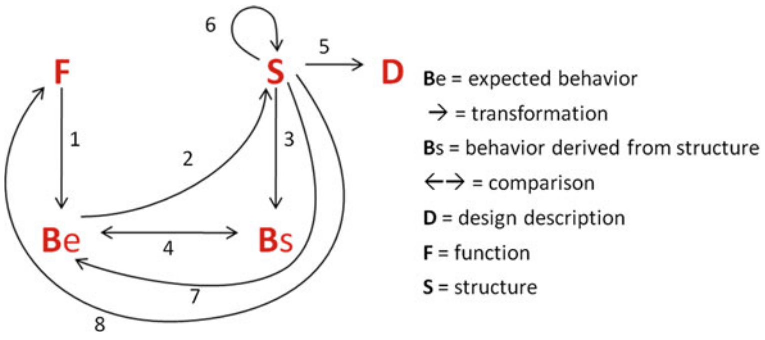


**Fig. 4** The FBS framework (after Gero [27])

for design iterations with a range of values of variables. These processes are responsible for the transformation of three classes of variables into each other.

The FBS framework gives a model that represents design process at different states. It helps to understand the design process by making it explicit. This framework gives a design prototype that enables a designer to start designing with incomplete information and allows adaptations based on the information gathered during the design process. By adopting this framework, the knowledge representation schema obtainable with it can be utilized for routine, innovative, and creative design of graphics.

## *FBS Representation of Grids*

The grid as design activity is analyzed based on Function-Behaviour-Structure schema. This schema can be useful to articulate the use of grids in graphic design process. It can then be possible to create a computational model. Such model can facilitate the use of grids in new digital media. It can help designer to use grids more creatively.

### Two FBS Paths

The process of graphic design with grids can be articulated by FBS framework with FBS path for each of the two levels as shown in Fig. 3. In this framework, function, behaviour, and structure are assumed to be connected with each other by processes that transform these states into each other. Out of eight processes shown in Fig. 4, reformulation processes are omitted here for simplicity. These omitted processes are secondary in order and can be considered after maturation of the initial concept. The processes retained for the explanation are formulation, synthesis, analysis, evaluation, and documentation.

As explained above, there are two levels—(1) grid level and (2) visual level. On visual level, function of a visual is to communicate with viewer effectively. Aesthetic pleasure can be primary or secondary function. The Behaviour is defined by the hierarchy, order, harmony of elements and other visual means by which the communication is received. Structure can be understood by various visual elements, their shapes, colours, respective spacing, typography, and placement with respect to each other. On grid level, function is to assist designer to channel the behaviour of visual by controlling order, hierarchy, harmony etc. Behaviour of grid is 'the possibilities it creates' for designer to follow, like alignments, spatial divisions etc. And structure is the physical form and description of grid.

Hence, there are two FBS paths representing 'making of visual' and 'making of grid'. However, the process of designing a visual involves making of a/the grid as an intrinsic part of the process. The two paths should be connected to each other by the flow of causality. For articulating the knowledge of applying grid, it is essential to understand how these two frameworks are connected to each other and the correlation between them.

## Connection Between Two Levels of FBS

To connect these two paths, let us examine the process of visual design based on grids. The process of 'creating grid for a particular visual and creating the visual based on it' is hardly ever articulated. The design literature, in the form of books, booklets and web pages, demonstrate this process through the examples [11, 15, 18, 24]. The available articulation is quite incomplete and abstract. The steps involved can be generalized [17] as below:

1. Understand the communication problem
2. Know the physical resources and content
3. Conceptualization
4. Laying down the final grid
5. Designing based on the grid.

This articulation is not explicit and doesn't give the clear idea for the exact representation of knowledge. In design practice, the stress is given on the internalization of knowledge through observations, hands-on trials, repetitions and onsite practice. But this direct internalization through empirical methods takes lots of practice and hence longer time. The externalization may help to make this process faster. It will also help to improve the design process involving computers. Table 1 given below tries to explicit the implicit knowledge in the process of layout design for required communication. The last column indicates the FBS path of the design process. For the clarity in the symbolic notation the two FBS loops, at grid level and visual level, are separated by adding 'G' as prefix to FBS path at grid level. Arrow '→' indicates the transformation and '↔' indicates the comparison of two states.

**Table 1** Explicit process of communication design with grids

| Design process | Step | Implicit steps made explicit | FBS path |
|---|---|---|---|
| Understand the | | communication problem | 1 |
| | | Understand the purpose to be fulfilled by the expected communication | F (function) |
| | 2 | Note the agents involved as a owner or user of communication | These steps can be explained under—Situatedness[a] of design agents and design problem |
| | 3 | Understand the context of design | |
| 4 | | Decide the channel and medium of communication | |
| Know the physical resources and content | 5 | Check all possible resources available for use | |
| | 6 | Prepare the content (if required) or study it | |
| | 7 | Understand the acceptable code of communication within the users | |
| Conceptualization | 8 | Based on the knowledge gathered in steps 1–6 contemplate the means by which purpose of design can be solved | F → Be (expected behaviour) |
| | 9 | Assign the structural properties to elements. e.g.: font, size, colour, etc. | Fixing values of structure variables[b] |
| | 10 | Based on the expected behaviour prepare the rough layout | Be → S (structure) |
| | 11 | Make optional layouts | F → Be → S → Bs quick iterations |
| Laying down the final grid | 12 | Based on the rough structure the behaviour of visual is finalized | Fixing values of behaviour variables[b] |
| (Level 2) | 13 | The purpose of grid is to assist the designer to create a structure for achieving the expected behaviour of visual | Be → GF (grid function) |
| | 14 | Based on the rough layout the working of grid is realized | GF → GBe (expected grid behaviour) |
| | 15 | The grid is designed to support the expected behaviour of grid | GBe → GS (grid structure) |
| Designing based on the grid | 16 | Designed grid with its actual behaviour (GBs) is used to decide the placing and shapes of elements in the visual layout | GBs → S<br>Fixing remaining values of structure variables[b] |
| | 17 | The layout is designed as planned | S |
| | 18 | The layout is verified for the expected result by comparing actual behaviour to expected behaviour | Bs↔Be [GBs↔GBe, if required] |

(continued)

**Table 1** (continued)

| Design process | Step | Implicit steps made explicit | FBS path |
|---|---|---|---|
| | 19[c] | Variations for improvement (without changing grid) | Reformulations on the level of visual design |
| | 20[c] | Variations for improvement (with changes in grid) | Reformulations on grid level |

[a] Situatedness implies that where you are when you do what you do matters. This concept is introduced in design research by Gero (1998) and proved useful to understand the context of design

[b] Three types of variables of grids are discussed in 3.3

[c] There are three Reformulations as shown in Fig. 4 and each involves multiple steps. These steps are not detailed out to reduce the complexity

It is the behaviour of grid which gets transferred to visual. Once the design is over the structure of grid has no further use and hence it is removed. Two level FBS derived from this process is represented in Fig. 5. The FBS path for visual design is denoted as 1v → 2v → 3v↔4v → 5v and FBS path for grid design is marked as 1 g → 2 g → 3 g↔4 g → 5 g.

As represented by Fig. 5, designer works on the two levels. First, the design brief is studied and the requirements are articulated. It gives the function of the visual to be designed (F). From the function the expected behaviour (Be) of visual is imagined by designer and a rough structure (S) is conceptualized. The conceptualization of rough structure does not require the grid, but a designer follows the notion of tacit grid that is based on the provisional behaviour of grid. The tentative structure, such formed, gives the idea about actual behaviour (Bs) of visual. This actual behaviour is compared to expected behaviour and if the designer is not satisfied with the potential design output, the variables are reformulated. These iterations continue until the rough layout shows the potential to satisfy the design requirements.

Once the rough layout is finalized designer starts with final design. Expected visual behaviour leads the designer to FBS path of grid design. The expected behaviour of grid (GBe) imagined by the designer while creating rough design helps him to design final grid structure (GS). The actual grid behaviour (GBs) provides the set of values for designer to choose while creating the final visual structure (S). The layout such created is tested by comparing actual behaviour and behaviour expected by designer. If required, the reformulation of variables is carried out at all three stages of both levels.

First reformulation preferably occurs at visual level without changing the grid. The grid behaviour has created many possibilities for structure variables of visual design. For example, sets of different positions, orientations, proportions etc. These values are altered to get the expected results. If by any means the expected behaviour and thus the function are not achieved by the visual then the designer might enter into the grid level FBS path to make reformations of grid variables.

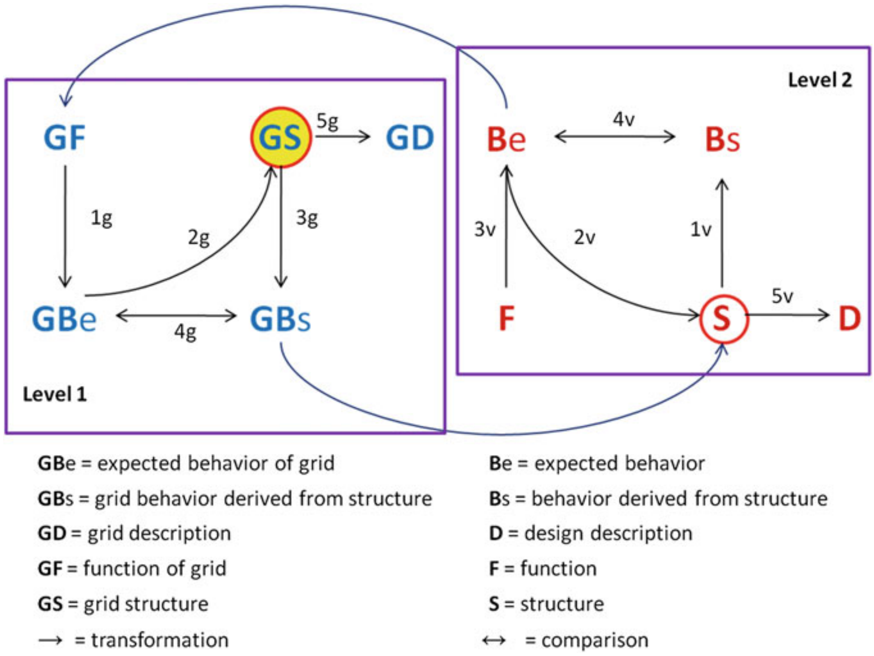


**Fig. 5** The FBS framework on two levels of visual designed using grids

## Variables of Grid

The FBS path gives a framework to articulate the graphic design process involving grids. To get more control on the process and to understand the influential factors in visual design it is necessary to define the variables of grid. The variables in FBS framework are defined and generalized by Qian and Gero [31]. By extending the same approach, the variables of grid are defined here.

### *Structure Variables*

A structure variable is any member from the set of elements, attributes, relationships, operations, processes that define the structure. In case of a grid, the structure is static. The grid structure can be defined by three finite sets of elements, attributes, and relationships.

$$\mathrm{GS} = \{\mathrm{E},\ \mathrm{A},\ \mathrm{R}\} \tag{2}$$

where,

| | |
|---|---|
| E | a finite set of element variables; |
| A | a finite set of attribute variables; |
| R | a finite set of relation variables. |

Grid is made out of primitive elements likes points, lines, curves and sometimes these primitive elements arranged together to form structure elements like equally spaced rectangles in the case of modular grid. The attribute variables are the properties of elements like the co-ordinates, lengths, thickness, and the equations of the curves. Relationship variables are mainly spatial relations among the elements. The set of relation variables is consists of distances between every two elements, their contact points, and intersections.

The grids are mainly defined for static visuals and hence they are static structures. They do not involve any process or operations as a part of structure. They are assumed not to change independent of designer.

## *Behaviour Variables*

Behaviour is the way by which the meaning of structure is inferred. It is useful in design process for problem formulation, synthesis, analysis, evaluation, and reformulation [27]. The behaviour of grid is spatial behaviour. It is responsible for the final layout of visual. There is no temporal factor involved in grids created for static visuals. Since behaviour connects structure and function [32] the behaviour variables of grid plays very important role in the creation of visual. Behaviour variables of grid on the basic level are structural type. It means they are derived from the structure itself without any external effect.

There can be large number of grid structures possible based on structure variables but the number of observed behaviour variables are limited. Our previous study [33] has demonstrated five variables of grid that have been called as syntactic variables. After applying FBS structure, it is realized that these are nothing but five structural type behaviour variables. These variables are as follows—position, orientation, area, proportion, and sequence. Given a structure of a grid, some of these behaviour variables, or all, attain a finite set of values, which help designer to take appropriate design decisions. It puts some restrictions on design but it helps to work faster and bring the discipline in the process.

When applied with the content of design, each of these direct behaviour variables, alone or in combination, gives exogenous (indirect) variable. Exogenous type of behaviour variable is a variable that is shown when external object (in this case ‘the content’) is applied to the structure. The exogenous type variables such derived are semantic module, direction, plan, perspective, and hierarchy.

### *Function Variables*

There is no uniform representation or fixed definition of function [31]. In case of grid, the design literature gives range of descriptions from physical to abstract nature. The relationship analysis chart (Fig. 1) shows the basic aspects of function of grid. The grid is created and used by designer. Grid assists designer by creating a finite set of values, to choose from, for structural variables of visual design. It helps a designer to create a visual with an expected effect, which is the result of structural attributes and behaviour of visual. Hence, the function of grid is to be transformed into the behaviour of visual.

Behaviour variables when applied with the visual content transformed into primitive function variables. The functional model of grid as shown in Fig. 2 gives three categories of function variables—creation, correlation, and space management. Creation variables help in creating any visual forms (structure elements) within a visual design process. Correlation variables define the structural correlation among the structure elements of visual. Space management variables control the distribution and the flow of negative (empty) space.

## Discussion

The application of FBS framework for grid in graphic design is described in the previous section. This knowledge representation can be proved beneficial for graphic designers. It will improve their understanding of the design process and it will open a window for new explorations.

### *FBS Framework for Better Understanding*

The lack of proper understanding of design activity limits the ability to create the support for design. The FBS framework gives elaborate theoretical foundation for representation of knowledge about ‘grids in action or stand alone’. It might looks complicated but it is essential for the development of a knowledge-based approach in graphic design. This framework makes the process of design and application of grid more explicit hence provide the better understanding of this design tool.

With this framework, it will be possible to establish correlation between FBS path of visual and FBS path of grid. It can lead us to create better correlation among the behaviour of visual and structure of grid. The most expressed concern about the grid is unimaginative use of grid. However, there is no explicit knowledge available for using the grids creatively. The behaviour variables as defined above can be used to create such articulation for achieving flexible layouts from a single structure of grid.

Due to the lack of externalization of knowledge in design domain, it takes long time to internalize this tool by observation and practice. This framework can help to reduce this time. It can be advantageous in design pedagogy.

In the last two decades, there has been a noticeable development in design research from cognitive perspective. Researchers have created computational and cognitive models to understand the design, process, and context. FBS framework, as discussed above, is well situated in such methodology. It gives the advantage of extending theories within same methodology and testing them for visual design.

## *FBS Framework for Grid Definition*

The definition, created by relationship analysis, is mentioned in previous discourse. However, this definition is subjective and dependent on the context, designer, and/or effect on the visual. The FBS framework can be used to derive an objective definition.

Many function variables can be expected from a designed grid. Each variable can have different values. The function based on the effect on visual is subjective and so is the definition derived from it. It may vary from person to person. In addition, it can be biased. Although the grid is claimed to bring the order, discipline, consistency, homogeneity, these variables cannot be included in the definition. The definition of grid should hold true independent of its effect on visual.

The grid cannot be defined in terms of structure variables. The structural elements can be in the primitive form of point, line, curves or in the structure form of squares, circles, columns or can be more complex like combination of regular, irregular shapes. It is not feasible to include all possible descriptions of structure in the definition.

However, independent of the structure and independent of the function being achieved or not, the grid is characterized by its behaviour. Although, there are a variety of behaviour variables as mentioned in Sect.“FBS Framework for Grids in Graphic Design”, it is possible to generalize various structural behaviour variables into single behaviour description. In whichever form the grid fixes the set of options from the continuation of infinite possible values, which can be used by a designer while designing the visual. Hence, characteristic behaviour of grid is the discretization produced by its structure.

Thus, the following definition emerged:

> ‘Grid is a structure used to discretize any continuum’

In graphic design, the definition is limited to space continuum but it can be extended for other applications. This definition is independent of structure variables and function variables and hence independent of content, context and design agents. It can cover all existing and potential forms and functions of grid.

## *New Experimentation with Grids*

The grids are used in modern graphic design for almost a century. Many expert designers have shown a creative and effective use of grids in creating graphics. All the variations in the application of grid are intuitive and evolved in years of practice. Still the experimentation of grids is limited to few set patterns. FBS framework can help to locate these patterns and can highlight the scope for innovative experimentation. The affordance of the grid can be systematically extended from reflexive to reflective type [28].

In addition, most of the grid application is controlled manually. The software applications of grid are highly influenced by manual design process. New explorations of grid can make suitable for intelligent design applications in evolving technology. For example, grid can be applied with Bezier technology to have automatic control over image making. FBS path can provide an essential tool to monitor such experimentation.

By explaining the grid with FBS path, it is possible to extend it for other models within the same methodological framework. It can be subjected to analogy based innovation approach [25, 31]. By appropriate matching and mapping of FBS path of grid in different domains, it is possible to create innovative grids or methods for applying grids. One of such experiment with grids [34, 35] has demonstrated an analogy based approach where a grid behaviour from one design domain is transferred to another design domain (see Fig. 6).

This experiment has shown a promising result in terms of innovative design approach for font design.

## *New Application Potential*

The knowledge representation in FBS will help to understand the process of graphic design and to articulate it to minute details. It is very essential in the process of transferring this manual knowledge into computers. This will lead the designer towards new potential applications of grid.

Once the behaviour of grid and visual is decomposed in various variables, it will be possible to introduce a mechanical control over the layout design. The system of user interface during Human–Computer interaction can become intelligent to vary the grid variables and hence the visual design based on the user response.

The way different variables of grid are defined above, it ascertains the static nature of grid and its application. With FBS framework, new variables in terms of structure variables (operations and processes) or temporal behaviour variables can be introduced. It will open a completely new range of applications like, interaction design, films, new media etc. It is possible to conduct a systematic study by introducing and controlling various variables with respect to time.

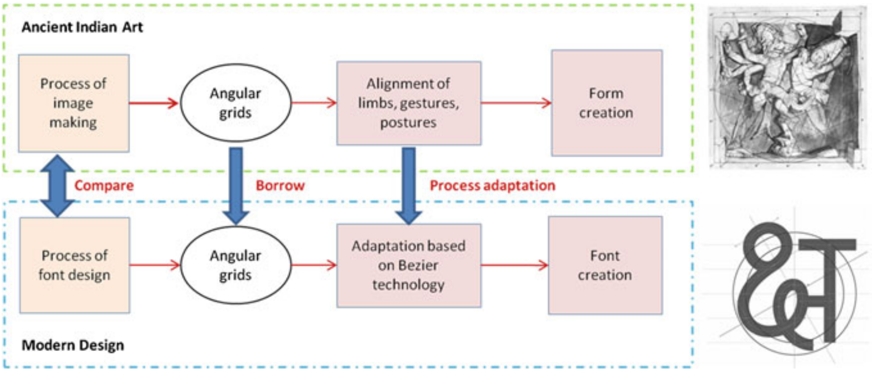


**Fig. 6** Compare and Borrow angular grids from image making to font design

As a tool, the grid is exploited in the field of typography but it rarely comes in the discussion of any non-typographic design applications. Etymologically the very concept of grid is still deprived of the proper attention in the research domain. The use of grid is limited, even though not restricted, to create rectangular space distribution that is used to decide the placement of elements and their alignment in vertical and horizontal direction. Grid is employed for the arrangement of different elements but hardly for the shaping of the internal structure of elements. It would be worth to deconstruct the concept of grid from typography to overcome its limitations and to make it open for new explorations.

## Conclusion

The process of Graphic design is not yet well understood. The knowledge representation and knowledge transfer in graphic design are not enough explicit. The grid in graphic design is one of the commonly used tools but in terms of cognition and design thinking, it is not yet investigated.

The main contribution of this paper is in knowledge representation of process of designing visual layouts using grids. This paper has discussed the articulation of design process involving grids. It has highlighted its potential in graphic design. This framework endows designer with better understanding and theoretical foundation for further research and experimentation with grids. It provides a good analytical tool and an objective perspective to study creative complex process like layout design.

As a limitation of the paper, the possible advantages of this approach are only discussed here and they are not well supported by the appropriate examples and evaluations. However, it is just the beginning of this endeavour. It needs to be explained and explored further.

## References

1. Bennett A (2006) The rise of research in graphic design. In: Bennett A (ed) Design studies, theory and research in graphic design. Princeton Architectural Press, New York
2. Throop E (2006) Being serious, being popular: positioning design research. In: Bennett A (ed) Design studies, theory and research in graphic design. Princeton Architectural Press, New York
3. Krippendorff K (2007) Design research, an Oxymoron? In: Michel R (ed) Design research now. Birkhauser Verlag AG, Berlin
4. Lawlor R (2002) Sacred geometry—philosophy and practice. Thames and Hudson, London
5. Boner A, Sarma SR, Baumer B (2000) Vastusutra Upanisad—the essence of form in sacred art. Motilal Banarasidas, Delhi
6. Locher JL (1982) Escher. Thanmes and Hudson, London
7. Vatsyayan K (1983) The square and the circle of the Indian arts. Roli Books International, New Delhi
8. Sthapati VG (2002) Indian sculpture and iconography; forms and measurements. Sri Aurobindo Institute of Research in Social Sciences in association with Mapin, Pondicherry
9. Williamson JH (1986) The grid: history, use and meaning. Des Issues 3(2):15–30
10. Gottschall E (1991) Typographic communication today, 2nd edn. The MIT Press, London
11. Swann A (1989) How to understand and use GRIDS. Quarto Publishing plc, London
12. Bokil PP, Ranade SR (2009) Diagonal grid in graphic design. IASDR conference electronic proceedings,pp 3263–3272. ISBN 978-89-963194-0-5
13. Kramrisch S (1976) The Hindu temples. Motilal Banarasidass, Delhi
14. Boner A (1962) Principles of composition in Hindu sculpture. Brill, Leiden
15. Müller-Brockman J (1981) Grid systems in graphic design. Arthur Niggli, Niederteufen
16. Vignelli M (1976) Grids: their meaning and use for federal designers. National Endowment for the arts USA, Washington. ISBN 036-000-00038-4
17. Hurlburt A (1978) The Grid—a modular system for the design and production of newspapers, magazines, and books. Van Nostrand Reinhold, New York
18. Bosshard HR (2000) The typographic grid. In: Bluhm A (Trans) Verlag Niggli, Zurich. ISBN 13: 978-3-7212-0340-0
19. Samara T (2005) Making and breaking the grid. Rockport, Beverly
20. Vinh K, Boulton M (2007) Grids are good. Available at http://www.subtraction.com/pics/0703/grids_are_good.pdf. Accessed 23 Sept 2010
21. Ruder E (1984) Typography, A manual of design. Hastings House, New York. ISBN 0-8038-7223-2
22. Baines P, Haslam A (2005) Type and typography. Watson Guptill, New York
23. Elam K (2007) Typographic systems. Princeton Architectural Press, New York. ISBN 13: 978-1-56898-687-6
24. Elam K (2004) Grid systems. Princeton Architectural Press, New York
25. Goel AK (1997) Design, analogy, and creativity. IEEE Expert, Intell Syst Appl 12(3):62–70
26. Sarkar P, Chakrabarti A (2008) Studying engineering design creativity—developing a common definition and associated measures. In: Gero JS (ed) Studying design creativity. Springer, Berlin
27. Gero JS (1990) Design prototypes: acknowledge representation scheme for design. AI Mag 11(4):26–36
28. Gero JS, Kannengiesser U (2012) Representational affordances in design, with examples from analogy making and optimization. Res Eng Des 23:235–249
29. Sembugamoorthy V, Chandrasekaran B (1986) Functional representation of devices and compilation of diagnostic problem solving systems. In: Kolodner J, Riesbeck C (eds) Experience, memory and reasoning. Erlbaum, New Jersey
30. Gero JS, Kannengiesser U (2007) A function-behaviour-structure ontology of processes. AIEDAM 21(3):295–308

31. Qian L, Gero JS (1996) Function-behaviour-structure paths and their role in analogy-based design. AIEDAM 10(4):289–312
32. Tham K, Lee H, Gero JS (1990) Building envelope design using design prototypes. ASHRAE Trans 96(2):508–520
33. Bokil PP (2009) Functions of grid, a key for flexibility in framework. In: Poovaiah R (ed) Des Thoughts 2:42–48
34. Bokil PP, Ranade SS (2011) Image making of the letterforms. In: Chakrabarti A (ed) Research into design. Research Publishing, Singapore
35. Bokil PP, Ranade SS (2012) Across the time and space: application of systems thinking to bridge the past and the future of design practices. Int J des manage prof pract 6(2):51–71

# Beyond Function–Behavior–Structure

**Mahmoud Dinar, Chris Maclellan, Andreea Danielescu,**
**Jami Shah and Pat Langley**

**Abstract** Our research is investigating the relationship between design problem formulation and creative outcome. Our research is investigating the relationship between design problem formulation and creative outcome. Towards that goal we have conducted experiments with designers engaged in problem formulation. In order to analyze such empirical data, a formal representation is needed. One popular model is Function-Behavior-Structure (FBS) and its several variants. Our problem map (P-map) model shares many features with FBS but also has important differences. We introduce a hierarchical representation not only in each of the F, B, S domains but in additional domains (requirements and issues). We also identify generic inter and intra-domain relationships between these entities, leading to a more expressive and flexible model that is domain independent and well suited for representing problem formulations of designers with different expertise levels and creativity. We have used the model for coding protocol data in a formal predicate logic language (Answer Set Prolog).

## Introduction

The main objective of our research is to investigate how design problem formulation is related to creativity. We have conducted exploratory protocol studies with designers engaged in problem formulation. In order to analyze such empirical data and utilize further investigations, a formal representation and a more efficient data collection method are needed. A framework that satisfies these needs provides the level of detail for the discovery of patterns in formulating problems that lead to

M. Dinar (✉) · C. Maclellan · A. Danielescu · J. Shah · P. Langley
Arizona State University, Tempe, AZ, USA
e-mail: mdinar@asu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_28,

more creative outcome. In addition, the framework might provide intrinsic measures that correlate well to measures of design outcome, e.g., design quality.

In our previous work, we created our first models [1] by trying to group similar types of data fragments taken from exploratory protocols. We identified some entities in problem formulation such as function, component, and analogy. Instances of each entity appeared in order of emergence with unique tag names. Wherever the designers mentioned two or more instances of different entities in relation with each other, a line was drawn to show the link. In the second version [2], we modified the models to capture more specific relationships. We called both types of our models *P-maps*.

We are concerned mainly with two aspects of the framework we built our model in: the ontology, i.e. the entities, their attributes, and the relationships among them; and the formalism for representing the data. In defining the entities we looked at other well-known models in design research. One popular model is the Function-Behavior-Structure (FBS) model [3] and its several similar variants such as the Structure-Behavior-Function (SBF) model [4] and the Functional Representation [FR] model [5].

In an ongoing effort to capture pathways of problem formulation towards creative design, we have changed both the ontology and the notation in our *P-map* from the previous version. Even though our model shares many common features with FBS, it has many differences. These differences arise from potential ways of building upon FBS to be incorporated in new avenues in design research, especially pre-ideation stages, i.e. problem formulation. We focus on improving flexibility, representation of hierarchical structure, and expressiveness of the model. We use Answer Set Prolog (ASP) [6], a formal predicate logic language, for representation.

## Function-Behavior-Structure Models

FBS is an established framework in design research that has prompted a great body of work within the past two decades since its introduction. The concepts of the model, namely the main three entities and the relations among them, were not unfamiliar themes in engineering design. Around the same time Gero [3] published his seminal paper, Gui [7] proposed a scheme for representing mechanical components and assemblies that used predicate logic to describe function, behavior and structure in an object model.

The extensive work related to FBS in engineering design research deserves a dedicated survey to the subject. However, a few useful critiques of the model help to that end. Dorst and Vermaas [8] take a dialectical stance and follow the changes in the definitions in the FBS model in three papers of Gero's work. They search for distinctions between intentional descriptions of artifacts and structural descriptions, and conclude that the changes proposed to improve or clarify the definitions were actually prompted by changes in the objectives that the model was supposed to serve. They argue that the shift from a descriptive model to a prescriptive model was a source of ambiguity. Gero did not seem to claim that his model laid focus on either of the directions in his research. If the model can be used to both ends, that should be considered an advantage of it. In our research we have tried to create a flexible model of design, however, we too do not claim that our model can be used in all phases of the design process, for all types of design problems, or solely for descriptive or prescriptive models.

Galle [9] finds the modified definitions that Dorst and Vermaas propose incomplete because they refer to an artifact that does not yet exist. Although he supports the separation of intentional and non-intentional concepts and the addition of *purpose* in their modified FBS, he offers two distinct definitions for FBS. On the one hand he defines a *nominalist* model that is purely logical and symbol-oriented. On the other hand the *realist* model sees design and thus FBS as a mathematical abstraction of experiential knowledge about the behaviors that embody structure in terms of material objects.

Besides FBS, other variants of the model have been developed independently. The most noteworthy of these variants are the FR and SBF models. Chandrasekaran [5] developed FR as a language which described the function of an artifact in terms of causal processes in order to simulate, diagnose or explain how the artifact works. In his later work with Josephson [10], he added an environment-centric view in addition to the original device-centric view to allow for more precise ways of representing design knowledge. In describing how FR can benefit from attempts in finding a common functional modeling (FM) ontology [11, 12], he argues that artificial intelligence research that aims to model artifacts can lend a more formalized representation to FM [13].

Goel et al. [4] have developed the SBF modeling language for a teleological description of complex systems. In this language, structure, behavior and function are represented in terms of components and their connections, transitions among a sequence of states, and pre- and post-conditions respectively. The syntax is similar to notations that are used to represent production rules. The model is a top-down description scheme, in which each fragment of the model is defined by a lower level fragment. At the top there is an instance of SBF while at the bottom there are the building block fragments such as strings and integers. For example, an *element* (or in other words a component in a structure model) is defined by an integer *Id*, a string *name*, a string *description*, an optional set (can have zero number of fragments) of *property*, and an integer *subelement Id*.

FBS has mainly been used in modeling the design process, protocol analysis, and design automation. Gero [3] identified activities in the design process in terms of transformations from one of the three domains to another, considering a difference between expected and actual behavior. For example, transforming a function to an expected behavior is considered formulation or specification. Gero and Kannengieser [14] took into account the dynamic character of design by considering the notion of situatedness. They extended FBS further to the explicit modeling of design processes [15]. Howard et al. [16] adopted FBS to describe a creative design process proposed by integrating engineering design process views and creativity from the standpoint of cognitive psychology.

FBS has been used as an attempt to create a common consensus among design researchers in defining fundamental concepts, it has also been used as a coding scheme in analyzing protocol studies [17–19]. The application of such ontological description has been extensively used in developing design automation. Umeda et al. [20] proposed a computer tool Function-Behavior-State Modeler for supporting synthesis in the conceptual design stage. Anthony et al. [21] integrated engineering knowledge as a description of function and behavior with a semantically annotated representation of 3D graphical models. For a list of some design automation systems that have used SBF see Goel et al. [4]. Maher and Gomez de Silva Garza [22] also provide examples of application of FBS in case-based reasoning tools in design.

## Aspects to Consider in Improving the Model

Inspired by the FBS framework, we realized that we should consider a few aspects to change in order to capture the problem formulation space more effectively. In this section, we explain why FBS needed flexibility, hierarchy and expressiveness.

### Flexibility

Dorst and Vermaas [8] argue that the changes in the definition of the entities in FBS were made mostly to accommodate a distinction between an intentional and a structural purpose of a design. One might argue that adhering to such distinction may not offer significant advantages in defining reasoning mechanisms. Moreover, design activities are often separated from manufacturing. It is convenient to assume design ends with a document that specifies design, as Gero assumed in [3]. However, regardless of the difference in how intended or actual behavior achieves a function, not all of what is expected from a design can be solely defined in terms of its functions. Consider the example of designing an artifact that moves a load from one point to another: if one ignores the conditions of the environment, load capacity and distance should still be specified. Such requirements are obviously, not functional requirements.

Umeda et al. [23] stated that function, structure and behavior represent a high to low level of abstraction, respectively, that is, the emergence of behavioral descriptions follow that of the structure and function monotonically and away from abstract concepts. However, this is not always the case in design practice. For example, a behavior is not always expressed in terms of rigid mathematical equations. Sometimes the designers' knowledge resembles qualitative physical expressions. For example the designer may only be aware of an existing relationship among different parameters without knowing the specifics of that relation. A more flexible model allows the abstraction of each of these entities.

Additionally, in real life experience, design data fragments do not appear monotonically. If there were attributes that defined the entities in FBS models, rarely are these attributes completely filled once a new instance of an entity is added. In the SBF modeling language [4] there is an extensive description of each entity and some of the attributes are optional sets. However, it seems the addition of new attributes in an instance of an entity alters the initial object entirely.

## Hierarchy

The ability to flexibly represent a hierarchical structure is instrumental to modeling complex and evolutionary systems [24]. Products are getting ever more complex and design is inherently an evolutionary process [25, 26]. In addition, hierarchies often help define levels of abstraction. Design studies also agree that designers move among different levels of abstraction intermittently during design [27, 28]. As explained before, the need for flexible representation of different levels of abstraction is necessary in capturing problem formulation data. Abstraction is not limited to different entities but also in a hierarchy of any of the entities. The hierarchical structure spans all the main entities. Requirement elicitation is as common an approach in design as functional decomposition is. Artifacts especially in complex and adaptable designs have a hierarchical structure, where sometimes components are independent modules. Therefore a model in the FBS framework that is restrained to function, behavior, and structure descriptions without representing compositions and multiple levels of abstraction at each entity, is not sufficient.

## Expressiveness

Gero and Kannengieser [29] state that FBS ontology is a high level model. FBS still can benefit from a more elaborate definition of attributes to enhance its expressiveness. Parametric relations for example can be described in more detail in behavior at different levels of abstraction, i.e. in terms of mathematical expressions, qualitative models, or unknown relations as explained before.

As stated earlier there is a need for a more flexible model to explicitly express non-functional requirements. Other details such as the importance of the requirements (e.g. in terms which are often used such as musts and wishes), their sources (e.g. customer needs, regulations, conditions of the operating environment), and the mating conditions among components help improve capturing more of the formulation space and thus open potential ways for the discovery of paths to creative outcome.

## Our Proposed Model

To address the points we made above, a hierarchical representation is introduced not only in each of the three domains (F, B, S) but in additionally proposed domains, which are requirements and issues. We use the term *artifact* instead of *structure* because the latter has a connotation to a common characteristic of all of the main domains in the model for which we use the term *hierarchy*.

Although modeling design processes is not our objective, it is convenient to explain our model following a typical design process. In developing novel products, designs usually start with explicitly specified requirements from customers or marketers, which include what the artifact is supposed to do (its main or highest level function); the performance levels desired (technical specifications) and overarching design goals. To that, the designer applies his domain knowledge which includes procedural knowledge (functional decomposition, search strategy, etc.) and artifact knowledge (candidate solutions, physical laws governing behavior). From an analysis of the requirements the designer may gain key insights, particularly in the discovery of problematic issues. This is consistent with our earlier observations [1, 2].

From these observations we can say that at the most general level, problem definition elements can be grouped into the following categories: *Requirement*, *Function*, *Artifact*, *Behavior*, and *Issue*. The groups are built around base entities and their hierarchical structure in terms of upper-lower level, and preceding-succeeding entities. The base entities in functions, requirements, and issues share the name of the group. In addition to the hierarchies and the directional (sequential) or non-directional relations among them there are supporting entities such as parametric relations under the group *Behavior*. All groups except the *Issue* group are related with bidirectional links, while the *Issue* group can have a relation to any combination of the rest of the entities. Before we explain the details of our model any further we should talk about the formalism that we used for our representation. Notice that entity names are italicized in lower case while group names are in upper case.

## The Formalism

The improved P-map model has been utilized to code protocol data, which necessitates the use of a consistent formalism. The formalism that we chose to encode P-map data was Answer Set Prolog [6], a non-monotonic declarative logical programming language that is based on Answer Set semantics [30]. Similar to Prolog, Answer Set Programs consist of two main components: facts, which are the ground literals over which the system reasons, and rules, which are used to perform logical reasoning over the facts. An Answer Set Program solver finds all of the answer sets which are consistent with by the given set of facts and rules. On the one hand, our answer set representation is a simple way to codify protocol data. On the other hand, it provides a means to reason about the coded *P-maps*. For this paper, we focus solely on representing protocol data in a concise way. Our primary reasons for choosing Answer Set Prolog were the simplicity of the logical formalism, the direct mapping between the *P-map* and an Answer Set representation, and the ease with which we could perform automated reasoning over the represented *P-map*.

The main ingredient of the answer sets are the predicates. A predicate is the name of a relation whose values—shown inside braces—are the related elements. Since the ASP representation is declarative, only the mandatory attributes of an entity reside in the predicate that define it. For example, a *solution principle*, which is one of the entities in the *Artifact* (S in FBS) group, is represented as:

solutionPrinciple(<Id>,<description>)

When used, the actual values would be substituted into the *Id* and *description* spots. The IDs (which are unique) start with two letters corresponding to an entity. In an example that shows a design of an airplane seat that can be automatically turned into a bed, a solution principle may look like this:

solutionPrinciple(sl_telescope,"pads inserted into one another come out of the seat")

As another example, consider the *function*, which is the main entity of the *Function* group. To keep the model simpler, the Ids are attributes that may represent the entity in their own right. The Id for a function is taken as its action verb:

function(fn_moving_to_flat_position)

In addition to these main components, there are a number of supporting predicates, some of which are common among several groups. The hierarchical structure is defined by:

parentOf(<parent>,<child>,<hierarchy>)

Sequences are defined by:

before(<preceding>,<succeeding>)

A functional decomposition for the airplane seat example is presented as follows:

function(fn_supporting_sleeping)
function(fn_moving_to_flat_position)
function(fn_supporting_in_flat_position)
parentOf(fn_supporting_sleeping,fn_moving_to_flat_position,hy_fn1)
parentOf(fn_supporting_sleeping,fn_supporting_in_flat_position, hy_fn1)
before(fn_moving_to_flat_position,fn_supporting_in_flat_position, hy_fn1)

The hierarchy Id is used to allow for disjunctive decompositions and make the model more flexible.

## The Entities

As explained in the introduction of our proposed model, each group has a base entity and some of the other entities are compositions of the base entities in a hierarchical structure. In the *function* group, the hierarchy manifests functional decompositions. Our model incorporates disjunctive composition, making it possible to have multiple functional decompositions using common sub-functions.

In the *artifact* group, the product architecture is a hierarchy of physical embodiments and solution principles. Our model allows partial compositions of solution principles and physical embodiments, since, in reality, the designer follows different parts of the sub-solutions at different times corresponding to different levels of abstraction. Similarly, in the *requirement* group, requirements and goals can participate in the same hierarchy.

In the *behavior* group, a physical effect is a hierarchy of physical laws. Physical effects may be expressed by parametric relations, which are composed of sets of parameters. In the *issue* group, the hierarchy entails the priority the designer gives to the issues that should be addressed, that may correspond to their problem solving strategy. The ontology of our *P-map* model can be seen in Fig. 1 in terms of Barker notations [31].

As can be seen, the names given to the hierarchies are different in some groups to reflect common terminology in design corresponding to the same data structure. In the *Artifact* group, the *product architecture* is the hierarchical structure accommodating a composition of physical embodiments, i.e. components, and solution principles. The *connection* entity represents mating conditions among components and since it is a non-directional sequence, a different predicate from the *before* predicate, defined as connects (<em_Id/sl_Id>,<em_Id/sl_Id,hy_Id>), is used for it.

We explain the additional entities to the FBS model, i.e. *requirement* and *issue*, and the inter-group relations in more detail.

## Requirements

Different definitions and interpretations of FBS have tried to place the intention or the purpose of the design along the lines of any of the three domains [8]. Design problems, however, are different in that they do not always start with a known functional decomposition that requires a search for components for each function, or with known components that should be adapted to accommodate new functions. Requirements, especially in the design of novel products for specific customer needs, are explicitly identified by marketers, senior technical officers, or the designers themselves.

A design problem is usually given as a design brief or problem statement. Problem formulations and solution candidates co-evolve and a change in one can

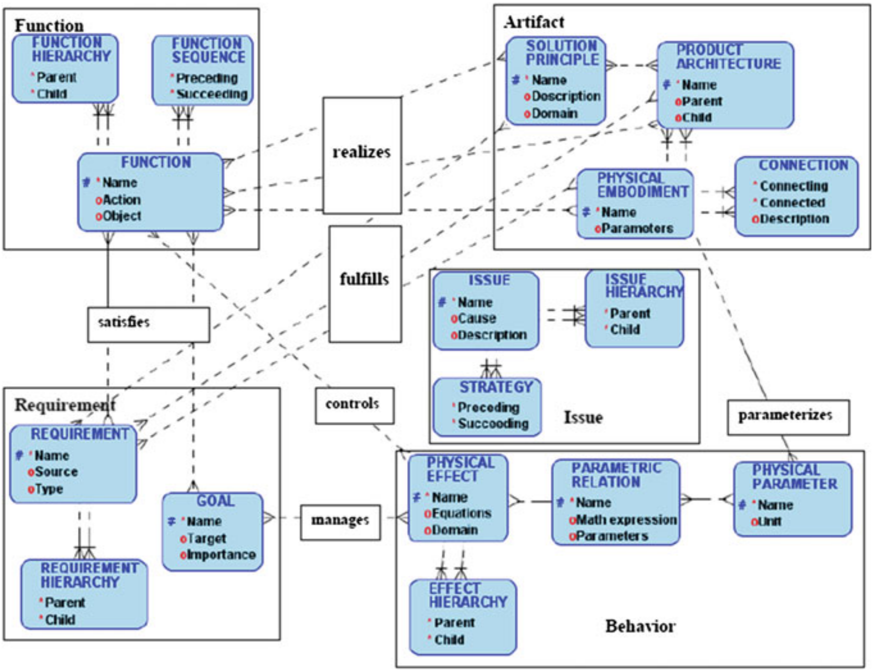


**Fig. 1** The ontology of the *P-map* model

lead to a change in other. For example, proposing an artifact to satisfy a requirement can lead to introduction of a new requirement. Therefore, additional requirements are elicited from existing requirements or triggered by the addition of other elements. The problem is specified by a set of design goals and requirements. A desired design is one that realizes all the requirements. A more desired design is one that achieves better design goals. In our model, the set of design requirements and goals is defined as follows:

requirement(<Id>)
requirementSource(<rq_ Id>,< source>)
goal(<Id>)
goalImportance(<gl_ Id>,< importance>)
goalTarget(<gl_Id>,<target>,<improvement_direction>)

The source can be defined by specific tag names that might be set for the user to choose from: “safety”, “aesthetics”, “ergonomics”, “use_environment”, “affordance” etc. The source might also specify whether the requirement was explicitly given in the problem statement or whether it was discovered by the designer.

Requirements and goals are subtypes of the *requirement* entity. We assume that requirements are binary; they are either met or not met. Goals on the contrary are defined with their importance levels and the targets with ranges of values that should be achieved. They might have a single bound or be discrete. There is usually a relation between the level of satisfaction in using the solution and the degree to which the goal is achieved in terms of a utility function. The improvement direction can take one of the three values: "more", the goal is desired to be more than the given target; "less", the goal is desired to be less than the given target; "about", the goal is desired to be as close to the given target as possible.

## Issues

An issue is a point that the designer believes to be pivotal or problematic in achieving a design objective. An issue can arise in realizing a function with a specific artifact or behavior, in realizing conflicting design goals such as lower weight and strength of a structure or in accommodating different components in a product architecture due to incompatible interfaces to name a few. Issues, therefore, can be seen as intimately related to the other domains. The designer gains insight in the discovery of key issues in the design and the areas of the design that should be prioritized. The hierarchy of issues may support particular problem solving strategies. To avoid repeating the same example, we show this hierarchal structure in the next section when we talk about the relations. Entities and their attributes in addition to corresponding entities from FBS are given in Table 1.

## Distinct Relations

Generic inter and intra-domain relationships between these entities are also identified in addition to optional attributes of the entities. This leads to a more expressive and flexible model that is domain independent and well suited for representing problem formulations of designers with different expertise levels and creativity. Table 2 shows the inter-group relations among entities.

Issues can be related to any combination of other entities. For the airplane seat example, design issues in our model can be represented as the Id for an issue starts with iu and the Id for a physical behavior starts with ph):

issue(iu_comfortable_support_at_flat_position, "deflection")
issue(iu_covering_length_at_flat_position,"increasing number of boxes")
issue(iu_support_weight_at_flat_position, "load on a cantilever causes high bending stress")
relates(iu_comfortable_support_at_flat_position, fn_supporting_in_flat_position)
relates(iu_covering_length_at_flat_position,
rq_length_of_seat_cushion_less_than_20in)
relates(iu_covering_length_at_flat_position, sl_telescope)

**Table 1** A summary of entities and their attributes

| Group | Entities | Attributes | Corresponding to FBS |
|---|---|---|---|
| Requirement | Requirement | Id, importance, source, domain | Purpose/Function |
| | Goal | Id, importance, target, direction | |
| Function | Function | Id, parameter | Function |
| Artifact | Solution principle | Id, description, domain | Structure |
| | Physical embodiment | Id, parameter | |
| | Parameter | name, unit, value | |
| Behavior | Behavior | Id, domain | Behavior |
| | Parametric relation | Id, equation, abstraction level, parameter | |
| Issue | Issue | Id, description | ———— |

**Table 2** Inter-group relations

| Entity | Relation | Entity |
|---|---|---|
| Function | satisfies | Requirement |
| Artifact | fulfills | Requirement |
| Behavior | manages | Requirement |
| Artifact | realizes | Function |
| Artifact | parameterizes | Behavior |
| Behavior | controls | Function |
| Issue | relates | All entities and their combinations |

relates(iu_support_weight_at_flat_position, ph_bending_stress)
relates(iu_support_weight_at_flat_position, rq_support_250lb_weight)
relates(iu_support_weight_at_flat_position, sl_pivoting_recliner)
parentOf(iu_comfortable_support_at_flat_position, iu_covering_length_at_flat_position, iu_hy1)
parentOf(iu_comfortable_support_at_flat_position, iu_support_weight_at_flat_position, iu_hy1)
before(iu_support_weight_at_flat_position, iu_covering_length_at_flat_position, iu_hy1)

## Discussion

As noted earlier, the main objective of our research is to discover the relation between problem formulation and creative design. Design is an important type of human problem solving [32]. Newell and Simon [33] described a computational framework for studying cognitive aspects of human problem solving in terms of states, operators and goals. Although it is plausible to create models to investigate all cognitive aspects of design, it might be more efficient to focus on parts of them. We have focused on modeling the mental states that arise during problem formulation, and we believe that our *P-map* model can be useful for this purpose.

In real life, design is not monotonic or procedural. At each step, the designer may think about new solution principles, use alternate behaviors and corresponding functions, or add new requirements. A benefit of a declarative representation is that elements can be added and deleted without worrying about how they fit together.

## Uses of the *P-map*

Protocol analysis is time consuming and is not easy to implement for extended traces. Thus we want to utilize our data collection and analysis method. Additionally, we are searching for pathways that lead towards creative design by comparing problem formulations of designers of different expertise and creativity levels. We are building an interactive tool for collecting data about solving design problems in our *P-map* format. The tool is not the focus of this paper.

In our earlier work [1, 2], we have tried to find a representation that captures differences among how designers formulate problems in terms of changes in the representation over time, i.e., formulation states. At this point, we are not making any hypotheses about the relation between creativity and formulation. *P-map*, however, can help test new hypotheses by providing a fine level analysis tool. Some possible examples, inspired by our earlier findings [2] may point to patterns in the sequence in which solution principles and parameters emerge. Another example concerns the coverage of issues; one might argue that the fewer entities are related to an issue, the less it is justified, leading to less creative designs or an unnecessary inhibition.

When encoding P-maps in an Answer Set Prolog formalism, all of the various entities and connections between them are represented as facts. A given representation of a *P-map* by itself will contain no rules. This means that, when querying the set representation of a *P-map* with a solver, only one answer set is returned, the answer set containing the facts that we gave our system. In future work, rules could be developed that govern how the system should reason over the *P-map*. This will result in more interesting behavior when executing the answer set program. An example would be computing the number of different solution principles that have mathematically expressed behaviors that satisfy all the requirements. Another example would involve determining the number of unresolved issues that are inhibitory. This means that the designer considers an issue in a short-sighted manner, e.g., in satisfying a requirement by a solution principle without understanding that there is a related behavior that should be examined as well.

Ultimately, we can develop intrinsic measures within the framework which we find correlate well with design outcome. Such measures can be graph-based, corresponding to our graphical representation. One example is the proportion of closed cycles of *n* degrees (from two to five, the number of entity types in our framework) to all connections. Other intrinsic measures can relate to the sequence of emerging entities or the proportion of instances of an entity to all instances.

## Examining and Validating the *P-map*

We have proposed a framework to capture problem formulation in design in the form of a sequence of state representations. There are two aspects representing models within the framework, formalism and content. Developing formalism imposes requirements on a data model. We describe some of the requirements that we set for our data model.

## Domain Independence

The model should not be limited to representing a specific class of objects. For example, it should accommodate the design of a combustion engine with its known behaviors, or an engine with a general function of providing power, including but not limited to a solar-powered engine.

## Compactness

The model should have a simple and compact representation. This is a relative measure but it provides hints for including some entities with similar properties in the same group or class. For example, we consider safety and ergonomics as types of requirements.

## Richness

We want to show creative vs. non-creative designer problem formulation and we assume we can capture that in comparing their *P-maps*. Thus our data model should not only be compact enough to be easily created or translated, but also it needs to be rich enough to provide such contrasts.

## Unambiguity

We need a formalism that can be communicated among interpreters who will code problem formulation data in design. Though there may be differences in labeling the data, a design episode should lead to close enough P-maps for different coders. Thus the entities should be semantically distinct. This also gives the possibility of automatic coding, and interactive conversation for our tentative aid software tool.

## Flexibility

Our model should accommodate incomplete or redundant data structures. However, the tentative aid software should be able to detect such patterns. For example two functional hierarchies with a common parent and common children may coexist (F1 is a parent of f2, f3; F1 is a parent of f2, f3, f4). In addition, the model should be able to hold relations among entities, representing different levels of abstraction. For example a function can be related to a physical embodiment or a product architecture including that physical embodiment.

We refer to entities in the same way that others refer to classes in Object-Oriented data models. The notions of tables and joints in Relational Database models can be easily mapped into Answer Set Prolog. For example, we want to capture different instances of function sequences or product architecture. We represent such instances with an entity that is represented as a composition or aggregation in an Object-Oriented model and as a relation table in a Relational Database model.

We then assign attributes to these entities. Some of the entities in our older models can now be defined as attributes which helps achieve compactness. For example a physical rule (either in an abstract form or as a mathematical equation) may be considered an attribute of a behavior.

To validate these data model requirements, we will measure the expressiveness of the model by examining how much of protocol data can be expressed in terms of the entities and their relations. Additionally, we will measure inter-rater agreement; we check how different two raters analyze a given protocol.

## Conclusion

In order to investigate the relationship between design problem formulation and creativity, we have created data models within a framework which we call *P-maps*. In defining its entities we looked at other models in design research. We were inspired by the Function-Behavior-Structure (FBS) model and its several variants such as the SBF model and the Functional Representation model. The *P-map* shares many common features with FBS, although it has many differences. Similarities include the common function, behavior and structure (artifact) group of entities and the way behaviors link functions and structures. The differences are: allowing disjunctive hierarchical compositions; the explicit declaration of requirements; the consideration of issues designers face in meeting requirements, achieving functions, controlling behaviors, or in any combination of these states; and the distinct relations among the groups. These differences between our *P-map* and FBS provide ways of improving flexibility, representation of hierarchical structure, and expressiveness of FBS.

The ability to flexibly represent a hierarchical structure is instrumental to modeling complex and evolutionary systems. In addition, hierarchies often help define levels of abstraction. FBS can still benefit from a more elaborate definition of attributes to enhance its expressiveness. Parametric relations for example can be described in more detail in behavior at different levels of abstraction, i.e. in terms of mathematical expressions, qualitative models, or unknown relations.

To address these points, we introduced a hierarchical representation not only in each of the three domains (F, B, S) but in the additional domains of requirements and issues. Different definitions and interpretations of FBS have tried to place the intention or the purpose of the design along the lines of any of the three domains. Design problems, however, are different in that they do not always start with a known functional decomposition, or with known components that should be adapted to accommodate new functions. Requirements, especially in the design of novel products for specific customer needs, should be explicitly identified.

An issue is a point that the designer believes to be pivotal or problematic in achieving a design objective. An issue can arise in realizing a function with a specific artifact or behavior, in realizing conflicting design goals such as lower weight and strength of a structure or in accommodating different components in a product architecture due to incompatible interfaces, to name a few. Therefore, issues can be seen as intimately related to the other domains.

We identified generic inter-domain and intra-domain relationships between these entities in addition to optional attributes of the entities. This led to a more expressive and flexible model that is domain independent and well suited for representing problem formulations of designers with different expertise levels and creativity.

The model has been used for coding protocol data which necessitates the use of formalism. Answer Set Prolog as a formal predicate logic language was used for coding data.

The designer at each step of the design may think about new solution principles, using alternate behaviors and corresponding functions, or add new requirements. A benefit of implementing a declarative computational framework for our representation is that partial data fragments can be added and deleted without worrying about the consistency of the P-map.

We are building an interactive tool for collecting data about solving de-sign problems, in our P-map format. We believe that our *P-map* model can help testing hypotheses about the relation between formulation and creativity by providing an analysis tool at a fine level of detail. Some possible examples, inspired by our earlier discoveries, are finding the sequence of emergence of solution principles, and checking how issues are covered in relation to other entities.

**Acknowledgments** This study is supported by the National Science Foundation, CMMI grant number 1002910. The opinions expressed in this paper are those of the authors and are not endorsed by the National Science Foundation. We thank Glen Hunt for his contributions to the earlier versions of our *P-map* and his work on building the interactive computer tool.

## References

1. Dinar M, Shah JJ, Langley P, Hunt GR, Campana E (2011) A structure for representing problem formulation in design. In: Culley SJ, Hicks BJ, McCloone TC, Howard TJ, Chen W (eds) Proceedings of the 18th international conference on engineering design, pp 392–440
2. Dinar M, Shah JJ, Langley P, Campana E, Hunt GR (2011) Towards a formal representation model of problem formulation in design. In Proceedings of ASME DETC, pp 263–327
3. Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36
4. Goel AK, Rugaber S, Vattam S (2009) Structure, behavior and function of complex systems: the structure, behavior, and function modeling language. Artif Intell Eng Des Anal Manuf 23(1):23–35
5. Chandrasekaran B (1994) Functional representation: a brief historical perspective. Appl Artif Intell 8(2):173–197
6. Gelfond M (2008) Answer sets. In: van Harmelen F, Lifschitz V, Porter B (eds) Handbook of knowledge representation. Elsevier, Amsterdam, pp 285–331
7. Gui JK (1990) A Function-Behaviour-Structure machine design model and its use in assembly sequence planning. J Eng Des 1(3):239–259
8. Dorst K, Vermaas PE (2005) John Gero's Function-Behaviour-Structure model of designing: a critical analysis. Res Eng Des 16(1–2):17–26
9. Galle P (2009) The ontology of Gero's FBS model of designing. Des Stud 30(4):321–339
10. Chandrasekaran B, Josephson JR (2000) Function in device representation. Eng Comput 16(3–4):162–177
11. Stone RB, Wood KL (2000) Development of a functional basis for design. J Mech Des 122(4):359–370
12. Hirtz J, Stone RB, Mcadams DA, Szykman S, Wood KL (2002) A functional basis for engineering design: reconciling and evolving previous efforts. Res Eng Des 13:65–82
13. Chandrasekaran B (2005) Representing function: relating functional representation and functional modeling research streams. Artif Intell Eng Des Anal Manuf 19:65–74
14. Gero JS, Kannengiesser U (2004) The situated Function-Behaviour-Structure framework. Des Stud 25(4):373–391
15. Gero JS, Kannengiesser U (2007) A Function–Behavior–Structure ontology of processes. Artif Intell Eng Des Anal Manuf 21(04):379–391
16. Howard TJ, Culley SJ, Dekoninck E (2008) Describing the creative design process by the integration of engineering design and cognitive psychology literature. Des Stud 29(2):160–180
17. Gero JS, Mc Neill T (1998) An approach to the analysis of design protocols. Des Stud 19(1):21–61
18. Neill TM, Gero JS, Warren J (1998) Understanding conceptual electronic design using protocol analysis. Res Eng Des 10(3):129–140

19. Pourmohamadi M, Gero JS (2011) LINKOgrapher: an analysis tool to study design protocols based on FBS coding. In: Proceedings of the international conference on engineering design. Copenhagen, Denmark, pp 1–10
20. Umeda Y, Ishii M, Yoshioka M, Shimomura Y, Tomiyama T (1996) Supporting conceptual design based on the Function-Behavior-State modeler. Artif Intell Eng Des Anal Manuf 10(4):275–288
21. Anthony L, Regli WC, John JE, Lombeyda SV (2001) An approach to capturing structure, behavior, and function of artifacts in computer-aided design. J Comput Inf Sci Eng 1(2):186–192
22. Maher ML, Gomez de Silva Garza A (1997) Case-based reasoning in design. IEEE Expert 12(2):34–41
23. Umeda Y, Takeda H, Tomiyama T, Yoshikawa H (1990) Function, behaviour, and structure. Applications of artificial intelligence in engineering V. Computational Mechanics Publications and Springer, Berlin, pp 177–194
24. Simon HA (1996) The sciences of the artificial. MIT Press, Cambridge
25. Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem–solution. Des Stud 22:425–437
26. Maher M, Poon J, Boulanger S (1996) Formalising design exploration as co-evolution: a combined gene approach. Advances in formal design methods for CAD: In: Proceedings of the IFIP WG5.2 workshop on formal design methods for computer-aided design
27. Eisentraut R, Gunther J (1997) Individual styles of problem solving and their relation to representations in the design process. Des Stud 18:369–383
28. Akin O, Chengtah L (1996) Design protocol data and novel design decisions. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, Chichester, pp 35–63
29. Gero JS, Kannengiesser U (2007) Locating creativity in a framework of designing for innovation. In: León-Rovira N (ed) Trends in computer aided innovation. Springer, Boston, pp 57–66
30. Gelfond M, Lifschitz V (1988) The stable model semantics for logic programming. In: Kowalski RA, Bowen KA (eds) Proceedings of the fifth international conference on logic programming. MIT Press, Massachusetts, pp 1070–1080
31. Barker R (1990) CASE method: entity relationship modelling. Addison-Wesley Longman Publishing Co. Inc, Boston
32. Goel V, Pirolli P (1992) The structure of design problem spaces. Cogn Sci 16(3):395–429
33. Newell A, Simon HA (1972) Human problem solving. Prentice-Hall, Upper Saddle River

# Functional Design Space Representations for Lead Qualification Situations

**Julian R. Eichhoff and Wolfgang Maass**

**Abstract** For businesses offering complex customized solutions the capability of their sales force to engage in problem-solution discovery is a crucial success factor in selling. In this context we investigate the application of functional representations to model design spaces related to situations where a salesperson is screening for potential customers (lead qualification). Therefore we present a conceptual approach on how to cast functional representations in the domain of lead qualification. We propose computational design space representations based on probability theory that take account for the uncertainties inherent in lead qualification. And we show results from a case study in which we test the practicability of the presented approach.

## Introduction

In industries that offer customized products and services, which meet their customer's individual business needs, vendors are often required to employ a consultative sales strategy called "solution selling" (cf. [1]). It comprises mainly four interdependent processes carried out on a per project basis: requirements definition, customization and integration, deployment and post-deployment support. The groundwork for these processes is laid by the vendor's sales force screening for potential customers (leads) and assessing their willingness and ability to buy a solution. This task is termed "lead qualification".

---

J. R. Eichhoff (✉)
Furtwangen University, Furtwangen im Schwarzwald, Germany
e-mail: julian.eichhoff@hs-furtwangen.de

W. Maass
Saarland University, Saarbrücken, Germany

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_29,

Lead qualification in solution selling industries reflects a critical and well-known problem, where the customer isn't fully aware of its needs and the vendor doesn't exactly know what to offer. In this situation both sides jointly explore the design domain while trying to minimize the investigations needed to gain a shared understanding. However, the usual approaches to this problem are not applicable to lead qualification: Organizing a series of collaborative design workshops would be too expensive to be carried out for every potential customer. And further, requirement lists that are available at this stage are most often incomplete and ambiguous. Therefore the salesperson himself is required to take the role of a designer and estimate the fit between lead and vendor.

Well, this in turn is highly dependent on a salesperson's individual knowledge of a lead's needs, of products and services offered by the vendor and its partners, and of how certain bundles of products and services may be used for problem solving, we term this design knowledge. Using his design knowledge the salesperson starts a dialogue with the lead to explore the design domain. This pretty much resembles Schön's notion of having a conversation with the design situation [2]: a cyclic process of "seeing-moving-seeing", i.e. acquiring and interpreting information from the lead about the design situation (seeing), followed by performing changes on the conceptualized offer and making these explicit to the lead (moving), and gaining further insights on the design situation by rediscussing the lead's feedback (seeing).

A designer constructs the required connections between problem, solution and its realization through experience. However, salespersons may lack this experience for several reasons: Especially external salespersons are not directly involved in product development at the employing vendor, and thus may have narrow insights on how their work affects downstream processes. Experiences from other salespersons may not be considered, due to limited reporting or inconsequent knowledge reuse. And limited possibilities or rigid policies for inter-organizational communication may exclude design insights from partnering organizations.

To overcome these shortcomings in intra- and inter-organizational design knowledge reuse for lead qualification, we suggest the use of a formalized design space representation, which can be accessed by services that support the conceptual design tasks carried out by a salesperson during lead qualification. In this paper we present a conceptual approach and computational representations of such design spaces that specifically address the uncertainty inherent in a salesperson's picture of a lead. Results from a case study in the field of office fit-out projects underline the practicability of the approach but also highlight further issues.

## Related Work

The literature on design research holds several approaches to represent the reasons behind design decisions made during a design process, the so-called design rationale. Existing approaches can principally be categorized in process-oriented and feature-oriented representation methods [3]. Instead of capturing the history of design processes, feature-oriented methods focus on representing designed artifacts. Functional representations are one prominent type of feature-oriented representations [4]. Yet a number of functional concept ontologies have been proposed that all share the idea of describing a design object in the context of its purpose to solve a problem or show a wanted effect [5]. Such representations seem especially appropriate for describing design spaces of lead qualification situations as they intrinsically support the idea of drawing a conceptual link between a lead's problem space and the solution space that may be offered by a vendor. Though functional concept ontologies have been applied to domains of conceptual design [6], to the best of our knowledge there exists no concise approach on how to cast functional representations in the domain of lead qualification.

Related problems to lead qualification are contractor pre-qualification and the bid/no-bid problem. Contractor pre-qualification views the problem from the lead's perspective. It deals with measuring the capabilities of potential vendors with respect to a given procurement task (cf. [7]). The bid/no-bid problem in turn is a task in organizational selling that follows lead qualification. After having assessed enough information about a lead's problem situation the vendor often needs to decide whether it is economically feasible to attend in a tender process (cf. [8]). So far there has been no approach in the field of contractor pre-qualification or the bid/no-bid problem that makes use of functional representations. However, especially the works in the field of contractor pre-qualification highlight critical representation issues, which apply to lead qualification as well: an appropriate representation should account for error-prone and subjective judgments of decision makers and it should be able to cope with noisy and uncertain data [7].

In a previous work [9] we presented a computational functional representation of the (situated) Function-Behavior-Structure (FBS) framework [10]. It is based on first-order probabilistic belief networks (cf. [11] ) in order to explicitly model uncertainties and thus address the mentioned representation issues. Like other models that operationalize the FBS framework [12–17] we provided a method to describe the components of a design object, their properties, and relations among both. In contrast to previous approaches we took especially account of the fact that the certainty of a design component being associated with an attribute may vary throughout the design process and we proposed a method to highlight needed information that would most useful for reducing uncertainty.

Similar to our work the parameter analysis approach of [18] describes how to reveal the customer's problem situation (need identification and analysis) and how to generate conceptual designs grounding on these insights (parameter analysis). However, there are fundamental differences: To promote the ideation of innovative solutions their methodology suggests to maximize the set of potential solutions and thus favors need statements that are least constrained, whereas our approach focuses on finding concrete offers as fast as possible and thus favors need statements that are most effective in narrowing the set of potential solutions. To avoid the generation of need statements that are not solvable, our approach directly integrates the problem and solution parts in a single representation. This allows for "micro" design cycles, where every added constraint is directly evaluated against the current set of potential solutions. A salesperson may then discuss complicating constraints just when they arise.

The contribution of this paper is an extension to the works of [9]. In the following we give a detailed description of the conceptual relations between functional representations and lead qualification, we provide additional methods for supporting design tasks during lead qualification, and we report our findings from an ongoing case study in the domain of office fit-out projects.

## Conceptual Approach

At the beginning of lead qualification the vendor has rather limited insights on the lead's problem situation, and vice versa detailed knowledge about solutions mostly resides exclusively at the vendor and its partner organizations. Thus the design space is ill-structured [19] and adequate offers cannot be directly identified. In this context a salesperson is asked to engage in problem-solution discovery, i.e. developing a clear idea of the lead's issues and formulating a battery of potential responses [20]. This is typically done in a series of sales calls, where the salesperson not only asks for problems to be solved but also proves whether the suggested solution candidates actually fit the lead's expectations. Here a suggested solution is a salesperson's individual projection of how a certain bundle of products and services will help to satisfy the lead's business needs at the time when it is implemented in the lead's organization.

However, time and resources for problem-solution discovery in lead qualification are limited. And if problem-solution discovery is carried out by a salesperson that does not have the required design knowledge lead qualification is likely to fail, because (1) the salesperson does not assess the information needed to narrow down the set of potential solutions or (2) the salesperson may draw wrong conclusions on the gathered information leading to inadequate solution candidates.

As mentioned above, our conceptual approach to support the identification of needed information and the selection of adequate candidate solutions is based on functional representations [4] that model the design space of possible solutions. However, especially interpretations of "function" vary between the proposed functional representations [5]. The discussion of [21, 22] provides clarification. We now cast their notion in our domain.

## Functional Design Space Representation

Let the world $\mathbf{W}$ be a set of variables[1] $\{W_1, W_2, \ldots\}$ that describe the aspects of a generic organization that may be affected by the products and services of a design object, and let $\alpha$ be a conceptual model that defines dependency relations over these variables.

Let the design object D be a set of design components $\{d_1, d_2, \ldots\}$, and let $R \subset D \times (D \cup \mathbf{W})$ be a dependency relation over the set of design components and the world, which denotes a structural relationship between two design components or between a design component and the world. Moreover, every design component $d_i$ is associated with a set of variables $\mathbf{C}_i = \{C_{i,1}, C_{i,2}, \ldots\}$ describing properties of the design component (note same index i). A subset of these properties describe the vendor's products and services used for realizing the design component, we refer to this subset as "offer". Further, let $\beta$ be a conceptual model that defines dependency relations over all design component properties.

Building on this we consider a design object as being a potential solution if it is integrated in the world, i.e. it is implemented in the lead's organization. To define how it is integrated let the mode of deployment $\gamma$ be a conceptual model that defines dependency relations between design component properties and world variables.

The lead's problem situation is defined as a set of constraints on the world and the design object. Let N, $F_W$, and U, represent sets of logical constraint functions on world variables $\mathbf{W}$, where a constraint function evaluates to true if the value of the variable in its domain meets the constraint (otherwise it evaluates to false). $F_D$ represents a similar set of constraint functions but over variables $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \ldots$ of design component properties. N, $F_W$, U, and $F_D$ differ in terms of abstraction from the design object. $F_D$ has the lowest abstraction and specifies desired behaviors of the design object itself. Still close to the design object, $F_W$ specifies desired effects of the design object's behavior on the behavior of the world in which it is embedded; this is also termed "function-as-effect" [21]. Finally, business needs N and usage constraints U of the lead have the highest degree of abstraction from the design object's structure. Together N, $F_W$, U, and $F_D$ are used to characterize the lead's organization and to set the goals and objectives of solution development.

### *Example*

Consider the following example from the domain of office fit-out projects, which is also graphically depicted in Fig. 1. Office fit-out projects deal with the design and

---

[1] Notation: upper-case letters represent variables (e.g. X), bold upper-case letters represent sets of variables (e.g. $\mathbf{X}$), lower-case letters represent value assignments to variables (e.g. x).

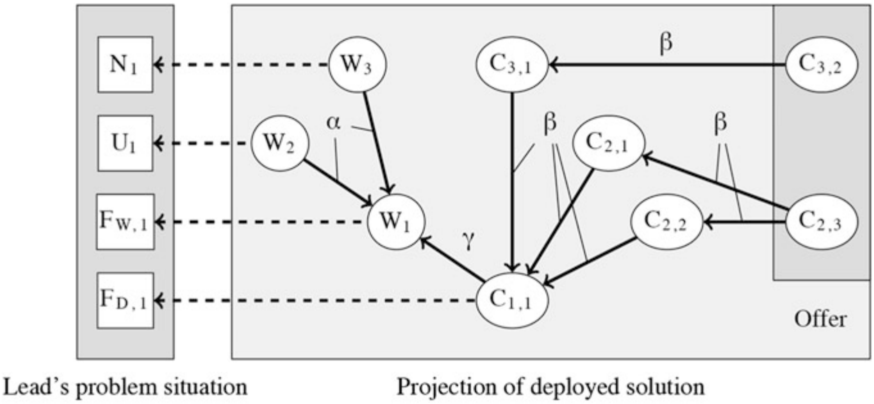


**Fig. 1** Exemplary design space

construction of scenery (interior elements such as ceilings, partitions and finishes) and settings (furniture and equipment) for office accommodation [23].

Office accommodation has several influences on the business operations of an organization (e.g. [24]). In this simplified example we describe the world **W** as follows:

- $W_1$: Workspace flexibility can be low or high.
- $W_2$: User mobility can be low or high.
- $W_3$: High office productivity can be mandatory or neglectable.
- Conceptual model $\alpha$:

– $W_2$, $W_3 \rightarrow W_1$: Workspace flexibility should be high if and only if high office productivity is a mandatory objective ($W_3 \rightarrow W_1$) and user mobility is high ($W_2 \rightarrow W_1$). In any other case workspace flexibility cannot be determined.

Design object D consists of three components: workspace $d_1$, furniture $d_2$, and partitions $d_3$. They stand in structural relation $R = \{(d_1, \mathbf{W}), (d_1, d_2), (d_1, d_3)\}$, i.e. the designed workspace will be implemented in the lead's organization ($d_1$, **W**), and the workspace is made up of workstation furniture ($d_1$, $d_2$) and partition elements ($d_1$, $d_3$). The following variables and conceptual models describe the components and their dependencies:

- Workspace $d_1$:

– $C_{1,1}$: Workspace layout can be territorial, semi-territorial or non-territorial.

- Furniture $d_2$:

– $C_{2,1}$: Noise reduction of partitions can be low, moderate or high.
– $C_{2,2}$: Adjustability of partitions can be low or high.
– $C_{2,3}$: Partition offer can be none, curtains, screens or walls.

- Partitions $d_3$:

- $C_{3,1}$: Adjustability of furniture can be low or high.
- $C_{3,2}$: Furniture offer can be system A or system B.

- Conceptual model $\beta$:

- $C_{2,3} \rightarrow C_{2,1}$: Noise reduction will be low if no partitions are included in the offer, moderate if curtains or screens are offered, and high in case of solid walls.
- $C_{2,3} \rightarrow C_{2,2}$: Adjustability of partitions will be low if walls are offered, and high in cases of curtains, screens or no partitions.
- $C_{3,2} \rightarrow C_{3,1}$: Adjustability of furniture will be low if system A is offered, and high in case of system B.
- $C_{2,1}$, $C_{2,2}$, $C_{3,1} \rightarrow C_{1,1}$: Semi-territorial and non-territorial workspaces require both a high adjustability of furniture ($C_{3,1} \rightarrow C_{1,1}$) and partitions ($C_{2,2} \rightarrow C_{1,1}$) as well as a high noise reduction of partitions ($C_{2,1} \rightarrow C_{1,1}$).

- Conceptual model $\gamma$:

- $C_{1,1} \rightarrow W_1$: When implemented in the lead's organization, workspace flexibility will be high if the workspace layout is semi-territorial or non-territorial and low in case of a territorial layout.

If we consider this design space with respect to a specific lead qualification situation where the lead's central need $N_1$ is high office productivity ($W_3$), we would constrain $W_3$ with $N_1(W_3)$ to be in state mandatory. Now the conceptual dependency models $\alpha$, $\beta$, and $\gamma$ allow us to identify constraints on a lower abstraction level and even identify candidate offers: $\alpha$ suggests that need $N_1$ will be satisfied if the external effect of the design object on the lead's organization is a high workspace flexibility ($W_1$) under the premise of the lead's organization having a high user mobility ($W_2$). However, we cannot be sure about $W_1$ unless we know the state of $W_2$. Therefore it would be important to assess the information (e.g. in the next sales call) whether user mobility ($W_2$) is high or low. Given that $W_2$ was constrained with $U_1(W_2)$ to be in state high, we can implicitly identify the constraint $F_{W,1}(W_1)$ that workspace flexibility should be high. In the other case, i.e. $U_1(W_2)$ was set to be low, workspace flexibility ($W_1$) is yet unknown and $F_{W,1}(W_1)$ needs to be assed explicitly. In this example we continue with workspace flexibility ($W_1$) being constrained to state high. Now considering the mode of deployment $\gamma$ the desired "function-as-effect" ($F_{W,1}$) can be achieved with a design object that has either a semi-territorial or non-territorial workspace layout ($C_{1,1}$), which leads to constraint $F_{D,1}(C_{1,1})$. In the end $\beta$ allows us to identify system B ($C_{3,2}$) in combination with curtains or screens ($C_{2,3}$) as an adequate offer, as they provide the needed adjustability ($C_{3,1}$, $C_{2,2}$) and noise reduction properties ($C_{2,1}$) for implementing semi- and non-territorial workspaces ($C_{1,1}$). See below for the complete list of identified constraint functions:

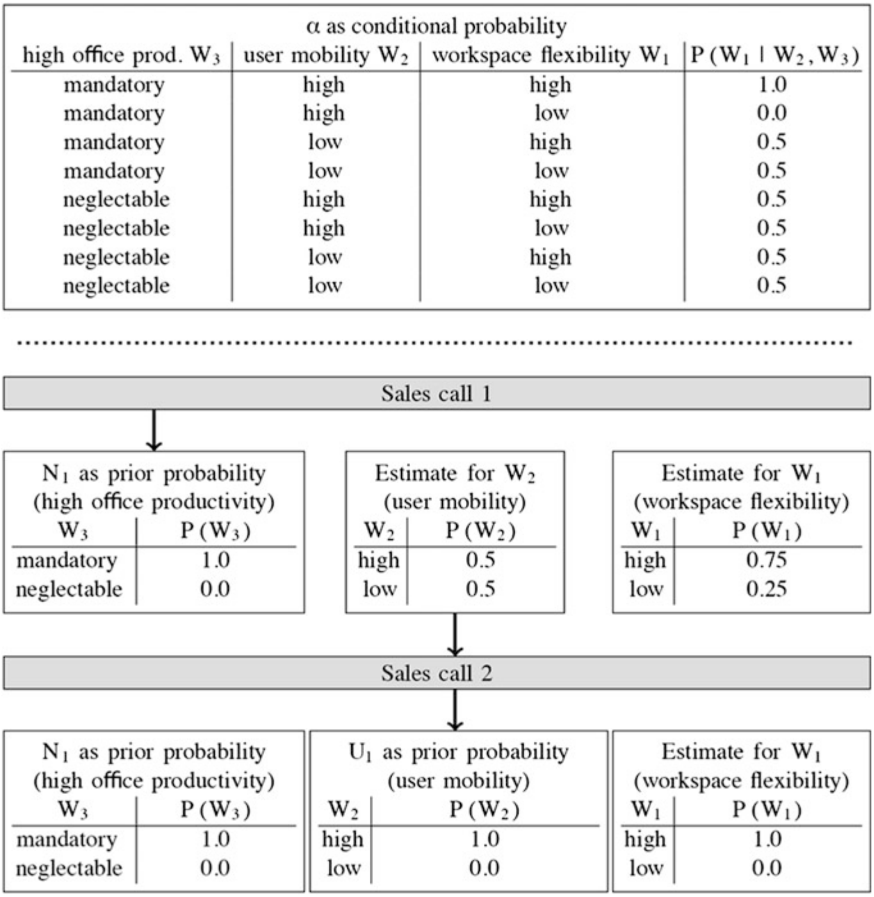
- $N_1(W_3)$ = {(mandatory, true), (neglectable, false)}
- $U_1(W_2)$ = {(high, true), (low, false)}
- $F_{W,1}(W_1)$ = {(high, true), (low, false)}
- $F_{D,1}(C_{1,1})$ = {(semi-territorial, true), (non-territorial, true), (territorial, false)}

## Probabilistic Functional Design Space Representation

The exemplified reasoning tasks can also be expressed in probabilistic terms. Probability theory enables us to explicitly address the open world assumption regarding a salesperson having incomplete knowledge about the actual design space (cf. [25]). The idea is to represent the solution part of the design space (design object embedded in the world) as a probabilistic belief network and represent the problem part as constraining prior probabilities on the random variables of the belief network [9]. This provides a framework to automatically infer probability estimates for those variables that have not been constrained yet. The computed probability estimates can then be used to generate a list of candidate solutions that are most likely to solve the lead's problem. Further, metrics can be defined to determine the uncertainty of variables and their influence on other variables (herein termed "conclusiveness"). By combining both metrics we're able to identify variables that should be constrained in order to reduce overall uncertainty about the design space. A salesperson can then be advised to specifically assess more information about those variables. See Fig. 2 for an example.

In Fig. 2, the upper part shows the encoding of conceptual model $\alpha$ as conditional probability distribution $P(W_1 \mid W_2, W_3)$. The lower part shows how it is used in combination with prior probabilities to generate estimates for yet unconstrained variables. After the first sales call high office productivity is known to be mandatory, which is reflected by the prior probability $P(W_3)$. Multiplying $P(W_3)$ with $P(W_1 \mid W_2, W_3)$ produces estimates for $P(W_2)$ and $P(W_1)$. $P(W_2)$ is now the most uncertain variable since all of its states are equally probable. Given that the conclusiveness for all three variables is equal, because the only considered influence is $P(W_1 \mid W_2, W_3)$, the salesperson would be asked to assess $W_2$. After the second sales call the answer for $W_2$ is represented as new prior probability $P(W_2)$. Multiplying $P(W_2)$ and $P(W_3)$ with $P(W_1 \mid W_2, W_3)$ now suggests that $W_1$ should be very likely in state high.

However, the structure of the design space needs to be configured for each lead qualification situation individually. Therefore we use first-order probabilistic models, which provide formalisms for composing probabilistic belief networks [11]. First-order probabilistic models combine probabilistic graphical models (Bayesian networks, Markov random fields, or more generally factor graphs) with a relational language (e.g. first-order logic) to represent probabilistic dependencies among attributes of multiple entities. In this case we're interested in the



α as conditional probability

| high office prod. $W_3$ | user mobility $W_2$ | workspace flexibility $W_1$ | $P(W_1 \mid W_2, W_3)$ |
|---|---|---|---|
| mandatory | high | high | 1.0 |
| mandatory | high | low | 0.0 |
| mandatory | low | high | 0.5 |
| mandatory | low | low | 0.5 |
| neglectable | high | high | 0.5 |
| neglectable | high | low | 0.5 |
| neglectable | low | high | 0.5 |
| neglectable | low | low | 0.5 |

……

Sales call 1

$N_1$ as prior probability (high office productivity)

| $W_3$ | $P(W_3)$ |
|---|---|
| mandatory | 1.0 |
| neglectable | 0.0 |

Estimate for $W_2$ (user mobility)

| $W_2$ | $P(W_2)$ |
|---|---|
| high | 0.5 |
| low | 0.5 |

Estimate for $W_1$ (workspace flexibility)

| $W_1$ | $P(W_1)$ |
|---|---|
| high | 0.75 |
| low | 0.25 |

Sales call 2

$N_1$ as prior probability (high office productivity)

| $W_3$ | $P(W_3)$ |
|---|---|
| mandatory | 1.0 |
| neglectable | 0.0 |

$U_1$ as prior probability (user mobility)

| $W_2$ | $P(W_2)$ |
|---|---|
| high | 1.0 |
| low | 0.0 |

Estimate for $W_1$ (workspace flexibility)

| $W_1$ | $P(W_1)$ |
|---|---|
| high | 1.0 |
| low | 0.0 |

**Fig. 2** Probabilistic representation of reasoning steps carried out in the world part of exemplary design space depicted in Fig. 1

dependencies among design components and the world. In the following we describe how the design space model can be implemented as first-order probabilistic model.

## Implementation

We represent **W** and $\mathbf{C}_1 \cup \mathbf{C}_2 \cup \ldots$ as random variables, where each possible value assignment of a variable is mapped to a probability. This probability expresses the belief of the vendor that the variable is in a certain state. The notion of parameterized random variables (par-variables) allows us to define random variables as a function of one or more logical variables, which are called the par-variable's parameters (cf. [26] ). In this sense a par-variable A(X) ∈ **A** represents a set of normal random variables, one for each possible parameter assignment $A(x_1)$, $A(x_2)$, …, $A(x_n)$. In our model we use the parameter X to assign a variable to the world or a specific part of the design object.

To represent the dependencies of α, β, and γ we use parametric factors (par-factors) that define probability distributions over sets of random variables. A par-factor is a triple ⟨**B**, φ, ω⟩ where **B** ⊂ **A** is a set of par-variables, ϕ is a probability distribution that maps from the Cartesian product of ranges of variables in **B** to a positive real value (a probability), and ω is a set of logical terms that constrain the possible value assignments to the parameters of **B**.

The used par-variables and par-factors form a parameterized belief network. It can be transformed into a normal belief network (Bayesian network) by considering every possible combination of value assignments to the used parameters (grounding). In the following we show how parameterized belief networks facilitate different inference tasks for design support (see also [10] for a discussion of the mentioned design processes).

## Formulation

In the formulation process a salesperson acquires information about a lead's problem situation through discussions with the lead's representatives and possibly identifies implicit problems. Since time and possibilities for information exchange are limited the salesperson should ask for information that is yet uncertain and which allows to draw as many conclusions on the solution space as possible.

To provide a support service that highlights such important questions we first compose a design space: By defining par-factors for each type of dependency relation in α, β, and γ we can construct a design space with respect to some given component structure R:

$$\langle\{A_1(X_1), A_2(X_2), \ldots, A_n(X_n)\}, \phi, \{(X_1, X_i) \in R\}\rangle \tag{1}$$

This type of par-factor represents a dependency relation between its first par-variable $A_1(X_1)$ and some further variables $A_2(X_2)$, …, $A_n(X_n)$, where the component (or world) $X_i$ of each variable has to stand in structural relation to the component (or world) $X_1$ of the first variable. The actual dependency is expressed as a conditional probability distribution φ that maps the possible combinations of value assignments of the par-variables to a probability. The conditionally dependent variable in φ is the first par-variable $A_1(X_1)$.

In addition to par-factors describing dependencies among variables we introduce par-factors representing the constraints of N, U, $F_W$ and $F_D$. These provide prior probability distributions π for every par-variable in the model:

$$\langle \{A(X)\}, \pi, \emptyset \rangle \tag{2}$$

Initially π is uniformly distributed. But changing π by assigning a relatively high probability to some value of the variable would express its preference over other values. Information gathered in a sales call may now be expressed in terms of high prior probabilities for variable values that represent assessed information. Implicit information about the design space can then be determined by applying a belief propagation (BP) algorithm [28]. BP configures the probabilities of each variable in the grounded network with respect to the defined prior probabilities and dependency relations.

Having every variable configured with BP we can apply a scoring function on each variable to determine its importance for being assessed in future dialogues with the lead. In [9] we presented a measure that can be used to calculate a variable's uncertainty and conclusiveness. We define U($\phi$) as measure of uncertainty for a discrete probability distribution $\phi$ of size n:

$$U(\phi) \overset{\text{def}}{=} \left(\sum\nolimits_{\phi} \phi\left(\log\phi - \log\frac{1}{n}\right)\right) \Big/ \log\frac{1}{n}$$

U($\phi$) → [−1, 0] measures the Kullback-Leibler divergence [27] of $\phi$ with respect to a uniform distribution of same size n. Its maximum 0 is reached if $\phi$ is close to the uniform distribution, and thus all variable states being equally probable. To assess a variable's uncertainty we simply calculate U(ϕ) for its probability estimates, which were inferred through BP.

Assessing the conclusiveness of some variable A(X) is based on its influence on other variables. A dependency relation par-factor (Eq. 1) has a high influence on its variables if its conditional probability distribution scores low on the uncertainty measure, i.e. U($\phi$) is close to −1. Now given all dependency par-factors involving A(X), the sum of their uncertainty scores should be minimal to have a high conclusiveness.

Combining both a variable's uncertainty and conclusiveness we're able to highlight important variables in need of further investigation by the salesperson.

## Synthesis

Having formulated the lead's problem situation to a certain extent the salesperson gives consideration to candidate offers that seem capable of providing the expected "function-as-effect". Aiding the selection of a candidate offers that are likely to serve as a solution is intrinsically supported by the design space model: BP can be used to propagate gathered evidence on the problem situation towards variables describing offerable products and services. The joint probability estimate over all offer variables can then be seen as a configurational space of alternative offers, where offers that are likely to meet the given constraints are given a relatively high probability. The most likely offer is the maximum a posteriori (MAP) estimate.

## Analysis and Evaluation

After choosing a candidate offer, e.g. the MAP-estimate, the salesperson should evaluate to what extent the offer fits to the given problem situation. To measure this fit we first need to decouple the parts of the design space describing the offer from the parts describing the problem situation so that we can perform inference on both parts separately. Therefore we simply duplicate the design space representation. The first copy represents the problem part, which is actually identical to the belief network used for formulation and synthesis. The second copy represents the offer. Here we remove all prior probabilities (Eq. 2) standing for constraints of N, U, $F_W$ and $F_D$. Instead we change the prior probabilities of offer variables. By choosing prior probabilities of exactly 1 or 0 for offer variables we can express the selection of a certain candidate offer (clamping). Now we can apply BP on both copies separately and compare the resulting probability distributions for every pair of corresponding variables. Especially we're interested in differences of variables that can be assigned to N, U, $F_W$ and $F_D$ constraints.

We use a conditional probability distribution to measure the fit of such a variable pair. Two copies A(X) and A′(X) of the same variable, which has n possible values, are compared by conditioning a third random variable M with range {match, mismatch}:

$$P(M|A(X), A'(X)) \overset{\text{def}}{=} \begin{cases} 1/2n & \text{if } A(X) = A'(X) \text{ and } M = match; \\ \frac{1}{n-1}/2n & \text{if } A(X) \neq A'(X) \text{and } M = mismatch; \\ 0 & \text{Otherwise} \end{cases}$$

Multiplying this with the probability estimates for A(X) and A′(X) gives the joint probability for "expected variable" A(X) and "offered variable" A′(X) having "matching" or "mismatching" values. We can determine such probabilities for each pair of variables separately and then compute a mean value to assess the overall fit of an offer with respect to a problem situation. The result can be used to guide the process of reformulation.

## Case Study

The computational representation of the design space and the described inference techniques have been integrated in a prototype application. It realizes an information system for supporting the lead qualification process. This section presents preliminary results of an ongoing case study, in which we test this prototype with designers, consulting experts and salespersons from a major furniture manufacturer that is concerned with office fit-out projects.

## Prototype Application

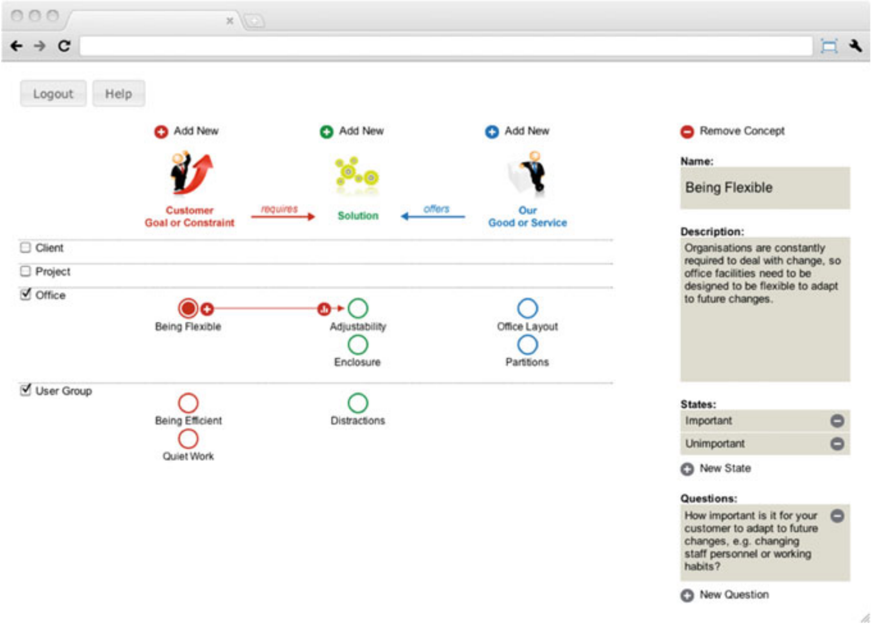
To define par-variables and par-factors that can be used for instantiating a design space the prototype application provides a web-based interface for knowledge engineering shown in Fig. 3.

Variables are defined by providing (1) a name and description of the underlying concept, (2) a rating of the concepts general importance with respect to the lead qualification process, (3) a set of questions that will guide a salesperson in assessing information about the variable, and (4) the variable's range that also frames the possible answers to the questions. Each variable is assigned to the world or a distinct type of design component. In this case the world is termed "Client" and we have three types of components: "Project", "Office" and "User Group". Further, the knowledge engineering interface discriminates between variables that are closely related to the vendor's offer ("Our Good or Service") and variables that are affected by abstract constraints ("Customer Goal or Constraint"). All other variables reside under the category "Solution".

Having described a set of variables in this way a knowledge engineer may define dependency relations among those variables. By specifying which variables are affected by the dependency relation the knowledge engineering interface generates a tabular representation of all possible value combinations as shown in Fig. 4. Each combination is phrased as a conditional statement and assigned with a rating that represents its probability. Initially set to "I don't know" (0.5) these ratings can be adjusted by the knowledge engineer. This is done by choosing a statement as being "always true" (1), "often true" (0.75), "seldom true" (0.25) or "never true" (0). Leaving the rating at 0.5 represents a neutral state, i.e. either the statement may be as often true as it is false or the knowledge engineer is unable to determine. However, these ratings cannot be directly used as probabilities of a par-factor's probability distribution. First they need to be normalized so that all probabilities sum to 1, where in the case of all ratings being 0 the probabilities are reset to be uniformly distributed.

In total the furniture manufacturer's experts defined a set of 33 par-variables and 44 par-factors. These provide the basis for the second part of the prototype application, which resembles a dynamic sales questionnaire. This sales questionnaire (cf. Fig. 5) is also web-based and can be used by a salesperson in preparation and follow-up to a sales call or even during sales calls via a tablet computer.

The central part of the questionnaire is a list of questions, which is generated from those questions defined by the knowledge engineers. Its sortation is dynamically determined by using the scoring function for assisting the formulation process. To setup the necessary design space a salesperson simply defines the projects associated to a client, the offices that are subject to these projects, and the



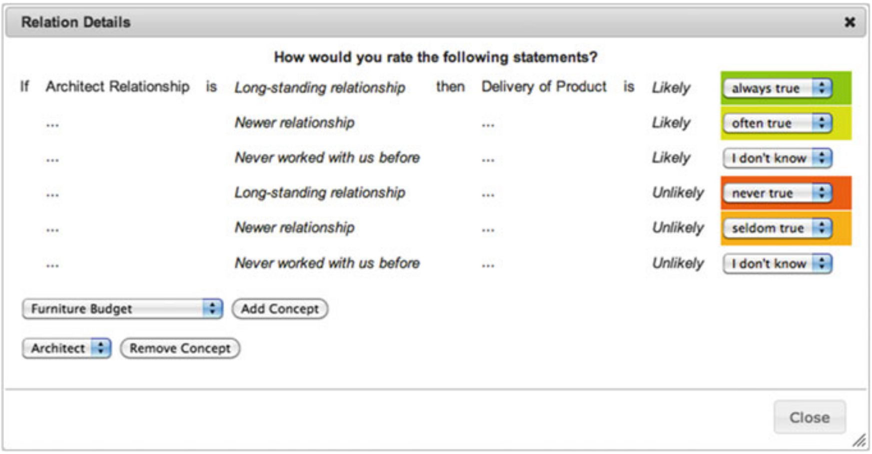
Name: Being Flexible

Description: Organisations are constantly required to deal with change, so office facilities need to be designed to be flexible to adapt to future changes.

States:
- Important
- Unimportant

Questions: How important is it for your customer to adapt to future changes, e.g. changing staff personnel or working habits?

**Fig. 3** Screenshot of knowledge engineering interface

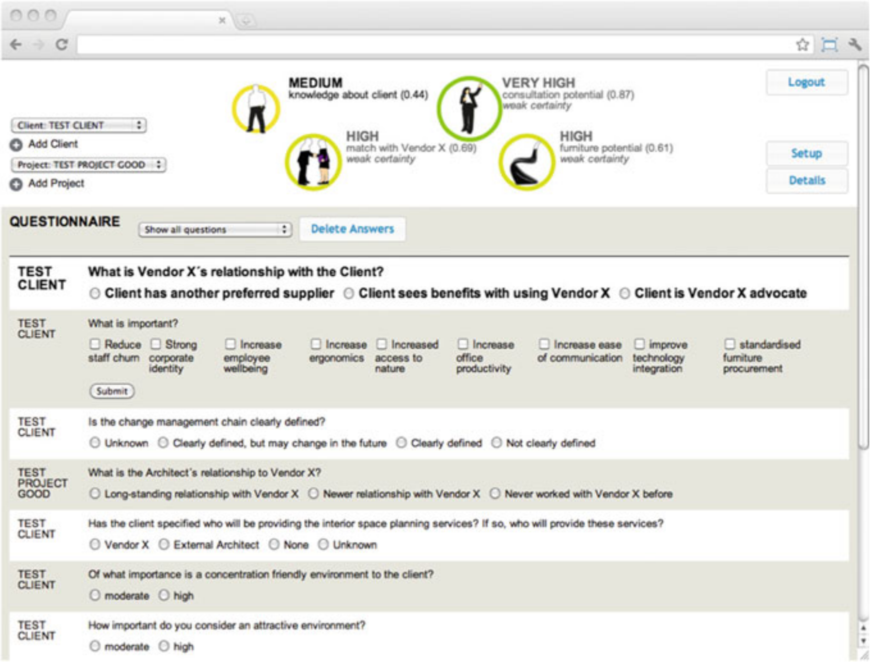


How would you rate the following statements?

|  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|
| If | Architect Relationship | is | *Long-standing relationship* | then | Delivery of Product | is | *Likely* | always true |
|  | ... |  | *Newer relationship* |  | ... |  | *Likely* | often true |
|  | ... |  | *Never worked with us before* |  | ... |  | *Likely* | I don't know |
|  | ... |  | *Long-standing relationship* |  | ... |  | *Unlikely* | never true |
|  | ... |  | *Newer relationship* |  | ... |  | *Unlikely* | seldom true |
|  | ... |  | *Never worked with us before* |  | ... |  | *Unlikely* | I don't know |

**Fig. 4** Screenshot of dependency relation definition

user groups, which are planned to work in the offices. When giving answers to the questionnaire the prior probabilities of associated variables are adjusted and the list is automatically resorted. Questions of variables with a widespread influence that haven't been answered explicitly or implicitly, and which have been defined



**MEDIUM**
knowledge about client (0.44)

**VERY HIGH**
consultation potential (0.87)
*weak certainty*

**HIGH**
match with Vendor X (0.69)
*weak certainty*

**HIGH**
furniture potential (0.61)
*weak certainty*

**QUESTIONNAIRE**

| | |
|---|---|
| **TEST CLIENT** | **What is Vendor X´s relationship with the Client?**<br>○ **Client has another preferred supplier** ○ **Client sees benefits with using Vendor X** ○ **Client is Vendor X advocate** |
| TEST CLIENT | What is important?<br>☐ Reduce staff churn ☐ Strong corporate identity ☐ Increase employee wellbeing ☐ Increase ergonomics ☐ Increased access to nature ☐ Increase office productivity ☐ Increase ease of communication ☐ improve technology integration ☐ standardised furniture procurement |
| TEST CLIENT | Is the change management chain clearly defined?<br>○ Unknown ○ Clearly defined, but may change in the future ○ Clearly defined ○ Not clearly defined |
| TEST PROJECT GOOD | What is the Architect´s relationship to Vendor X?<br>○ Long-standing relationship with Vendor X ○ Newer relationship with Vendor X ○ Never worked with Vendor X before |
| TEST CLIENT | Has the client specified who will be providing the interior space planning services? If so, who will provide these services?<br>○ Vendor X ○ External Architect ○ None ○ Unknown |
| TEST CLIENT | Of what importance is a concentration friendly environment to the client?<br>○ moderate ○ high |
| TEST CLIENT | How important do you consider an attractive environment?<br>○ moderate ○ high |

**Fig. 5** Screenshot of sales questionnaire

as being important by the knowledge engineers are placed on the top. This sortation is intended to guide the salesperson's priority in assessing information from the lead.

Further, the techniques described for synthesis, analysis and evaluation are used to generate ratios that are of special interest to a salesperson during lead qualification. First, to give the salesperson an orientation about the progress of lead qualification the mean of all computed uncertainty scores is used to indicate the current knowledge about the lead. Second, probability estimates of certain variables are presented that help to determine opportunities for business, i.e. the potential for providing consultation services or selling furniture to the lead. Third, the mean of all "match"-probabilities is used to show a simplified overall score for the fit of the most probable offer (MAP-estimate) with respect to the problem situation.

## Preliminary Results

Main parts of the knowledge engineering have been conducted in a three-day workshop with an interior-design architect, a representative from the sales training department, and a representative from the project-consulting department. After an introduction to the usage of the knowledge engineering interface, participants were asked to use the system to model the concepts being most relevant to lead qualification situations. Thereby participants took the role of knowledge expert and knowledge engineer at the same time. The parts that should be described were predefined as “Client”, “Project”, “Office” and “User Group”. Since there was no predetermined process that would have guided this task, the participants had to find a strategy on how to develop the knowledge base from scratch and how to do this cooperatively. The participants decided to proceed inductively by using an example project of the interior-design architect. Following several discussions about the details, the participants generalized important issues of this case under the architect’s lead to defined variables of the knowledge model. After a period of using the system jointly to enter one variable at a time and discussing its semantics, participants decided to work in parallel. Therefore every predefined component “Client”, “Project”, “Office” and “User Group” was assigned to a certain participant whose responsibility was to model the variables and dependency relations within that component. Afterwards the participants gathered again to define dependency relations between the components jointly.

During this process several obstacles occurred, which are now subject for our ongoing refinement of the prototype:

First, in some cases the participants found it hard to assign a variable to the given categories. Some variables could be assigned to one or another category depending on the point of view. E.g. storage capabilities may be defined for each user group individually or for an office as a whole. In this case the knowledge engineers decided to model storage capabilities on the lowest detail level (user group) to allow a greater flexibility at the cost of a salesperson having to answer more questions.

Second, dependency relations used in the current state of the prototype are limited to multinomial probability tables. However, when using multinomial distributions in a Bayesian network for encoding dependency relations, parents of an influenced variable affect each other (explaining away phenomenon [28]). In some cases this is not desirable. E.g. probability estimates for furniture potential and consultation potential are derived from different variables of the whole design space. Now when some variable (e.g. providing storage) has a positive influence on furniture potential this should not automatically affect another semantically unrelated parent (e.g. relationship to architect). In this case each parent should contribute independently to furniture potential. Thus knowledge engineers should be able to choose from different types of dependency relations, like causally independent noisy-OR, noisy-MAX or noisy-addition distributions [29].

Third, in close relation to issue two, knowledge engineers experienced unwanted side effects in the reaction of the dynamic questionnaire while modeling the knowledge base. To make these effects comprehendible it was suggested to implement a mechanism that explains the inference results produced by the questionnaire. It should offer the possibility to easily retrace the causes for a specific suggestion of the questionnaire.

## Conclusion and Future Work

Lead qualification in solution selling industries marks the very beginning of a design process that is initiated by discussions of a salesperson with representatives of a lead. Depending on the results of lead qualification this design process may or may not be carried on by design experts in a successive project. Thus, supporting the process in its initial phase is an important issue that underlines the need for representing design spaces in lead qualification situations.

Though functional representations were originally developed to model devices, products, objects, and processes based on their functionalities [5], this approach can also be applied to model problem-solution discovery in lead qualification situations. As demonstrated these models can help to bridge the gap between abstract business needs of a lead and the potential product and service bundles offered by a vendor and its partner network.

Throughout lead qualification salespersons try to iteratively structure a design space that is only vaguely defined in the beginning. To support this process a design space model should explicitly account for the representation of uncertainties. We showed that first-order probabilistic models are suitable for its implementation. The preliminary results from a case study suggest that the proposed representation can be practically used to formalize design knowledge in form of a knowledge base. This in turn can be used by inference mechanisms during lead qualification to simulate and assist formulation, synthesis and evaluation tasks.

However, the generation the proposed first-order probabilistic representation requires a substantial knowledge engineering effort. In our future work we'll refine the prototype application to simplify the knowledge acquisition process. Therefore we seek to realize an explanation component that helps knowledge engineers and salespersons to understand the inference results. Further, we'll implement machine-learning capabilities that make use of the answers given to the questionnaire in order to inductively learn new dependency relations. And we seek to generalize the definition of dependency relations to allow the specification of individual types of conditional probability distributions.

**Acknowledgment** This work was partially funded by the German Federal Ministry for Education and Research (BMBF, contract 17N0409). The authors would like to thank Sabine Janzen, Simone Orth, Andreas Filler and Tobias Kowatsch for valuable discussions and an anonymous reviewer for helpful comments on earlier versions of this paper.

## References

1. Tuli KR, Kohli AK, Bharadwaj SG (2007) Rethinking customer solutions: from product bundles to relational processes. J Market 71(3):1–17
2. Schön DA (1992) Designing as reflective conversation with the materials of a design situation. Res Eng Des 3(3):131–147

3. Regli WC, Hu X, Atwood M, Sun W (2000) A survey of design rationale systems: approaches, representation, capture and retrieval. Eng Comput 16(3):209–235
4. Chandrasekaran B, Goel A, Iwasaki Y (1993) Functional representation as design rationale. Computer 26(1):48–56
5. Erden MS, Komotoa H, van Beeka TJ, D'Amelioa V, Echavarriaa E, Tomiyama T (2008) A review of function modeling: approaches and applications. AI EDAM 22(2):147–169
6. Wang L, Shen W, Xie H, Neelamkavil J, Pardasani A (2002) Collaborative conceptual design—state of the art and future trends. Comput Aided Des 34(13):981–996
7. El-Sawalhi N, Eaton D, Rustom R (2007) Contractor pre-qualification model: state-of-the-art. Int J Proj Manage 25(5):465–474
8. Mohemad R, Hamdan AR, Ali Othman Z, Mohamad Noor NM (2010) Decision Support Systems (DSS) in construction tendering processes. IJCSI 7(2):35–45
9. Eichhoff JR, Maass W (2011) Representation and reuse of design knowledge: an application for sales call support. In: König A, Dengel A, Hinkelmann K, Kise K, Howlett RJ, Jain LC (eds) KES 2011, Part I. LNAI 6881:387–396
10. Gero JS, Kannengiesser U (2004) The situated Function-Behaviour-Structure framework. Des Stud 25(4):373–391
11. De Salvo Braz R, Amir E, Roth D (2008) A survey of first-order probabilistic models. In: Holmes DE, Jain LC (eds) Innovations in Bayesian networks. Stud Comput Intell 156:289–317
12. Christophe F, Bernard A, Coatanéa É (2010) RFBS: a model for knowledge representation of conceptual design. CIRP Ann Manuf Techn 59(1):155–158
13. Goel AK, Rugaber S, Vattam S (2009) Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. AI EDAM 23(1):23–35
14. Gorti SR, Gupta A, Kim GJ, Sriram RD, Wong A (1998) An object-oriented representation for product and design processes. Comput Aided Des 30(7):489–501
15. Labrousse M, Bernard A (2008) FBS-PPRE, an enterprise knowledge lifecycle model. In: Bernard A, Tichkiewitch S (eds) Methods and tools for effective knowledge life-cycle-management. Springer, Heidelberg, pp 285–305
16. Szykman S, Sriram RD, Bochenek C, Racz JW, Senfaute J (2000) Design repositories: next-generation engineering design databases. IEEE Intell Syst App 15(3):48–55
17. Xue D, Xu Y (2003) Web-based distributed system and database modeling for concurrent design. Comput Aided Des 35(5):433–452
18. Kroll E, Condoor SS, Jansson DG (2001) innovative conceptual design: theory and application of parameter analysis. Cambridge University Press, Cambridge
19. Zdrahal Z (2007) The role of knowledge in design problems. In: Wagner R, Revell N, Pernul G (eds) DEXA 2007. LNCS, vol 4653. Springer, Heidelberg, pp 884–894
20. Bonney FL, Williams BC (2009) From products to solutions: the role of salesperson opportunity recognition. Eur J Market 43(7/8):1032–1052
21. Chandrasekaran B, Josephson JR (2000) Function in device representation. Eng Comput 16(3):162–177
22. Chandrasekaran B (2005) Representing function: relating functional representation and functional modeling research streams. AI EDAM 19(2):65–74
23. Baccarini D, Bateup G (2008) Benefits management in office fit-out projects. Facilities 26(7/8):310–320
24. Peponis J, Bafna S, Bajaj R, Bromberg J, Congdon C, Rashid M, Warmels S, Zhang Y, Zimring C (2007) Designing space to support knowledge work. Environ Behav 39(6):815–840
25. Nikolaidis E, Mourelatos ZP, Pandey V (2011) Design decisions under uncertainty with limited information. CRC Press/Balkema, Leiden
26. Kisyński J, Poole D (2009) Constraint processing in lifted probabilistic inference. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence (UAI '09), pp 293–302

27. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86
28. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Francisco
29. Heckerman D, Breese JS (1994) Causal independence for probability assessment and inference using Bayesian networks. IEEE T Syst Man Cyb 26(6):826–831

# Using Part Functions to Capture Various Lifecycle Requirements in Detailed Design

**Yong Chen, Jian Huang and Youbai Xie**

**Abstract** Although various lifecycle requirements are critical for detailed design, they often merely remain as tacit knowledge in designers' minds. This paper attempts to develop a formal approach for capturing them. Since detailed design primarily deals with part design, how to formally describe a part is introduced at first. A concept, *part function*, is then proposed for describing various lifecycle factors that should be considered for the detailed design of a part. A lifecycle requirements-capturing approach is then developed, where part functions are associated with the geometrical features or neighbor spaces used for achieving them so that various lifecycle constraints can be derived from those part function descriptions. A fixture design case illustrates the proposed lifecycle requirements-capturing approach.

## Introduction

Detailed design is usually regarded as a process that transforms a conceptual solution concept into a complete geometrical description about a product. Although the result of detailed design primarily deals with geometrical information (e.g. forms and dimensions), it is actually related to not only the upstream lifecycle requirements generated in conceptual design and embodiment design, but also the downstream lifecycle (e.g. manufacturing, assembling, transportation, maintenance, recycling, etc.) requirements predicted for the subsequent stages [1]. Therefore, detailed design is actually a very complex decision-making process, and requires various lifecycle requirements to be carefully considered.

---

Y. Chen (✉) · J. Huang · Y. Xie
Shanghai Jiao Tong University, Shanghai, China
e-mail: mechenyong@sjtu.edu.cn

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_30,

However, detailed design in industry is often simply misunderstood as a geometrical modeling process. Existing CAD software (e.g. Pro/Engineer, UG NX, etc.) and PLM platforms (e.g. Teamcenter, Windchill, etc.) have primarily been developed for managing geometrical models and related constraints. Various lifecycle requirements that have led to detailed design results are often regarded as tacit knowledge and merely exist in designers' minds. Failure to manage such lifecycle requirements effectively can cause multiple serious issues in a product design organization. For example, it is often difficult for a chief designer or a senior designer to understand a detailed design submitted to him for audit and approval, and to judge whether various lifecycle requirements have been correctly considered. As a result, design errors often have to be detected later in the implementation stage. A reported design failure was that a designer once forgot to consider the assembling requirements on a product, making the product unable to be assembled with a suitable tool. Another serious issue is that it often makes designers difficult to reuse an existing detailed design, since they don't exactly know what lifecycle requirements have been considered in the design, which has been reported in multiple literatures, e.g. in Ref. [2]. This issue is especially important in matured industry domains, where most lifecycle requirements often remain invariant.

Therefore, it is of great importance to develop an effective approach for assisting designers to capture and manage various lifecycle requirements in detailed design, which is just the primary concern of this research. Since detailed design primarily deals with designing parts and should take into consideration their functions in various lifecycle periods, this paper develops a part function-based approach for capturing various lifecycle requirements in detailed design. Here, it is assumed that a detailed design has been laid out, and a designer should retrospect the design process for capturing those lifecycle requirements that have been considered.

This paper is organized as follows. Section "Part" briefly introduces the concept of part and its representation. Section "Part Function" proposes the concept of part function and its representation. Section "Capturing Lifecycle Requirements from Part Functions" develops a formal approach for capturing lifecycle requirements from part functions. With a mechanical fixture design as a case, Sect. "An Illustrative Case" illustrates how the proposed lifecycle requirements-capturing approach works. Following the related work discussed in Sect. "Related Work", Sect. "Conclusions" concludes this paper.

## Part

The concept of part here comes from the prevalent mechanical CAD software (e.g. Pro/Engineer), which refers to an individual entity that cannot be decomposed into smaller product entities. Therefore, the concept of part here is different from the concept of component, which may be further decomposed into smaller

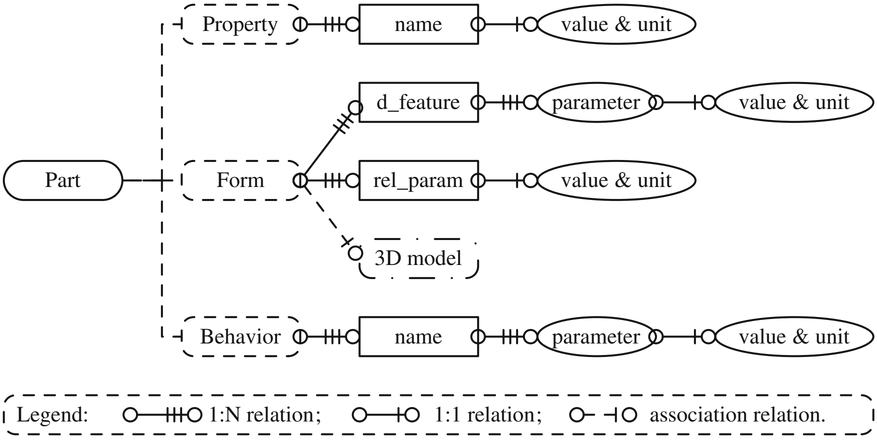


**Fig. 1** The physical information model of a part

sub-components [3]. To support modeling various lifecycle requirements on a part, a simplified information model is developed for describing the basic physical information of a part, as shown in Fig. 1. In this model, the physical information about a part falls into three major categories, i.e. property, form and behavior.

## *The Property Information*

The property information primarily denotes what physical characteristics (e.g. density, rigidity, stiffness, strength, etc.) a part has. Although such physical characteristics usually depend on what kind of material a part is made of, they are directly declared here for convenience.

To facilitate explanation, a property of a part is described in an objected-oriented syntax, i.e. *PART->property*, where the symbol "->" is an object-oriented member operator. For example, the rigidity of an axle can be described as: *AXLE->rigidity*.

## *The Form Information*

The form information describes the detailed geometry of a part. Although the 3D model of a part built with a commercial CAD system can explicitly describe its geometry, it is usually very complex from the designing point of view. Furthermore, it is also not convenient for a third party system to manipulate its geometrical information. Therefore, a descriptive feature-based approach is developed for representing the form of a part.

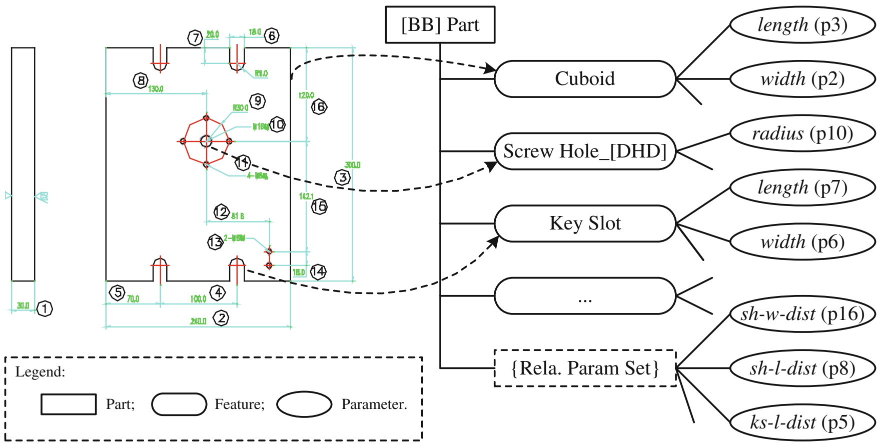


**Fig. 2** The descriptive feature-based representation of a part form

The descriptive feature-based approach employs a set of descriptive geometrical features and some relation parameters to describe the form information of a part. The geometrical feature set describes what geometrical features a part is composed of. Note that such geometrical features here are user-defined, which can either be simple features (e.g. a cuboid or a cylinder), or complex ones (e.g. an extension feature generated with an extension operation). A geometrical feature is further described with some feature parameters. The relation parameters define the location relations between the geometrical features in a part. With a Base_Board ([BB]) part as an example, Fig. 2 shows some geometrical features, feature parameters and relation parameters. Note that our descriptive feature-based model should be associated with a CAD model so that each of its parameters can correspond to one in the CAD model. For example, the length parameter (denoted as "p3") of the cuboid feature in Fig. 2 corresponds to the 3rd parameter (③) in the CAD model.

Based on the object-oriented syntax, the geometrical feature of a part can be described as a form, *PART->Feat*, while the parameter of a part feature can be represented as: *PART->Feat->param*. Similarly, the relation parameter of a part can be represented as: *PART->rel_param*. For example, the relation parameter, *sh-w-dist*, in the above [BB] part can be described as: *[BB]->sh-w-dist*, while the parameter, *length,* of its cuboid feature can be described as: *[BB]->Cuboid->length*.

## *The Behavior Information*

Based on our previous research [4, 5] and the scientific ontology [6], a part behavior is defined as an own state change of a part in a specific situation. Part behaviors include not only the function-related behaviors (e.g. the translation of a slider in a slider-crank mechanism), but also many other behaviors in various lifecycle periods of a product (e.g. the torsional deform of an axle), since such behaviors are also related to various lifecycle requirements on a part.

The part behavior description here is composed of two sections. One is the name information (i.e. the name of a behavior), and the other is the parameter information (i.e. the parameters for describing the quantitative degree). Note that a part may have multiple behaviors, and a behavior may also have multiple parameters. For example, the piston in an engine have multiple behaviors, such as translation, deforming, wearing, etc., while its translation behavior also has multiple parameters, e.g. *max-disp* (i.e. maximal displacement), *max-vel* (i.e. maximal velocity), etc.

With the aid of the object-oriented syntax, the behavior of a part here is described as: *PART->Behavior*, and the parameter of a behavior can be further described as: *Behavior->parameter*. Based on this description syntax, the maximal displacement of the translation of a piston can be described as: *PISTON->Translation->max-disp*.

## Part Function

It is widely acknowledged that function is a critical concept for engineering design. Multiple functional modeling approaches have been developed for supporting various conceptual design tasks in recent decades, as reviewed in [7]. However, the role of function in detailed design has been largely neglected, which is probably because detailed design is often misunderstood as a pure geometrical modeling process. Since detailed design primarily deals with designing the parts of a product, function in detailed design is largely related to a part, and therefore can be called part function. This section will discuss the part function concept and its description.

### *Concept*

Based on the subject-object relation, a *Part Function* is defined as an intended (purposeful) relation that a part has (i.e. subject) with another entity (i.e. object). Each part function (abbreviated as "PF" later) should belong to a part (i.e. its subject). Note that an intended relation here cannot only be one based on a physical action (e.g. force or moment), but also one based on some other kind of relation (e.g. layout relation). For example, man may either say that the top part of a computer table has a PF of supporting a computer (a force-based action), or that it has a PF of accommodating a human user (a layout-based relation). In this sense, part function is more like the concept of affordance, which refers to the complementarity relation between one system and another [8], but at a more detailed level (i.e. part).

According to the types of the objects, PFs can be classified as internal PFs and external PFs. For an internal PF, its object is another part of the product that the subject part belongs to. Therefore, an internal PF describes an internal relation between a product part and another product part. For an external PF, its object is an external entity that is not a part of the product that the subject part belongs to. Therefore, an external PF can describe an intended relation between a product part and an external object in the product's environment. For example, for a computer table, we can say that its top part has an external PF of supporting a computer, while its leg part has an internal PF of supporting the top part.

Internal PFs can be further classified as static internal PFs and dynamic internal PFs. A static internal PF means that its object is treated as a static part. For example, the leg part in a computer table has a static internal PF, to support the top part, since the top part is treated as a static entity. A dynamic internal PF means that its object is treated as a dynamic part (i.e. the part has a dynamic behavior that should be considered). For example, in a combustion engine, the cylinder has an important dynamic internal PF, *to allow the moving (translation) of the piston*, where what is of particular interest is the moving behavior of the piston.

It should be pointed out that the external objects that external PFs deal with involve not only those function- or operation-related objects, but also the external objects in various lifecycle periods (e.g. manufacturing, assembling, transportation, maintenance, etc.) of a product. This feature also differentiates our PF concept from the artifact affordance concept developed in [8], which primarily deals with the affordance of an artifact in its use (i.e. operation) period. Otherwise, even if a product could achieve its functions, it would be difficult or even impossible to manufacture, assemble, transport, and (or) maintain it. For example, when designing a computer table, a designer should not only consider the external objects related to its use (e.g. computer, human user, etc.), but also various tools for machining its parts and assembling them together. Therefore, external PFs can help designers understand what external lifecycle objects are considered in an existing design, which thus provides a possible foundation for capturing various lifecycle requirements in detailed design.

## *Part Function Versus System Function*

To have a more explicit understanding of PF, it is helpful to compare the PF concept with the traditional concept of function in engineering design.

According to Pahl and Beitz [1], function is defined as a general relationship between the input and output (i.e. flow) of a system, aiming at performing a task. Since this concept of function is related to a system, it is called system function here. The system function concept has widely been used to achieve functional reasoning (e.g. in [9–11]). It has three major features. First, just as a decomposable system, a system function can also be decomposed into sub-functions. Second, a system function must deal with the change of a flow from an input to an output.

Finally, a system function merely deals with the interaction between a system and a flow(s) through it, without the interactions between the components in the system. Note that the concept, *element function*, proposed by Goel et al. [3] is also similar to the concept of system function, since it is not only decomposable, but also deals with input-output transformation.

In contrast, the PF concept is largely different. First, just as part, a PF cannot be decomposed. Second, it doesn't require an explicit transformation of an object from input to output. For example, the aforementioned PF, *to accommodate a human user*, involves no transformation. Finally, it deals with not only the intended relation of a part with an external object, but also the intended relations between the internal parts (i.e. internal PFs). In addition, a PF deals with not only a relation of a part with a function-related external object (flow), but also one with an external object in various lifecycle periods of a product. For example, in order to make a part easy to be assembled, it probably should have an external PF of accommodating a wrench for assembling.

## *Descriptions of Part Functions*

To capture various lifecycle requirements from PFs, it is indispensable to have a formal approach for describing them. Through extending the "verb + flow" approach proposed by Szykman et al. [12], this research develops a "Subject + Verb + Object" (abbreviated as SVO) approach for describing PFs in detailed design. Since PFs falls into three categories (i.e. external PFs, static internal PFs and dynamic internal IPFs), the SVO approach also has three variants.

The first variant is for describing an external PF, which can be conceptualized as a triple, $\langle S, V, O_E \rangle$. Here, $S$ refers to the subject part, $V$ denotes the verb for describing the intended relation, and $O_E$ indicates the external object. For example, the function of the top part in a computer table, *to support a computer*, can be described as ⟨TOP, SUPPORT, COMPUTER⟩. Note that the subject part $S$ is associated with its information model shown in Sect. "Part", while the external object $O_E$ is further denoted with some parameters related to detailed design. For example, the above object, COMPUTER, should be further described with some parameters, such as *length*, *width*, *weight*, etc. The parameters of an external object are also described in an object-oriented form, i.e. *OBJ->param*.

The second variant is for describing a static internal PF, which can be conceptualized as a triple, $\langle S, V, O_I \rangle$. Here, $O_I$ indicates an internal part that is intentionally related with the subject part $S$. Of course, both the subject part $S$ and the object part $O_I$ here are associated with their formal part representations. Based on this model variant, the PF that the leg part fulfills in a computer table for supporting its top part can then be described as ⟨*LEG*, *SUPPORT*, *TOP*⟩, with the parts *LEG* and *TOP* have their own formal part representations.

The final variant is for describing a dynamic internal PF, which can be conceptualized as a triple, $\langle S, V, B_O \rangle$. Here, $B_O$ denotes the behavior of the internal object part related to the subject part *S*. Note that since each behavior belongs to a part, the behavior description here has already implicitly contained the object part of a dynamic internal PF, though it hasn't been clearly declared in the above triple. Based on this model, the aforementioned dynamic internal PF of the cylinder part in a combustion engine, *to allow the translation of the piston*, can then be described as ⟨*CYLINDER*, *ALLOW*, *PISTON->Translation*⟩, where "*PISTON->Translation*" denotes the translation behavior of the piston part. Note that this function cannot be simply described as ⟨*CYLINDER*, *ALLOW*, *Translation*⟩, since it can lead to the loss of the information of the object part *PISTON*.

Based on the above analysis, a general and unified PF description model can then be proposed, i.e. ⟨*S*, *V*, *X*, *type*⟩. Here, *S* and *V* have the same meanings as they are in the previous models; *X* can be an external object, an internal part or a behavior of an internal part, depending on the *type* variable, which indicates the type of a PF, with "E" for External PF, "S" for Static internal PF, and "D" for Dynamic internal PF. For example, the PFs mentioned above can then be described as, ⟨*TOP*, *SUPPORT*, *COMPUTER*, E⟩, ⟨*LEG*, *SUPPORT*, *TOP*, S⟩and ⟨*CYLINDER*, *ALLOW*, *PISTON->Translation*, D⟩, respectively.

## Capturing Lifecycle Requirements from Part Functions

Since PFs have contained various lifecycle objects that should be considered in a detailed design, it is possible to capture various lifecycle requirements from them. Here, lifecycle requirements will be formalized as quantitative constraints that can be used for determining the values of the geometrical parameters of a part. The lifecycle requirements-capturing process is primarily composed of four major steps, i.e. associating geometrical features with PFs, associating neighbor spaces with PFs, merging PFs into Extreme Functional Combinations (EFCs), and finally deriving parameter-based lifecycle constraints for detailed design.

### Associating Geometrical Features with Part Functions

As 3D CAD systems are widely used, designers are getting used to employing geometrical features to achieve various lifecycle PFs. For examples, assuming a designer is designing a top part, which has a PF of supporting a computer, it is very possible that he will employ a cuboid feature to achieve this PF; if the top part is also required to have a PF of containing the electricity cord, he will probably employ a through round hole feature to achieve this PF.

Therefore, it is both possible and reasonable to associate a geometrical feature with a PF. For examples, two associations can be built for the above two examples, i.e., the association between the through round hole feature and the PF of containing an electricity cord, and the association between the cuboid feature and the

PF of supporting a computer. Associating a geometrical feature with a PF can make related designers (e.g. a chief engineer) able to understand how a lifecycle PF is achieved by a geometrical feature. Note that the geometrical feature associated with a PF is not the only feature that can achieve the PF. Instead, there often exist multiple geometrical features that can achieve a PF. For example, a designer may also use a cylinder feature (its top plane) to achieve the PF of supporting a computer. Which feature is selected for achieving a PF usually depends on various factors, which needn't be discussed here.

The association between a geometrical feature and a PF can be conceptualized as a binary group, ⟨*Feat_ID, Func_ID*⟩, where both *Feat_ID* and *Func_ID* are the identifications (IDs) of the corresponding geometrical feature and the PF of a part. For example, the two aforementioned associations can be denoted as $\langle ID_{TOP\text{-}>Round\_Hole}, ID_{Contain\_Elec\_Cord}\rangle$, and $\langle ID_{TOP\text{-}>Cuboid}, ID_{Support\_Computer}\rangle$, respectively.

## *Associating Neighbor Spaces with Part Functions*

In detailed design, designers may also use a neighbor space to achieve a PF. Here, a neighbor space refers to an implicit space besides a part. Different from geometrical features that are visible, neighbor spaces are often implicit and cannot appear in the geometrical representation of a part. For example, besides the PF of supporting a computer, the top part of a computer table also has another PF, *to accommodate a human user*. In fact, this PF is not directly achieved by the top part, but by an implicit space besides the top part, since there should be a space to accommodate a human user. Therefore, neighbor space is also a very important approach for achieving a PF. However, it is often impossible to represent a neighbor space in a part model, since they are void of any physical entities.

An indirect approach is used here for associating neighbor spaces with PFs. This indirect approach is based upon our observation that there is always an association relation between a neighbor space for achieving a PF and some geometrical parameter(s) of a part. For example, the neighbor space besides the top part mentioned above for accommodating a human user can be associated with the length parameter of the top part, which can be understood as that in order for the top part to have a sufficient neighbor space for accommodating a human user, its length should be able to hold a human user (i.e. his width parameter); otherwise, the table would be too small). Thus, a neighbor space can be indirectly indexed by some geometrical parameters of a part. Thereby, the association between a neighbor space and a PF can then be conceptualized as: ⟨*{Param_ID}, Func_ID*⟩, where {*Param_ID*} is a set of IDs of the geometrical parameters that can be used for denoting the neighbor space of interest, and *Func_ID* is the ID of the corresponding PF. For example, the aforementioned association can then be represented as: $\langle \{ID_{TOP\text{-}>Cuboid\text{-}>length}\}, ID_{Accommodate\_Human\_User}\rangle$.

Note that since the geometrical parameters of a part are classified as feature parameters and relation parameters, the geometrical parameters used for building associations with PFs are not limited to feature parameters, but can also be relation parameters. For example, since the [BB] part shown in Fig. 2 should be assembled to the operation platform of a machine with 4 bolts, the distance between the two key slots in one side, *ks_dist* (see p. XX in Fig. 2), should then have a PF to hold the head of an adjustable wrench for allowing the assembling operation. This association can be represented as $\langle \{ID_{[BB]\text{-}>ks\_dist}\}, ID_{Hold\_Wrench}\rangle$. Here, the geometrical parameter, *ks_dist,* which is used for denoting the corresponding neighbor space, is a relation parameter of the [BB] part.

## *Merging Part Functions into EFCs*

It is the fact that a geometrical feature can often fulfill multiple PFs in a product. For example, besides the PF of supporting a computer, the cuboid feature of the aforementioned top part also typically has two additional PFs: to support a notebook and to support some books. These two PFs come from human users' needs of writing something in a notebook or finding information from books when they are using a computer. Similarly, it is also possible that a neighbor space is used for fulfilling multiple PFs. For example, the aforementioned neighbor space denoted by the distance between two key slots should also partially hold two nuts, besides the function of holding an adjustable wrench. Therefore, a geometrical feature or a neighbor space can be associated with multiple PFs. Such kind of phenomenon is becoming prevalent as the affordance-based design theory [8] is adopted in design practice.

For the detailed design of a part, the PFs associated with the same geometrical feature or neighbor space should often be merged together as an EFC (Extreme Function Combination) since they can often work together to influence the geometrical feature or the neighbor space. For example, if an office table is designed to fulfill multiple functions, such as to support lap-top computer, to support notebook, to support a printer, etc., such functions then should be merged together as an EFC for the associated geometrical feature (e.g. a cuboid feature). Therefore, an EFC actually represents a typical and extreme environment where a part is supposed to work.

However, it should be pointed out that not all PFs associated with a geometrical feature or a neighbor space should be merged as an EFC. For example, although a computer table is primarily designed for supporting a computer, it is also possible to use it to support a person, since a person probably need to stand on it to repair a light bulb under the ceiling. In such a situation, it is almost impossible that the computer table is being used by a human user either to make notes or to read a book. Therefore, the function of supporting a person and that of supporting a notebook or supporting a book then needn't be put into the same EFC. Therefore, multiple EFCs then should be built for a geometrical feature or a neighbor space,

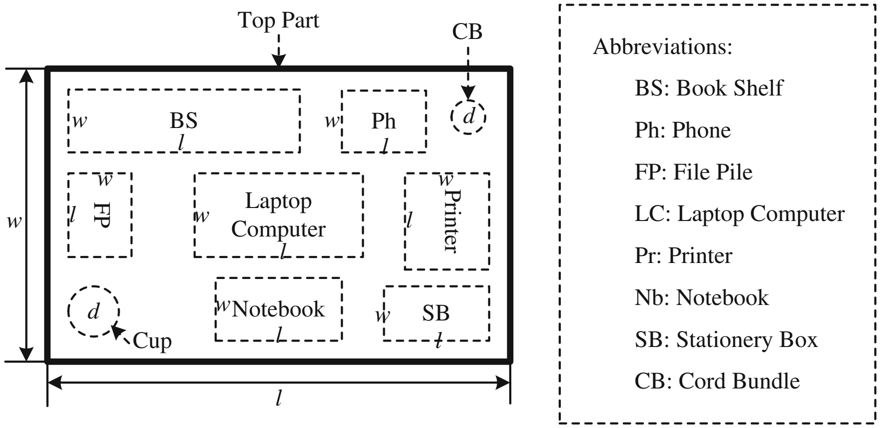


**Fig. 3** An EFC of an office table

i.e. a geometrical feature or a neighbor space can be associated with multiple different EFCs. In addition, it should be pointed out that PFs in different EFCs are not exclusive, i.e. a PF may appear in different EFCs.

The result of merging PFs into EFCs is a set of associations between a geometrical feature or neighbor space and an EFC. Such an association can then be described as $\langle GF_i, EFC_j \rangle$or $\langle NS_i, EFC_j \rangle$. Here, $EFC_j$ can be further described as a set of PFs connected together, i.e. $EFC_j = F_1 \& \ldots F_k \& \ldots F_p$. With the top part of an office table as an example, Fig. 3 shows an EFC scenario for using it, where its top part will ideally achieve multiple functions, e.g. to support a laptop computer, to support a notebook, to allow a cord bundle to go through, etc. For this situation, a typical association between a geometrical feature and an EFC can be built as, $\langle TOP\text{->}Cuboid,\ F_{FP} \& F_{BS} \& F_{LC} \& F_{Ph} \& F_{Pr} \& F_{Nb} \& F_{SB} \rangle$.

## Deriving Quantitative Lifecycle Constraints

A primary task of detailed design is to assign suitable values to the geometrical parameters of a part, considering various lifecycle requirements on it. However, the association between a geometrical feature or a neighbor space and an EFC built before can merely remind a designer of what PFs should be considered when determining the value of a part parameter. They fail to provide quantitative decision-making supports for detailed design. Therefore, a critical task here is to transform those associations into quantitative lifecycle constraints.

Our research has identified an axiom that can help designers to derive quantitative lifecycle constraints from the associations between geometrical feature or neighbor space and EFC. This axiom can be described as the Function-Constraining axiom shown as below.

> The Function-Constraining Axiom:
>
> If an entity S (Subject) is said to have a functional relation on an entity O (Object), then some design parameter(s) of S will be fully or partially constrained by some related design parameter(s) of O.

The function-constraining axiom exists and holds in each successful design case. For examples, if a seat is designed to hold an adult, then its parameter, *width*, will usually be constrained by (i.e. bigger than) the average width of adults; if a hole is designed for allowing a bunch of electrical cords to go through, i.e. its function is to accommodate electrical cord bunch, then the diameter of the hole should be constrained by (i.e. bigger than) the diameter of the cord bunch. Therefore, based on the function-constraining axiom, it is possible to derive some quantitative lifecycle constraints from the functional descriptions of a part.

To employ the function-constraining axiom to derive a quantitative lifecycle constraint from an association between a geometrical feature or a neighbor space and an EFC, a designer first should select a geometrical parameter of the subject part as the constrained parameter, since the related geometrical feature or neighbor space often involves multiple geometrical parameters. It should be pointed out that not all parameters of the geometrical feature in an association can be selected as the constrained parameters of the subject part. What parameters can be selected depends on specific EFC requirements in the association knowledge. For example, if the top part of an office table has an EFC that includes two functions, *to support a lap-top computer* and *to hold a notebook*, then its parameters *length* and *width* can be the primary constrained parameters, while its parameter *height* needn't be a constrained parameter; however, in another EFC that includes a function, to support a heavy object, then the parameter *height* (i.e. thickness) should be selected as a constrained parameter, while the other two needn't be considered.

After a constrained parameter has been selected, a designer then should select suitable parameters as the constraining parameters from the parameters of the objects involved in the PFs of an EFC. Since an EFC may deal with both external PFs and internal PFs, the objects here may either be external objects or the internal parts of a product. What parameters of the objects can be selected as the constraining parameters usually depends on specific physical knowledge. For example, if a restaurant table is designed for holding 4 persons (2 persons in two opposite sides), two external parameters, i.e. the average width of a person and the number of persons, can be selected as the constraining parameters when the parameter *length* of the table's top part is selected as the constrained parameter, while other parameters of the person can be neglected. In a quantitative lifecycle constraint, the constraining parameters related to an EFC will often form a polynomial to constrain the constrained parameter together.

The final step is to determine the quantitative relations between the constrained parameter and the constraining parameters. There are two kinds of quantitative relations that should be determined. One is the value relation (e.g. >=, <=, =, etc.) between the constrained parameter and the constraining polynomial. The other is the computation relation (e.g., +, −, etc.) among the constraining parameters in the polynomial. Both kinds of relations will depend on specific situations. What a designer should do here is to employ right domain knowledge to build reasonable relations for lifecycle constraints. As a result, for each geometrical parameter of a subject part, there will be a set of quantitative lifecycle requirements captured.

With the aforementioned association between geometrical feature and EFC, $\{\langle TOP\text{->}Cuboid,\ F_{FP}\&F_{BS}\&F_{LC}\&F_{Ph}\&F_{Pr}\&F_{Nb}\&F_{SB}\rangle\}$, as an example, how to derive a quantitative lifecycle constraint is illustrated here. First, a designer should select a suitable parameter as the constrained parameter. Here, assume that the length parameter of the cuboid feature is selected. Then, the designer should select some object parameters that are related to the constrained parameter as the constraining parameter. Here, the parameters, *FP->w*, *LC->l*, *PR-w*, *CU->d*, *NB->l*, and *SB->l*, are selected as constraining parameters, according to the EPF layout shown in Fig. 3. Finally, the quantitative relations (i.e. the lifecycle constraints) between the constrained parameter and those constraining parameters can then be built, leading to two lifecycle constraints derived. One is the formula, *TOP->Cuboid->l (>=) FP->w + LC->l + PR->w*, which means that in order for the top part to achieve the PFs of supporting FP (File Pile), LC (Laptop Computer) and PR (Printer), the length of its cuboid feature should be greater than the sum of the width of FP, the length of LC, and the width of PR. Here, the value relation symbol, >=, is surrounded by a parenthesis to avoid confusion with other similar symbols. The other is the formula, *TOP->Cuboid->l (>=) CU->d + NB->l + SB-*>w. Note that both formulae should be treated as the lifecycle constraints on the top part. Therefore, the value assigned to the length of the top part's cuboid feature should meet both lifecycle requirements.

## An Illustrative Case

The above lifecycle requirements-capturing approach has been implemented in a design knowledge modeling system, which, however, cannot be elaborated here, due to limited space. A mechanical fixture design is provided here to illustrate the lifecycle requirements-capturing approach.

This fixture is designed for clamping a Turbocharger Shell (TS) at a machining stage. Figure 4 show two primary drawings of this fixture. This fixture comprises 18 parts, such as the Base Board ([BB], □), the Shell Seat ([SS], □), two Double-Head Bolt ([DHB], ④), the Press Top ([PT], ⑤), the Adjusting Pillar ([AP], ⑥), the Pillar Shelf ([PS], ⑦), four Shell Seat Screws ([SSS], ⑨), etc. Here, the number in circle denotes the part number in Fig. 4b. Although it seemed not complex, a novice designer, who held a mechanical design bachelor degree and had been in this design team for about 6 months, said that he knew little about the related lifecycle requirements, and therefore was not clear about how to design it.

To capture lifecycle requirements, a designer first should build the part representation of the above fixture, i.e. primarily the form and behavior descriptions of each part. For example, the form of the [BB] part can be represented as a descriptive geometrical feature model shown in Fig. 2. Here, the part is composed

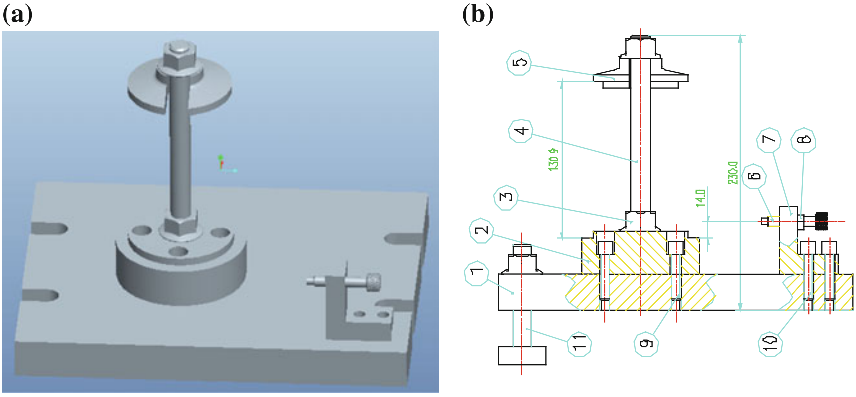
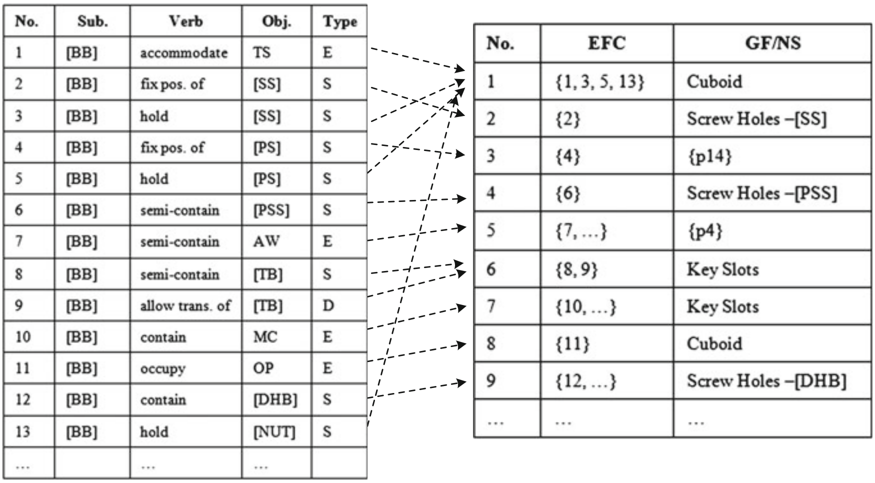


**Fig. 4** The drawings of a mechanical fixture. **a** The solid model. **b** An engineering drawing

multiple geometrical features (i.e. Cuboid, Screw Hole_[DHB], Screw Hole_[SS], Key Slot, etc.). The geometrical features of this part involve both the feature parameters, e.g. the length of cuboid (p13), the radius of a screw hole (p10), etc., and the relation parameters, e.g. *sh-w-dist* (p16), *kl-l-dist* (p5), etc. Since this mechanical fixture is more like a static device, the dynamic behaviors of its parts are not so detectable. However, they do have some dynamic behaviors in their lifecycle periods. For examples, a [TB] part has a translation behavior along the key slot of the [BB] part when assembled to the [BB] part; the [DHB] part has both a helical motion along the screw hole, and an extension deform when clamping a turbocharger; etc.

Then the designer should define the PFs of all parts of the fixture. About 60 PFs have been defined for the parts, which involve not only internal PFs, but also lifecycle external PFs. For examples, the [BB] part not only has such internal PFs as, to support the [SS] part, to fix the position of the [SS] part, to support the [PS] part, etc., but also multiple lifecycle external PFs such as to accommodate the Turbocharger Shell (TS), to hold Milling Cutter (MC), to hold Adjustable Wrench (AW), to occupy the Operation Platform (OP), etc., as shown in the left table of Fig. 5. Note that MC is an essential manufacturing tool for making the key slot, while AW is indispensable for assembling the fixture to a machine. Therefore, PFs enable designers to capture various objects (including both internal parts and the lifecycle external objects) that have been considered in a design.

After the PFs have been identified, designers should associate those PFs with the geometrical features of or the neighbor spaces related to the parts, and then merge suitable PFs into EFCs. With the detailed design of the [BB] part as an example, the right part of Fig. 5 shows the association between EFCs and the geometrical features or neighbor spaces. For examples, since the cuboid feature is for achieving an EFC, which includes such functions as to accommodate TS, to hold the [SS] part, to accommodate the [PS] part, etc., an association can then be built between the cuboid feature and the EFC (see association 1 in Fig. 5);



| No. | Sub. | Verb | Obj. | Type |
|---|---|---|---|---|
| 1 | [BB] | accommodate | TS | E |
| 2 | [BB] | fix pos. of | [SS] | S |
| 3 | [BB] | hold | [SS] | S |
| 4 | [BB] | fix pos. of | [PS] | S |
| 5 | [BB] | hold | [PS] | S |
| 6 | [BB] | semi-contain | [PSS] | S |
| 7 | [BB] | semi-contain | AW | E |
| 8 | [BB] | semi-contain | [TB] | S |
| 9 | [BB] | allow trans. of | [TB] | D |
| 10 | [BB] | contain | MC | E |
| 11 | [BB] | occupy | OP | E |
| 12 | [BB] | contain | [DHB] | S |
| 13 | [BB] | hold | [NUT] | S |
| … |  | … | … |  |

| No. | EFC | GF/NS |
|---|---|---|
| 1 | {1, 3, 5, 13} | Cuboid |
| 2 | {2} | Screw Holes –[SS] |
| 3 | {4} | {p14} |
| 4 | {6} | Screw Holes –[PSS] |
| 5 | {7, …} | {p4} |
| 6 | {8, 9} | Key Slots |
| 7 | {10, …} | Key Slots |
| 8 | {11} | Cuboid |
| 9 | {12, …} | Screw Holes –[DHB] |
| … | … | … |

**Fig. 5** The part function list and the related associations

since a key slot is for achieving such PFs as to allow the translation of a T-Bolt ([TB]) and to contain [TB], it is then associated with these two PFs (see association 8); since the key slot is manufactured by a milling cutter (MC), it should also achieve the function of allowing the rotation of the MC during the manufacturing stage (see association 9), which can explain why the key slot must have a half-round shape; since the neighbor space between two key slots (marked by {*p4*}) should has a function of accommodating adjustable wrench (AW) for assembling TB, an association can then be built (see association 5); etc.

According to the associations shown in Fig. 5, multiple lifecycle constraints can then be built. For examples, according to association 4, since the screw hole feature should have a function of semi-containing the pillar shelf screw part (i.e. [PSS]), a constraint can then be derived, i.e. $[BB]\text{->}ScrewHole_{[PSS]}\text{->}diameter\ (=)\ [PSS]\text{->}ScrewCylinder\text{->}diameter$; according to association 7, the [BB] part should have a function (i.e. to hold MC), a constraint can then be derived as: *[BB]->Key_Slot->width (=) MC->diameter*, which means that the width of the slot should also be determined by the diameter of a MC (a standard manufacturing tool), i.e. a designer cannot assign an arbitrary value to the width of a key slot. Besides such simple constraints, complex constraints can also be derived. For example, according to association 1, a complex association can be derived as: *[BB]->Cuboid->length (>) TS.length + [PS]->L_Extension->length + 2*([-NUT]->Nut->o_diameter)*. Here, *L_Extension* is a complex geometrical feature generated from extending an L-shaped polygon, *[NUT]* refers to a nut, which has a geometrical feature of Nut with a parameter *o_diameter* (i.e. outer diameter), while the number “2” means there should be two nuts assembled in one side. Note that the [SS] part doesn’t appear in this constraint since it is contained in TS when assembled together.

The above case demonstrates that the proposed approach can help designers capture various lifecycle requirements in detailed design. The PF concept allows a designer to conveniently manage various lifecycle objects related to the design of a part. In the above case, it not only allows a designer to manage the PFs involving fixture parts, but also those involving various lifecycle objects (e.g. milling cutter, adjustable wrench, etc.). Such PFs can tell designers what lifecycle factors have been considered in a part design. Based on the associations between PFs (i.e. EFCs) and geometrical features or neighbor spaces, the lifecycle requirements-capturing approach further enables designers to build quantitative lifecycle constraints. Such association knowledge and the related lifecycle constraints can provide explicit design knowledge about how those PFs are fulfilled by various geometrical objects, and therefore can help related designers (e.g. a chief designer) easily understand how a detailed design result is achieved.

## Related Work

There are multiple research topics that are related to our research, e.g. knowledge-based design, geometrical modeling, design rationale, and PLM, etc. Therefore, it is neither possible nor necessary to discuss all related work here. Instead, only some typical studies are discussed as below.

The first related work is Gero's [13] work on design prototype (more often known as the function-behavior-structure model). A design prototype describes multiple kinds of knowledge about similar design cases, e.g. function, behavior, structure, the relational knowledge, etc. However, unlike our work that is for managing detailed design knowledge, design prototype is primarily for organizing conceptual design knowledge. For example, Gero's function in design prototype primarily refers to the general function of a device, while our research deals with the function of a part. Furthermore, Gero's function merely deals with the external objects in the operation stage of an artifact, while our PF concept can deal with the objects in various lifecycle periods of an artifact.

The Structure-Behavior-Function (SBF) model proposed by Goel and his associates is also a significant work related to our research [3, 14]. Similar to the SBF model, our work also employs a structure-centered manner to organize design knowledge. However, there are multiple major differences between their SBF model and our work. For examples, their SBF model merely deals with qualitative knowledge for supporting conceptual design, while our approach involves quantitative constraints for supporting detailed design; as mentioned before, their function concept is more like the system function concept, but different from our PF concept; in the SBF model, function is achieved by causal behaviors, while in our approach, PF is achieved by either a geometrical feature or a neighbor space.

The Design Repositories Project (DRP) initiated in National Institute of Standards and Technology is another important work in knowledge-based design [15].

Similar to our work, DRP also deal with the representation of form and function. However, there are also multiple significant differences between DRP and our research. First, the function in DRP is based on the transformation of input–output flows and therefore is more like the concept of system function, while our research primarily deals with PF. Second, function and form are independent of each other in DRP, while in our research PF and form (geometrical or neighbor space) are closely associated for deriving lifecycle requirements. Finally, function in DRP primarily deals with system function-related flows, while function in our research deals with the external objects in various lifecycle periods of a product.

The design history (also known as design rationale) research is also closely related to our work. Based on the issue-based information system framework [16], Nagy et al. [17] propose an issue-proposal-decision-argument framework for organizing the knowledge about design history. Taura and Kubota [18] develop an experimental system for managing engineering design history with the aid of a semi-formal process information model. The primary difference between such design history studies and our research consists in that these studies primarily deal with representing conceptual design knowledge that is qualitative in nature, while our research aims at representing quantitative lifecycle knowledge for detailed design.

PLM (Product Lifecycle Management) is another important stream of related research. In industry, PLM is usually regarded as the process of managing the entire product lifecycle from conception, through design and manufacture, to service and disposal [19]. Originating from CAD systems and Product Data Management systems, PLM focuses more on managing geometrical information and the related data evolution process. Since existing PLM systems almost don't address the issue of representing function about a design, they fail to provide a formal and comprehensive framework to manage various lifecycle requirements (knowledge) that is necessary for understanding and evaluating a detailed design, which, however, is just the focus of our research.

The last related research stream that should be mentioned is constraint-based design, since the lifecycle requirements derived are represented design constraints. Constraint-based design research usually treats design as a constraint-based problem solving process, i.e. it assumes that all design constraints are explicit and available and a computer-based system should be developed for modeling those constraints and searching for a design solution that can meet all those design constraints [20, 21]. However, the fact is that many design constraints in detailed design are implicit. An important work that should be done is to elicit and capture those constraints to support design decision-making, which is just the primary concern of our research. In addition, unlike our research, constraint-based design also usually doesn't take into consideration various lifecycle constraints.

## Conclusions

Detailed design often involves various lifecycle requirements that are critical for various engineering design activities (e.g. understanding, auditing, reusing, learning, etc.). However, since existing CAD or PLM systems can merely manage geometrical information, such requirements often remain as tacit knowledge in human designers' minds, which can easily get lost. Therefore, this paper is devoted to developing a formal approach for capturing and managing various lifecycle requirements in detailed design.

Since detailed design primarily deals with designing the parts of a product, how to describe a part for modeling lifecycle requirements is introduced at first. A concept, *part function*, is then proposed for describing various lifecycle factors that should be considered in the detailed design of a part. Based on the function-constraining axiom, a part function-based approach is then proposed for capturing various lifecycle requirements, where PFs are either associated with geometrical features or neighbor spaces used for achieving them so that quantitative lifecycle constraints can then be naturally derived from those PF descriptions. Such lifecycle requirements actually can make clear how various lifecycle factors are considered in a detailed design.

A designing organization can benefit much from the captured lifecycle requirements. For example, they can help a chief designer judge whether various lifecycle factors have been correctly considered in a detailed design. They can also help designers understand existing detailed designs and then reuse them effectively. In addition, they can even help novice designers learn various lifecycle requirements about an existing design.

**Acknowledgments** We are grateful to the reviewers for their constructive suggestions. This research is supported by Natural Science Foundation of China (Granted No. 50975173 and 50935004), and Science and Technology Commission of Shanghai Municipality (Granted No. 09QA1402800).

## References

1. Pahl G, Beitz W (2007) Engineering design: a systematic approach, 3rd edn. Springer, London
2. Vijaykumar G, Chakrabarti A (2008) Understanding the knowledge needs of designers during design process in industry. ASME J Comput Inf Sci Eng 8:011004
3. Goel AK, Rugaber S, Vattam S (2009) Structure, behavior, and function of complex systems: the structure, behavior, and function modeling language. AI EDAM 23:23–35
4. Chen Y, Zhang Z, Liu Z (2011) Towards a scientific model of function-behavior transformation. In: Proceedings of the 18th ICED, Copenhagen, vol 2, pp 51–60
5. Chen Y, Zhang Z, Liu Z (2013) Towards a scientific concept of function for design science. AI EDAM (in review)
6. Bunge M (1977) Treatise on basic philosophy—Ontology I: the furniture of the world. D. Reidel Publishing Company, Dordrecht

7. Erden MS, Komoto H, Van Beek TJ, Tomiyama T (2008) A review of functional modeling: approaches and applications. AI EDAM 22:147–169
8. Maier JRA, Fadel GM (2009) Affordance-based design: a relational theory for design. Res Eng Des 20:13–27
9. Campbell M, Cagan J, Kotovsky K (2000) Agent-based synthesis of electromechanical design configurations. ASME J Mech Des 122:61–69
10. Chakrabarti A, Bligh TP (2001) A scheme for functional reasoning in conceptual design. Des Stud 22:493–517
11. Chen Y, Liu Z, Xie Y (2012) A knowledge-based framework for creative conceptual design of multi-disciplinary systems. Comput Aided Des 44(2):146–153
12. Szykman S, Racz JW, Sriram RD (1999) The representation of function in computer-based design. In: Proceedings of 1999 ASME DETC, September, Las Vegas, Nevada, DETC/DTM-8742
13. Gero JS (1990) Design prototype: a knowledge representation schema for design. AI Mag 11(4):26–36
14. Goel AK, Stroulia E (1996) Functional device models and model-based diagnosis in adaptive design. AI EDAM 10(4):355–369
15. Szykman S, Racz J, Bochenek C et al (2000) A web-based system for design artifact modeling. Des Stud 21:145–165
16. Kunz W, Rittel HWJ (1970) Issues as elements of information systems. Working paper 131. Center for Planning and Development Research, University of California at Berkeley
17. Nagy RL, Ullman DG, Dietterich TG (1992) A data representation for collaborative mechanical design. Res Eng Des 3:233–242
18. Taura T, Kubota A (1999) A study on engineering history base. Res Eng Des 11:45–54
19. John S (2004) Product lifecycle management: 21st century paradigm for product realisation. Springer, London
20. Lin L, Chen L-C (2002) Constraints modeling in engineering design. J Eng Des 13(3):205–214
21. Bilgic T, Fox MS (1996) Constraint-based retrieval of engineering design case. In: Proceedings of the 4th artificial intelligence in design, stanford, pp 269–288

# Part IX
# Design Processes

# Rule Based Stochastic Tree Search

**Mukund Kumar, Matthew I. Campbell, Corinna Königseder and Kristina Shea**

**Abstract** This work presents a new search process for composite decision processes (CDPs; also known as a tree-search problems [1]) that is especially suited to problems represented by grammars. Many of the methods that are used to find an optimal or near-optimal solution in a large tree have been developed for path-planning problems (like A* [2]) and thus have requirements that are not well suited to design problems. With the recent attention on grammars in design, we find that design trees are often produced but difficult to search. Since existing path-planning methods are sensitive to the size of the space, and often put a low priority on the number of objective function evaluations, it is imperative to develop new search methods that can find the best solution within a large tree by doing the least number of evaluations as possible. In a previous paper, an interactive algorithm for searching in a graph grammar representation was presented.

## Introduction

In this work, a graph grammar is used to solve graph optimization problem in two different problem domains. The first problem is solving a network optimization problem to improve the performance of Photovoltaic (PV) cell arrays under partial shading conditions by optimizing the connectivity layout between PV modules. The second problem uses tree search for selecting and configuring the components

---

M. Kumar · M. I. Campbell (✉)
University of Texas at Austin, Austin, USA
e-mail: mc1@mail.utexas.edu

M. Kumar
e-mail: kmukund87@gmail.com

C. Königseder · K. Shea
Technical University of Munich, Munich, Germany

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_31,

of electromechanical products. By using graph grammars, complex relations between components and their functions can be modeled using grammar rules. The problem is therefore to continuously apply rules to generate candidates within the design space.

Seed graphs and grammar rules are created specific to the problem at hand. Design generation is an iterative process of recognize–choose–apply where the seed graph initiates an expanding tree of intermediate solutions until no more rules are recognized. The leaf nodes of the tree are the completed candidate solutions. If the tree branches at an average branching factor of $b$, then there are $b^d$ states at level $d$. Thus an intelligent search technique is required to reduce the computation time for scanning such an exponentially large tree.

This work presents a technique that is based on RBITS method which stands for Rule-Based Interactive Tree Search [3, 4]. In that work, input from a decision maker was used to learn which rules and rule ordering lead to the best design. In this work, the learning mechanism is further put to the test by searching on larger trees with real-valued metrics of quality (i.e. objective functions). Both test problems are solved by the developed technique which is specifically suited for design spaces defined by grammars.

## Related Work

Engineering design uses graphs extensively for representation. In conceptual design synthesis, function structures [5] and flowcharts have provided a quick and clear way to describe large and complicated designs.

Graph grammars have been adopted for design synthesis for their ability to easily create topological changes to graphs in a systematic fashion [6]. A variety of problems such as mechanism synthesis [7], synthesis of gear trains [8, 9], sheet metal design [10], truss structures [11] have been represented by researchers using graph grammars. One of the advantages is the use of graphical software to display and interpret the grammar in an intuitive fashion [12] which speeds up problem definition and rule creation. In this paper, two existing grammar representations are used as test problems. The first example is the network synthesis of solar cell arrays; the representation of which is described in more detail in [13].

The second problem that is considered is conceptual design synthesis for electromechanical systems. This involves generating topologies of components for a specified function. Hundal [14] and Ward and Seering [15] have developed methods to solve this problem. However, both methods work by breaking the product into a functional model and find components to satisfy each function. Here, the graph grammar representation uses Function Structure and Component Flow Graphs (CFG) to represent electromechanical products such as compressors, blowers, hair dryer, etc. The work shows that millions of alternative designs solutions exist on a generative tree. In this work, we try to use search techniques in such a generative tree to find the best solution.

There are several tree search algorithms that can be used for searching graph grammar based generative trees. Exhaustive methods such as breadth first and depth first search scan all states in the tree and stop when a goal solution is found. A more efficient approach can be achieved by using heuristics function such as in methods like A*, IDA* and uniform cost search [16]. These methods require the heuristic to be monotonic and admissable and will fail when this is not guaranteed. Such a heuristic functions are rarely achievable in design problems. A search method developed by Schmidt and Cagan [17] uses a simulated annealing approach to navigate the tree of design solutions as if it were a multidimensional design space.

The method TP$^2$ (Topological and Parametric Tune and Prune) has been explicitly developed for graph grammar based problems such as sheet metal design [10]. In this, the problem is split into two searches: finding the best topology and optimizing the parameters of the topology. Using genetic algorithms for synthesis is an attractive option, but in their traditional form, they require a fixed-vector representation. Formulating the design problem as a vector is a challenge in many cases leading to long vector lengths. The generation of new solutions from the bit string by arbitrarily changing the individual bits can lead to infeasible designs that have to be restricted by the use of explicit or implicit constraints or using penalty functions. This greatly slows down the search process and a comparison of the TP$^2$ search technique to genetic algorithms shows that the graph-based search technique performs better because the generation process is unrestricted in terms of complexity but restricted to feasible states by the grammar [10].

One method to overcome this limitation in genetic algorithms is to pursue genetic programming [18], which represents candidate solutions as trees. While the method borrows heavily from genetic algorithms, the crossover principle—the main method to generate new solutions—is solved by grafting branches onto the tree candidates which are cut from other candidates. There may be possibilities to use a similar concept for the configuration candidates in the example problems discussed here. However, the limitation that candidates are trees as opposed to arbitrary graphs (which may have cycles) is not easily overcome in genetic programming. Additionally, the grafting of branches negates the need for a grammar that describes the construction. The implicit constraints stored in the grammar rules would ideally guide the grafting process, but this is also not easily solved.

Rule Based Interactive Tree Search is a method mentioned earlier for grammar based generative trees that adopts a completely different approach. By using rules as the fundamental building blocks in a design problem, RBITS uses the representation as an aid in guiding the search process. The work that is presented in this paper builds on this method by addressing larger trees with real-valued objective functions. In RBITS, the stochastic search was only guided by user comparisons between (greater than, less than, equal to) between two or more designs.

## Rule Based Interactive Tree Search

RBITS [3, 4] is a search technique that scans a portion of the tree in order to build information about the effectiveness of each grammar rule and the combinations of grammar rules so that insight is gained in finding the best solutions. By generating via rules, a correlation between the genotype (sequence of rules) and the phenotype (final candidate) is sought. If such a correlation exists, the search can make predictable changes to the quality of the solutions. Within the methodology, candidate solutions are evaluated and this evaluation is reflected onto the rules that were used in the generation of the candidate.

A database, called Rule Knowledge, keeps statistics on rules and the relative "fitness" from the candidates it has contributed to. It must be noted that each rule must have enough entries for the average fitness to give an accurate correlation. The number of entries for a given rule is called the popularity of the rule. Rules with high average fitness, are repeatedly invoked since, the search seeks the best solution. However, given a limited number of candidates that can be evaluated, this becomes a struggle between exploration and exploitation. One should exploit the knowledge that is known (choose high fitness rules), while continuing to explore unpopular rules that may possibly lead to even better solutions. This struggle is captured in the calculation of the overall rule score, $u$.

$$u = Bf + p(1 - B) + u_{\min} \tag{1}$$

In this equation, $f$ is the fitness of a rule and $p$ is the inverse of popularity. When $B = 0$, the process only selects unpopular options and leads to a well distributed exploration of all possibilities irrespective of their fitness value. When $B = 1$, the process selects options with the highest fitness or in other words, repeatedly exploits the best options while neglecting other possibilities. Setting $B$ between 0 and 1 thus captures the amount of exploration versus exploitation that occurs in the search. However, trusting that the option with highest $B$ value is best is a potential fallacy that restrains the search to be deterministically bound to the initial and random sampling. In order to break this restraint, a stochastic mechanism is introduced where the option scores (represented as $u_j$) are converted into probabilities of being chosen. This probability is calculated using the following equation.

$$Probabiltiy(option\ \mathrm{j}) = \frac{u_j^p}{\sum_{i=1}^{N} u_i^p} \tag{2}$$

The additional parameter $p$ can be changed from 0 to $\infty$ to change the process from random to deterministic. When $p = \infty$, it can derived that the option with highest $u$ value gets a probability of selection of 1 or 100 % and when $p = 0$ all options get an equal probability of selection. Since computers cannot handle values of p approaching infinity and since the sensitivity of p varies drastically over its range from [0 to $\infty$], this parameter $p$ is converted into another parameter $Q$ that takes values between 0 and 1 and maps to manageable values of $p$ (0 to 300). The conversion is done using the relation shown in Eq. 3.

$$p = 300 * Q^{8.229} \tag{3}$$

Note that the constants in this equation are chosen such that at a value of $Q = 1$, p is set to 300. When $Q = 0.5$, p is set to 1, thus producing a directly proportional probability to the $u$-score. These two adjustment parameters, B and Q, provide a mechanism for tuning the search process, and through the results shown below, effective values of these parameters are determined.

Returning to the Rule Knowledge database, the calculation of fitness and popularity can be broken down to provide more specific information. We have established three distinct types of extracted fitness and popularity: total, level and context. For the first type, this is manifest as total average fitness and total popularity. Total average fitness is determined by averaging the fitness of all of its entries within Rule Knowledge. Total popularity is simply the count of the number of times a rule has been previously called. Level fitness and level popularity filter down the entries within the Rule Knowledge database to those at a specified level within the search tree. For example, rule 1 may have been called twenty times before (total popularity) but rule 1 has only been called twice as the first level (level popularity). The development of level fitness and level popularity allows the search process the fidelity to learn *when* a particular rule may affect the quality of the candidate (i.e. near the beginning of the search tree versus the end, for example). Context fitness and context popularity offers an even higher fidelity investigation into when and where a particular rule has an effect. Depending on the implementation, context is defined as the list of rules that are invoked prior to the rule in question, or as the location of graph elements that the rule in question operates on. In the first problem shown below, it is the former; in the second problem, it is the latter.

## Photovoltaic Network Optimization

Photovoltaic (PV) cells are the fundamental blocks of a solar panel. PV cells are assembled to form a PV module and a solar panel power system consists of several interconnected PV modules which are usually connected in series to maximize the voltage. To maximize the power output, control techniques such as Maximum Power Point Tracking (MPPT) are used that controls the load on the generator in order to run it at the most optimum voltage for the maximum power [19].

In domestic applications, partial shading is a common problem that affects the output of the entire solar array. A small number of shaded PV modules can have a large and negative effect on the output. By modifying the connectivity between PV modules, some robustness can be achieved under partial shading conditions [20] to give a reliable output. Previous work demonstrates a suitable representation for this problem and a comparative study between two different approaches, using graph grammar and genetic algorithm (GA) [13]. While this work shows that the GA approach does not converge on a global optimum, the networks produced by randomly searching the graph grammar tree are not much better.

In a candidate graph, the connectivity between the modules is represented by arcs that connect the PV modules (i.e. nodes) in the four compass directions (North, East, South and West) as shown in Fig. 1. In this representation, the following restrictions are built into the rules to properly reduce the scope of the problem.

1. Nodes can only connect to neighboring nodes.
2. Nodes not connected to a neighbor are assumed to connect to the global positive or negative leads.
3. Cycles between nodes are not allowed.

The seed graph is an array of unconnected nodes representing the position of solar modules. The first four rules in Fig. 1 generate connectivity between adjacent nodes so long as the nodes are not already connected in that direction. Rule 5 specifies that a given node be unconnected to the neighbors and instead to the overall ground and lead of the solar panel. This is achieved by removing the label "empty" preventing the node from being recognized by the other rules.

## *Specifications of the Problem*

The upper limit of the candidate space for a square grid with 9 nodes (3 × 3 nodes) is $6^9$ (with connections N, E, W, S, ground, and + lead). A more accurate size of the space is complicated by the three restrictions presented above. As a network optimization problem, the goal is to find the best configuration of arcs that would prove most optimal. Here, optimality requires running a multitude of solar simulations for various shading conditions to evaluate the robustness of providing consistent electrical power. While we have developed the simulation and have connected it to the graph topology generation process, it has proven to be too slow for testing the effects of the aforementioned search process. Therefore, in order to simplify the objective function, a target configuration (see Fig. 2) is selected as the chosen optimal and the objective function is simply the difference of a candidate from the optimal. Since the candidates are graphs, the difference is determined as the subtraction of the graphs' corresponding vector representations. The vector is created such that it is a series of 0s and 1s representing the connectivity of each node to the neighboring nodes and to the global positive and negative terminals. As the number of nodes remains the same, the vectors are of equal length and the Euclidean distance is computed and used as the distance between the two vectors. The objective function thus becomes unimodal where the magnitude is 0 at the optimal solution.

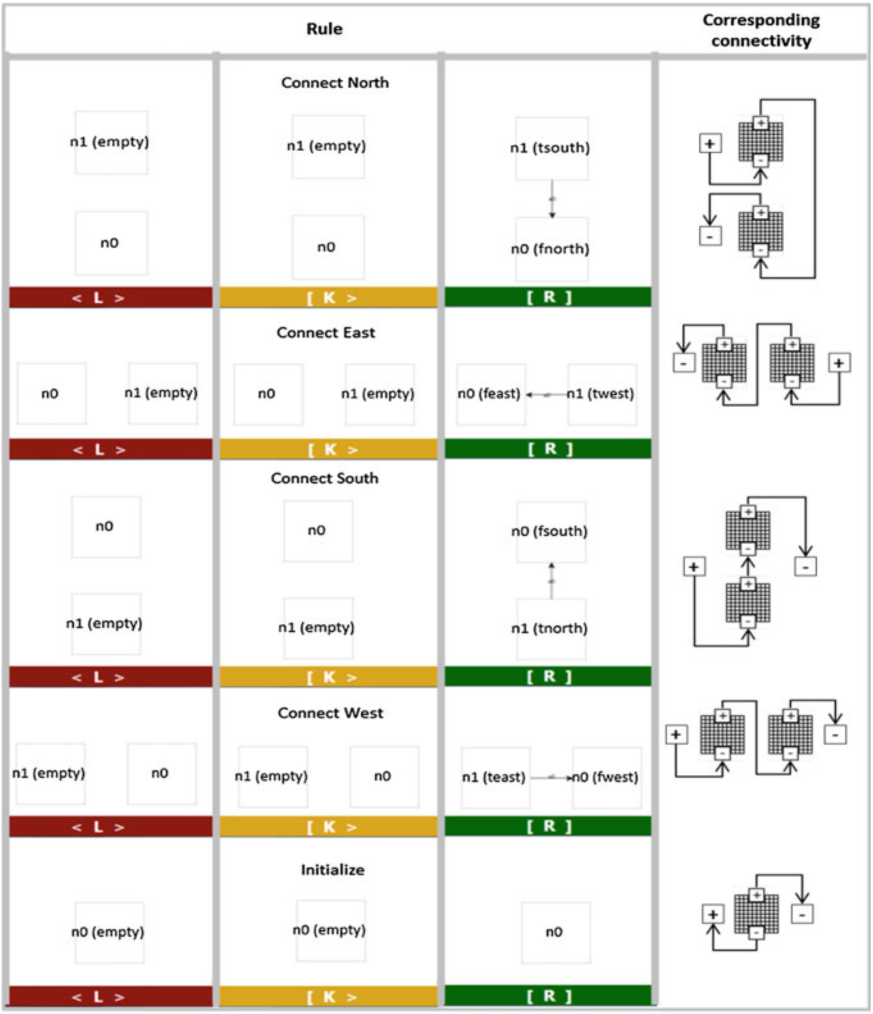


**Fig. 1** Rules for creating connection between nodes in a seed graph. Figures show rules to create arcs in N, E, S, W directions. Rule creates arcs to connect to global positive-negative terminals

## *Results*

The rule-based stochastic tree-search process and the above modifications are written in C#. In order to make the process of manipulating graphs and invoking grammar rules easier, the code is packaged as a plugin for the GraphSynth [12] software. As in a typical optimization problem, convergence is identified when the objective function does not change significantly over a series of iterations.

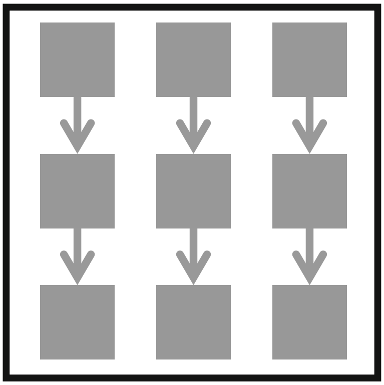


Fig. 2 The target solution is a balance between maximizing power and robustness

Additionally, in this search process, convergence can also be correlated with a stagnation in the knowledge repository.

It is assumed that the best candidate that can then be generated from a given repository of rule knowledge is achieved by starting from the seed graph and applying the options with the highest fitness at each level with no regard of popularity (i.e. $B = 1$) and no randomness ($Q = 1$). This represents a deterministic and exploitative search of the space.

Once the repository has sufficiently converged, adding additional candidates to it will yield negligible changes in the option fitness values and therefore the best candidate generated would not change significantly. This can be used to detect the convergence of the process. A candidate can be generated with values $B$ and $Q$ as 1 at every step and evaluated. Once the rating of this candidate stabilizes (i.e. remains within a certain tolerance), it can be assumed that the knowledge repository has attained a stable state and the process has converged. Figure 3a shows the variation in the objective function value for the solar panel problem with B and Q values of 0.2 and 0.8 respectively. It can be seen that the search algorithm eventually converges to find the candidate of rating 0.0 implying the distance of the candidate from the ideal solution. To prove the reliability of the process despite the stochastic nature, the same graph is recreated in multiple runs as shown in Fig. 3b, c. To obtain a convergence curve that better summarizes the effectiveness of the process, similar graphs are created for 20 runs and they are averaged over the iterations. The resulting graph is shown in Fig. 3d. In the remainder of the paper, we examine versions of Fig. 3d for various values of B and Q.

## *Experimenting with* Q *Values*

Consider the plots in Fig. 4. While a given plot shows six curves, each of these curves is an average of 20 complete search processes. The experiment is run with $B$ values from 0 to 1 with a step size of 0.2 while $Q$ takes the values 0.1, 0.3, 0.5,

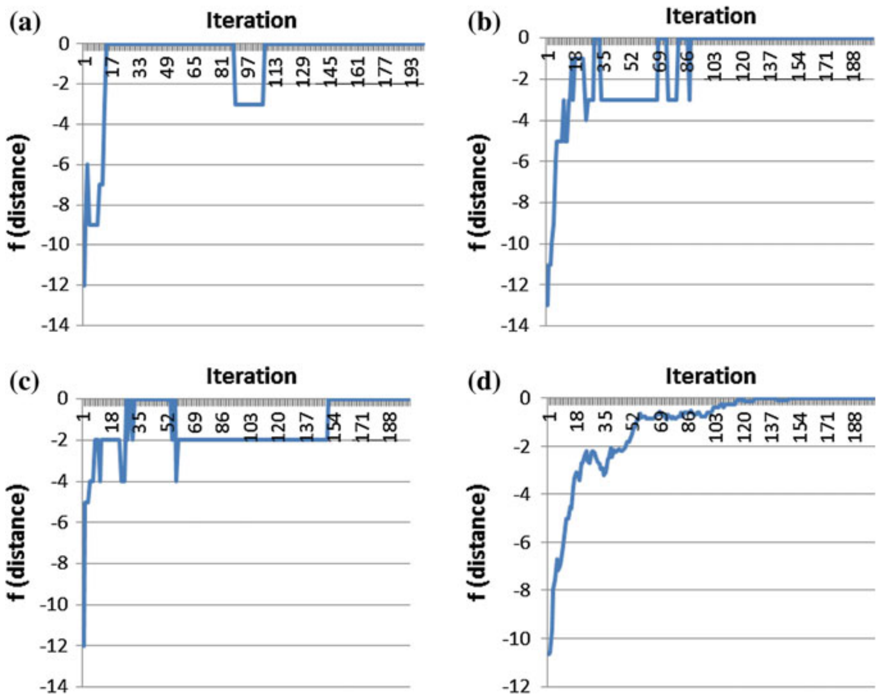


**Fig. 3** Convergence of rating versus number of iterations. Different runs produce different output (**a**, **b**, **c**). Plot **d** shows average of 20 such runs

0.7, 0.9. When $Q = 0.1$ (Fig. 4a), the process is nearly random. From the point of view of an optimization algorithm, this is an uninformed search where different candidates are evaluated without making intelligent decisions on the direction of exploration. However, as more candidates are explored, eventually almost all rule options will be added to rule knowledge. In examining the curves in Fig. 4a, we note that the process reliably achieves the optimum. This is because what is plotted is the best design achievable for the given rule knowledge repository—the single solution generated when $B$ and $Q$ are both set to 1. The result when $Q = 0.1$ is nearly the same as that for $Q = 0.3$ and $Q = 0.5$. Upon closer inspection (as is indicated in Table 1), there is consistently better results for $Q = 0.5$.

On the other hand, in the nearly deterministic case, when $Q = 0.9$, the process repeatedly selects only the best option. This leads to a quick convergence on a suboptimal candidate as shown in Fig. 4e. The results are also highly dependent on the starting point since the same option is repeatedly chosen at each decision point in the tree. When Q = 0.5 the rule knowledge has enough information for predicting future directions of search but a healthy level of uncertainty or skepticism is maintained.

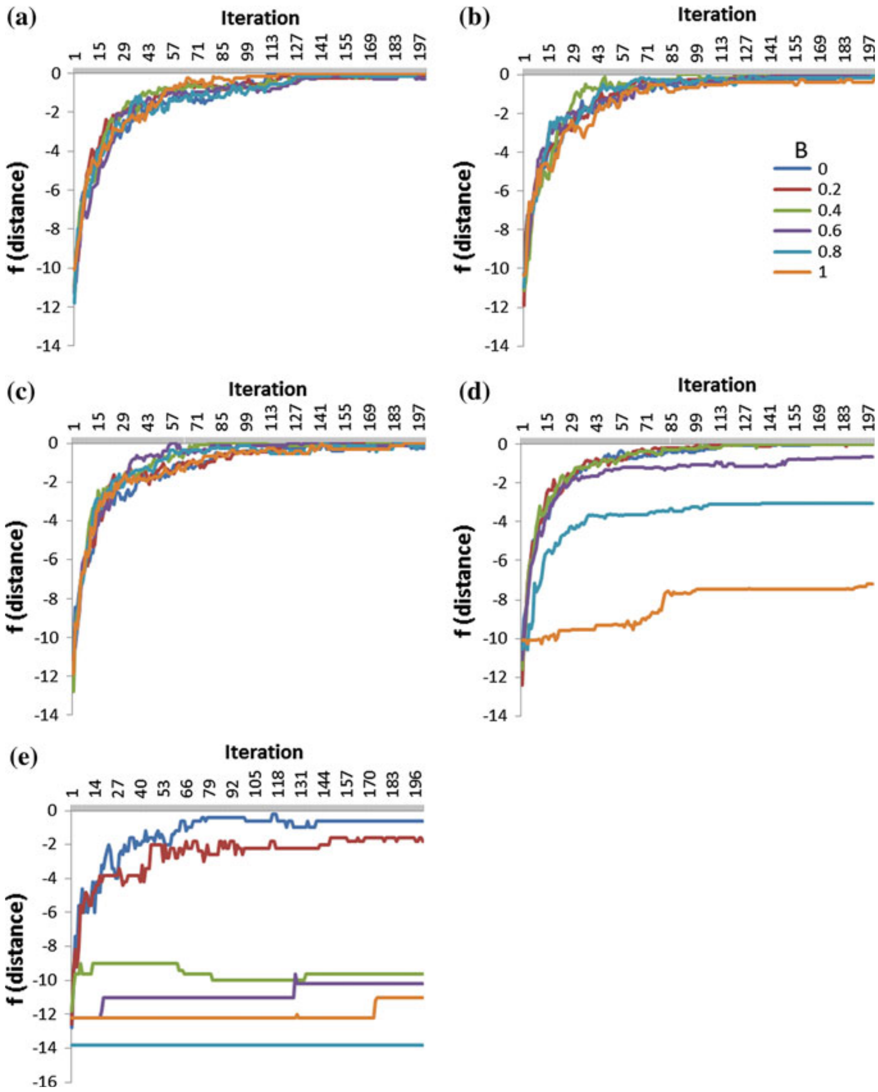


**Fig. 4** Progress of average rating versus number of iterations with different sets of B and Q values grouped by Q. **a** Q = 0.1. **b** Q = 0.3. **c** Q = 0.5. **d** Q = 0.7. **e** Q = 0.9

## *Experimenting with B*

Parameter $B$ controls the balance of exploration versus exploitation in the search. Using $B = 0$ implies that we are wholly interested in exploring more options rather than being restricted to the best. Regardless of the fitness value of each option, the option that has been explored the least receives the highest score. The search can

**Table 1** Variation of convergence and rating with B and Q

| | | Iteration of convergence | | | | | Average of best objective function value | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Q | | 0 | 0.3 | 0.5 | 0.7 | 1 | 0 | 0.3 | 0.5 | 0.7 | 1 |
| B | 0 | 114 | 118 | 120 | 116 | 139 | −0.11 | −0.10 | −0.25 | 0.00 | −0.6 |
| | 0.2 | 168 | 129 | 120 | 98 | 165 | 0.00 | 0.00 | 0.00 | 0.00 | −1.6 |
| | 0.4 | 131 | 93 | 76 | 118 | 133 | 0.00 | −0.15 | 0.00 | 0.00 | −9.6 |
| | 0.6 | 131 | 131 | 83 | 156 | 127 | -0.20 | 0.00 | −0.15 | −0.65 | −10.2 |
| | 0.8 | 130 | 134 | 161 | 108 | – | 0.00 | 0.00 | −0.15 | −3.05 | −13.8 |
| | 1 | 104 | 105 | 185 | 194 | – | 0.00 | −0.25 | 0.00 | −7.20 | −11 |

be effective at finding near-optimal solutions by building rule knowledge in an explorative way (again what is plotted in Fig. 4, is the best possible solution that can be found for the developed rule knowledge—$B = Q = 1$—deterministically exploiting what the rule knowledge contains).

At the other extreme, when $B = 1$, pure exploitation is achieved. Options of low fitness are not revisited as it is assumed they lead to a worse candidate. Without any exploration, the search always prefers options it has seen before, thus making the search highly dependent and limited to the initial random candidate. In the curves shown in Fig. 4 that correspond to $B = 0.8$ and $B = 1.0$ seem to appear the worst. This shows the importance of exploration in building the rule knowledge.

## *Selecting Best Values of B and Q*

For each run with a different $B$ and $Q$ value, the number of iterations to convergence and the distance of the final candidate to the optimal are calculated and tabulated in Table 1. The "average of best objective function value" is the found by averaging the value of the final solution that each process converges to. The iteration of convergence is calculated as the iteration at which the process fluctuates less than 1 % in the range of values seen by the process. The best values for B and Q are at neither extreme. The objective function values are more uniform even at low values of $Q$ but take a long time to converge. The best values of $B$ and $Q$ can thus be chosen as 0.4 and 0.5.

## *Application to Problem of Higher Complexity*

Problems with higher number of PV modules to connect converge slower due to there being more options in the search tree. The upper limit of candidates in the problem with 9 nodes is $6^9$, which is 10 million, and so the problem does not represent a high degree of complexity. To demonstrate effectiveness, we use an initial problem with 24 nodes and a target optimal graph that contains an arbitrary

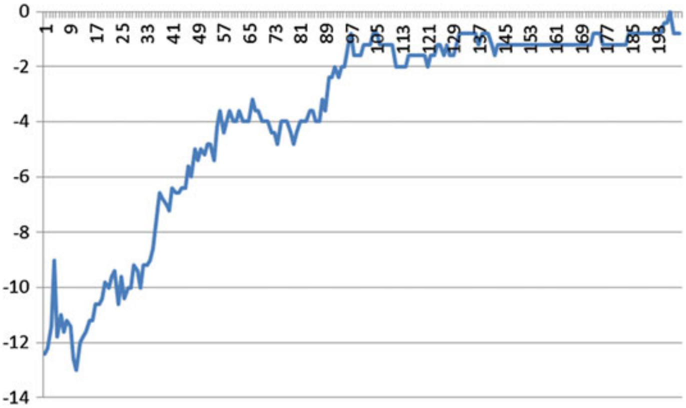


**Fig. 5** Candidate rating versus iteration number for a seed of size 24 nodes

connectivity between the nodes. The number of possible configurations grows exponentially and the upper limit of variations in the case of 24 nodes is $6^{24}$ which is $4.7 \times 10^{18}$. The size of the problem is significantly larger and convergence on a problem of this size will show the effectiveness of this method in searching over a large solution space.

The search process runs for 200 iterations and to offset the stochastic nature of the process, 20 identical runs are executed and the results are averaged. The value of B and Q are set to 0.4 and 0.5 as determined from the results of the previous experiment. The convergence graph is shown in Fig. 5. Even with such a large space of possible solutions, the search process has converged within a distance of 2 from the target solution within the first 100 iterations. The process also continues to improve by smaller amounts after the 100th iteration and at the end of 200 iterations has, on average, converged on a better solution (the final value is $-1$ as averaged over the 20 runs).

## Product Assemblies

The PV optimization problem can be characterized as a problem with a small set of rules that apply to many locations in the host graph. A second category of problems that we consider contains many rules with a relatively small set of locations for each rule to apply.

Electromechanical products such as fans, leaf blowers and staple guns are popular in the consumer market and many different companies have their unique designs that offer advantages in terms of packaging, efficiency, ergonomics and styling. In the conceptual design of such products, graphs are likely used such as Function Structures (FS) to capture the interaction between different functional

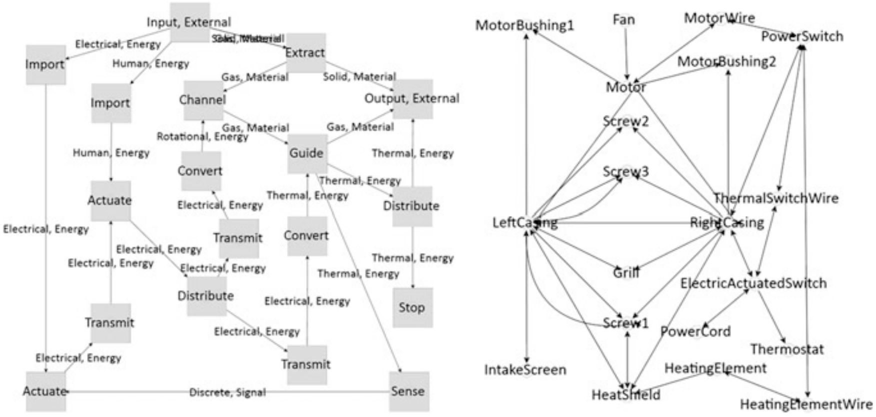


**Fig. 6** Screenshots of function structure (*left*) and component flow graph (*right*) of a common hair dryer

modules of the product. The FS is realized by actual components that are connected as a liaison graph or a Component Flow Graph (CFG). An example of a FS and a CFG are shown in Fig. 6, and were obtained by disassembling a hairdryer. From a previous research [21, 22], the FS and CFG of twelve different products were determined which are listed in [23, 24].

An example grammar rule is shown in Fig. 7. It can be observed that it creates a mapping between a single function, Convert, and two components while capturing the interaction between the components. Ninety rules were generated from the ten products using an automated rule generator which extracts rules from these products by comparing their Function Structures and CFGs. A generative tree is created by setting the seed graph as a new function structure and applying these 99 rules on it. A variety of solutions can be created and each solution represents a method showing how concepts and components from other products can be reused to inform the redesign a product (represented as a function structure). By including rules from other products, the original CFG of the product is just one solution in a large search space. This problem space and rules are described in more detail in [22–24]; it is used as an example problem for the developed rule-based stochastic tree-search (Table 2).

## *Problem Specification*

Similar to the previous problem, a pre-defined solution is identified as the best solution and the aim of the search process is to converge as close as possible to this target. To find the best values for the search parameters $B$ and $Q$, similar experiments with the same ranges as shown in the previous problem. These are repeated

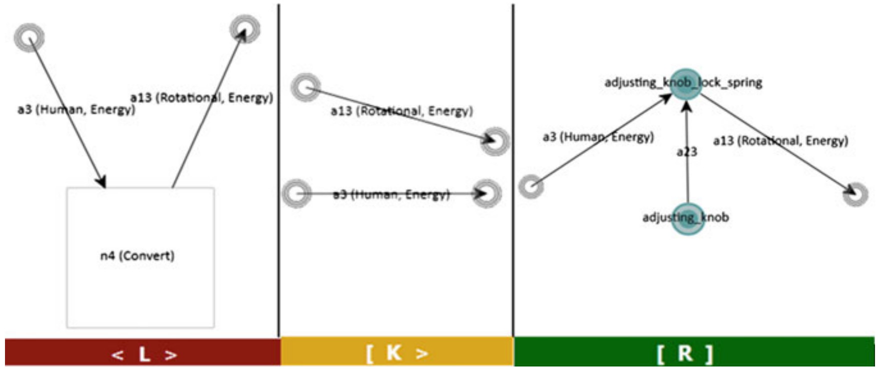


**Fig. 7** Rules transforming single function mode to multiple components

**Table 2** Candidates from which rules were generated

| ID | Product name | # functions in FS | # components in CFG |
|---|---|---|---|
| A | Common alternator | 13 | 36 |
| B | Hair dryer | 18 | 20 |
| C | Hydraulic jack | 18 | 64 |
| D | Lycoming T53 turbine | 14 | 20 |
| E | Portable air compressor | 25 | 64 |
| F | Proctor toaster | 17 | 54 |
| G | Squirt gun | 21 | 63 |
| H | Stanley staple gun | 26 | 60 |
| I | Troy bilt leaf blower | 10 | 19 |
| J | VW bug carburettor | 20 | 58 |
| Average | | 18.2 | 46 |
| Min | | 10 | 19 |
| Max | | 26 | 64 |

20 times to compensate for the stochastic nature of the search process. In these runs, the best obtainable value is −2,000.

## *Results*

Similar trends are observed as in the case of the PV array optimization problem. Table 1 tabulates the average of the best objective function value achieved and the iteration of convergence for each case of $B$ and $Q$ values. For values of $B$ and $Q$ for which the rating is the highest, the number of iterations to convergence is also higher. For example the case with $B$ and $Q$ as 0.0, it takes 62 iterations to converge when compared to the case with $B$ and $Q$ as 1.0 where it converges in the first step.

But in the latter, the maximum rating of the candidates is much lower than the ideal obtained in other cases (−2,817 compared to −2,042). To strike a balance between the two, the best values obtained are again when *B* and *Q* values are 0.4 and 0.5 respectively. Coincidentally, the results are identical as in the case of the PV array optimization problem. The number of evaluations done is far minimal when compared to the number of possible solutions. Previous work shows that the possibility for number of candidates is on the order of hundreds of thousands and the reduction in the number of iterations shows a significant reduction in the time complexity of the search method.

## Conclusions and Future Work

The work presented here shows how a graph optimization problem can be solved using a generative tree produced by a grammar. The results show that achieving a near optimal solution for a problem space containing an unmanageable number of solutions is possible within a few hundred iterations. The proposed method is limited to problems that use grammar rules as the fundamental building blocks since the statistics gathered on the rules is the basis of the method. Apart from this, optimum values of the two key parameters *B* and *Q* were also obtained by experimentation, and are coincidentally the same in both experiments ($B = 0.4$; $Q = 0.5$). The method has been shown to work on two kinds of problems, one with a few rules but many locations for the rules to be applied, and one with many rules but few locations. Many problems in graph grammars involve a similar situation and can possibly be solved by this method.

There is considerable scope for future improvement. The objective function in both problems is a distance to a chosen optimum and thus creates a unimodal search space. Certain modifications will need to be made to be effective on a multimodal problem. For example, in the PV array problem, two configurations could exist with a 3-by-3 grid of nodes, one having all arcs pointing south (as is shown in Fig. 2) and the other having all arcs pointing east. Although these two configurations are equivalent, they appear different to the implemented distance function. If both these solutions were to receive good objective function values, the process will then give equal fitness for arcs pointing south and east. The rule knowledge might create a "best" solution (when *B* and *Q* are equal to 1) that contains a mix of both types of arcs and as a result be far from optimal. To solve a problem of this kind, we will need to put more emphasis on the *context* measure that assigns a fitness for a rule choice based on which rules have been currently invoked in the candidate. Detailed study needs to be done to verify that context fitness can positively affect such multimodal problems.

Another search process improvement that can be made is to modify of *B* and *Q* values of the search process as a function of time similar to methods such as simulated annealing. The process could gradually move towards exploitation near the end when a reasonable amount of scanning has been done. Like simulated annealing, the rate at which the process becomes deterministic needs to be answered.

There are some other issues that could be investigated to study the effectiveness of the process. First, the same problems need to be solved using other techniques such as genetic algorithm to benchmark the performance and provide insight for further improvement. Such efforts are complex since many changes need to be made to these methods so that they can work on grammar-based systems.

In this work, the RBITS method is extended for use on larger search trees where a real-valued objective function is used to inform rule fitness. While grammars have proved to be useful in representing complex design domains, their unique qualities make it difficult to plug into existing optimization or artificial intelligence methods. The presented Rule-Based Stochastic Tree-Search method is based on handling the exploration versus exploitation balance needed to effectively search large trees. It uses the knowledge intensive quality of rules to guide the search to near-optimal solutions.

**Acknowledgments** The authors would like to acknowledge the support of the Institute for Advanced Study at the Technical University of Munich for funding this collaborative research.

## References

1. Kanal LM, Kumar V (1988) Search in artificial intelligence. Springer, London, p 482
2. Russell SJ (1995) Artificial intelligence: modern approach. Prentice Hall, Englewood Cliffs
3. Campbell MI, Rai R, Kurtoglu T (2009) A stochastic graph grammar algorithm for interactive search. In: Proceedings of the ASME 2009 international design engineering technical conferences IDETC/CIE 2009, vol 2009
4. Campbell MI, Rai R, Kurtoglu T (2012) A stochastic tree-search algorithm for generative grammars. J Comput Inf Sci Eng
5. Pahl G, Beitz W (1996) Engineering design: a systematic approach. Springer, London
6. Ehrig H, Rozenberg G, Kreowski H-J, Montanari U (eds) (1997) Handbook of graph grammars and computing by graph transformation. World Scientific Publishing Company, Singapore
7. Radhakrishnan P, Campbell MI (2010) A graph grammar based scheme for generating and evaluating planar mechanisms. In: Gero JS (ed) Design computing and cognition'10. Kluwer, Dordrecht, pp 663–679
8. Swantner A, Campbell MI (2010) Automated synthesis and optimization of expanded gear train topologies. Eng Optim (under review)
9. Lin Y-S, Shea K, Johnson A, Coultate J, Pears J (2009) A method and software tool for automated gearbox synthesis. In: Engineering conference, pp 1–11
10. Patel J, Campbell MI (2010) An approach to automate and optimize concept generation of sheet metal parts by topological and parametric decoupling. J Mech Des 132(5):51001
11. Shea K, Cagan J, Fenves SJ (1997) A shape annealing approach to optimal truss design with dynamic grouping of members. J Mech Des 119:388–394
12. Campbell MI (2012) GraphSynth | GraphSynth2: software for generative grammars and creative search. Available online: graphsynth.com. Last accessed 01 June 2012

13. Königseder C, Shea K, Campbell MI (2012) Comparing a graph-grammar approach to genetic algorithms for computational synthesis of PV arrays. In: CIRP design 2012—Sustainable product development
14. Hundal MS (1990) A systematic method for developing function structures, solutions and concept variants. Mech Mach Theory 25(3):243–256
15. Ward AC, Seering WP (1993) The performance of a mechanical design 'compiler'. J Mech Des 115(3):341
16. Pearl J, Korf RE (1987) Search techniques. Ann Rev Comput Sci 2(8):451–467
17. Schmidt LC, Cagan J (1995) Recursive annealing: a computational model for machine design. Res Eng Des 7:102–125
18. Gielen C (1994) Genetic programming. Neurocomputing 6(1):120–122
19. Esram T, Chapman PL (2007) Comparison of photovoltaic array maximum power point tracking techniques. IEEE Trans Energy Convers 22(2):439–449
20. Dougal RA, Iotova AP (2009) Parallel-connected solar PV system to address partial and rapidly fluctuating shadow conditions. IEEE Trans Industr Electron 56(5):1548–1556
21. Kumar M, Campbell MI (2010) Organizing a design space of disparate component topologies. In: Gero JS (ed) Design computing and cognition'10. Kluwer, Dordrecht
22. Kurtoglu T, Swantner A, Campbell MI (2010) Automating the conceptual design process: from black box to component selection. AI EDAM 24(01):49–62
23. Kurtoglu T, Campbell MI (2006) Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. Res Eng Des (6) (in review)
24. Kurtoglu T, Campbell MI (2009) Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. J Eng Des 20(1):83–104

# Capturing Ideation Paths for Discovery of Design Exploration Strategies in Conceptual Engineering Design

**Manikandan Mohan, Jami J. Shah, Sumit Narsale and Maryam Khorshidi**

**Abstract** In conceptual design, the ideation state of a designer changes over time. As the designer gets deeper insight, discovers hidden requirements and conflicts, finds poor fit between proto-solutions and requirements, he/she adjusts or abandons particular directions of investigation. Thus, there is a continual change in issue formulation and solution strategy. We have created an instrument which facilitates the capture of the sequence of search and solution strategies in association with ideation states. We term this history of an individual's process in generating design concepts, the "Ideation Path". Our instrument provides the designer access to a range of intuitive and experiential ideation methods and tools, such as TRIZ, Bio-mimetics, Physical Effects/Working Principles, Design Repositories and several others. Ideation states are characterized in terms of ideation blocks, such as fixation, and the current level of production or satisfaction with ideas generated (quantity, quality, variety, novelty). We hope to use this instrument in collecting data from large numbers of designers and problems, in order to assess the effectiveness of different methods and paths in generating creative solutions.

## Introduction

There are a variety of ideation methods that aid a designer to develop creative solutions for a design problem. Ideation methods can be classified into intuitive methods and logical methods [1]. Intuitive methods are techniques which depend on the knowledge of the designer and do not depend on any catalogs or physical principles. They rely on chance. Some examples of intuitive ideation methods are

---

M. Mohan (✉) · J. J. Shah · S. Narsale · M. Khorshidi
Arizona State University, Tempe, AZ, USA
e-mail: mmohan5@asu.edu; manee@asu.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_32,

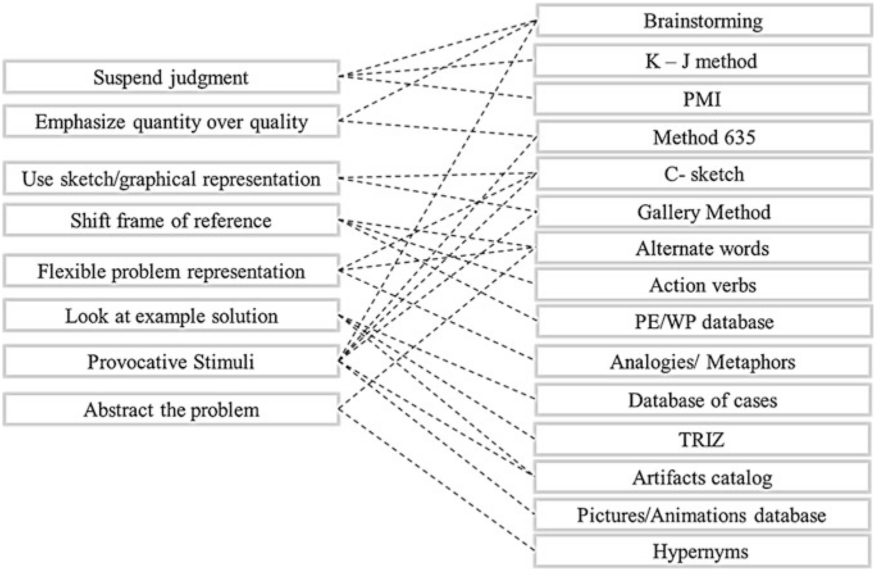


**Fig. 1** Ideation strategies embedded in Ideation methods

Brainstorming, C-sketch, Gallery method etc. Since these methods are not based on catalogs or principles, the novelty of ideas obtained using these methods is generally high, though the quality may be low [1]. On the other hand, logical ideation methods depend on physical principles, catalogs and databases. Idea generation using logical ideation methods is systematic relying on past solutions, so the novelty of ideas obtained is generally low when compared to the intuitive methods. Some examples of logical ideation methods are physical effects based catalog [2], working principles catalogs [3], bio-mimetics like Ask-Nature [4], Design repository relating functions to artifacts [5]. Certain techniques consist of both logical and intuitive ideation methods. Some examples of mixed ideation methods are TRIZ [6], Morphological charts [7] and Factorization methods.

Cognitive mechanisms, which we term *ideation strategies*, are embedded in every ideation method. Proponents of these strategies claim that they promote creativity. For example, ‘suspending judgment’ is a component of Brainstorming, while ‘provocative stimuli’ is a component of many ideation methods like C-sketch [8], Method–635 [9] and Gallery method [10]. Figure 1 shows a list of ideation strategies and corresponding methods in which they are embedded. Whether the claim is true can only be verified through empirical studies, a number of which have been done by various researchers in the past two decades.

The ideation state defines the current understanding of the design space and one’s current location in that space. It describes the current focus of the problem solver, factors that block creativity, elements that prevent exploration of new paths and reasons for not making progress. Specific states where the designer does not

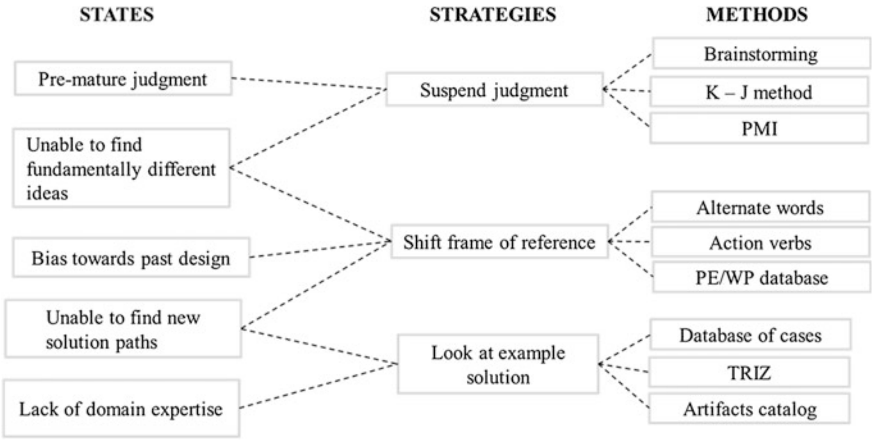


**Fig. 2** Example relationships between ideation methods, strategies and states

make progress are called Ideation blocks [11]. A block does not imply just lack of ideas, but also dissatisfaction with the novelty, quality and variety of ideas generated. Ideation strategies help overcome ideation blocks. The basic premise of this paper is as follows: If a variety of ideation methods were made available to designers, they would be able to pick and choose methods that are most effective in breaking ideation blocks, methods that contain the most effective strategies.

Until recently, research on ideation strategies and states is mostly limited to analysis of a single ideation method. Study of sequence or combination of strategies, states and methods is an area of conceptual design research yet to be explored. We believe that capturing ideation paths can be a rich source of data on the effectiveness of different strategies and the conditions (states) under which they work or don't, see Fig. 2.

The rest of this paper explains tracking and formalization of Ideation paths followed by a few example case studies which illustrate the difference in ideas generated when different ideation paths were taken.

## Capturing Ideation Paths

Capturing design history is not a new idea, but this is typically done to re-use history for similar design problems and cut down on development time [12–14]. However, our purpose is completely different: not re-use but data mining to evaluate the effectiveness of ideation strategies under different conditions. Shah [1] identified several ideation components in ideation methods like C-sketch [8]. Mohan et al. [15] made observations on different ideation strategies and ideation states that can be overcome. In this work, we have take design history in a new direction of research.

We consider the following items as part of capturing ideation paths:

- Sequence of Ideation states
- Sequence of ideation strategies/methods used
- Time spent on each ideation method (IM)
- Combination of Ideation states and Ideation strategies
- Duration of ideation states (IS)

As seen in Fig. 3, the paths can be tracked as a sequence of ideation methods, states, and time spent, or as a combination of ideation states and strategies or combinations of ideation states and time duration spent. Ideation paths can be visualized as flowcharts or graphs and can be queried for:

- Appropriate path to solve a function
- Appropriate path to solve a state (block mostly)
- Appropriate path for different human variables (Experience, knowledge, creativity, expertise and background)
- Appropriate path for different problem types (based on complexity, difficulty, decomposability and degree of innovation needed)

Several approaches can be used to track and document ideation paths:

- Computer tools (automatic capture)
- Protocol analysis [16]
- Case studies [17]
- Controlled experiments [18]
- Introspection

Using computer tools to capture ideation paths is most attractive for collecting massive amounts of data because it will make data mining much easier and can be carried out in the background. But it requires a holistic ideation tool that allows users to pursue different strategies at will.

Protocol analysis [16] can be used to directly observe designers engaged in design activities or similar observations can be done on recorded transcripts. Although protocol analysis is good for short term cognitive processes and very early stages of research, it takes a long time to analyze data recorded during protocol studies. Furthermore, there is no standard method for analysis and inconsistencies may arise when the same data is analyzed by different observers. Case studies are also done in a similar way. Case studies [17] are more focused compared to protocol studies since only specific parts of documentation and transcripts are analyzed.

Another method to track ideation paths is to run controlled experiments [18] with different design, human, method and environment variables. Observation from such experiments will be very useful to characterize different ideation paths based on such variables. Statistical factorial analysis can also be done to find the effect of different variables, the effect of their interaction on ideation paths. Additionally ideation paths can also be identified by introspection. During introspection, the designer needs to document his own path in a separate document. It is

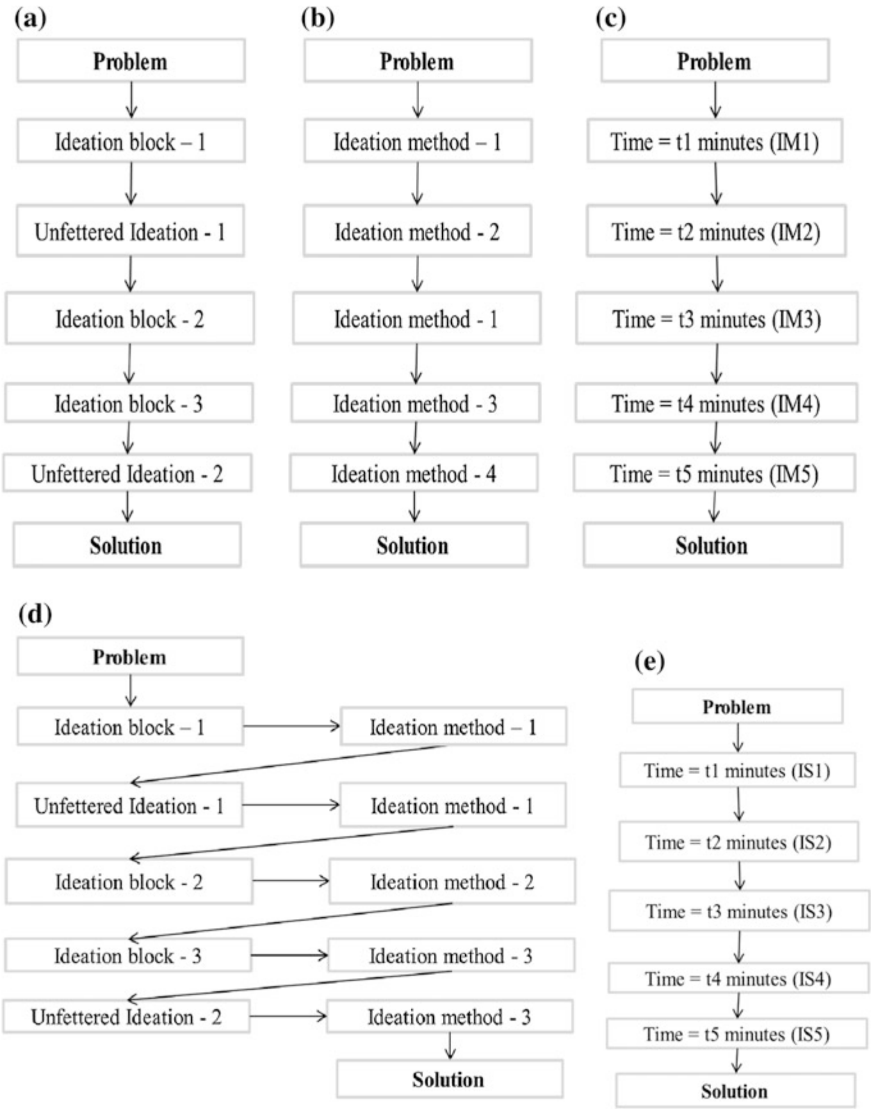


**Fig. 3** Different ways to formalize ideation paths (**a**, **b**, **c**, **d** and **e** 5 types of visual representation)

not a formal method, but just self-observation done in addition to idea generation which may hinder the designer's creativity.

Case studies need to be run for long durations in order to document a full ideation path where designer uses different ideation strategies and generates different creative solutions. Running shorter case studies may result in inadequate information. However, doing a protocol analysis on a long case study is a tiring process. Also, in all these approaches, the designer needs to be provided with a variety of strategies so that he can use them at different instances. Although, providing the designer with various ideation tools created by different owners is feasible, it would be better if the designer is facilitated with a holistic conceptual design tool where he can traverse between different ideation strategies seamlessly. In addition, it is known that controlled experiments help us draw conclusions on the effect of different variables on ideation paths. Hence, an appropriate method to track and analyze ideation paths is to conduct controlled experiments on a holistic ideation platform.

## Holistic Ideation Computer Tool

We have made considerable progress in developing a holistic framework for ideation [15]. The tool consists of several diverse ideation methods which can be used at different instances under varying ideation states. The tool queries the user at various stages about the state of the design and gives the designer access to several different internal and external ideation tools [15]. An overview of the tool is given here, see Fig. 4.

The following experiential methods are available in the tool: artifact mapping based on function decomposition, conflict resolution based on TRIZ, VDI systematic synthesis principles (working principles and physical effects) and Bio-TRIZ. For intuitive methods, we provide: Provocative Stimuli in the form of pictures, animations, sounds, words; Suspend judgment and Incubation (suggestion to take a break or work on an alternative problem). This constitutes a rich enough set for this study as it spans the entire spectrum of abstraction levels, from embodiment, to generalized principles to most microscopic level of physical effects. Figure 4 shows a high level architecture.

The computer tool supports three phases: Pre-ideation, Ideation, Post-Ideation. Figure 5 shows some of the user interface windows of the tool.

In pre-ideation, one does function decomposition using FunctionCAD from Oregon State. The designer can choose a sub-function to work on and proceed to the ideation stage. The post ideation stage involves documentation of ideas, evaluation of ideas, taking a survey, and grouping/organizing ideas into morphological charts. These cycles can be repeated as many times as the user wishes. The ideation methods are implemented as applications that can be invoked by the designer at appropriate times. All through the idea generation process, time is also monitored and each idea that is generated is attributed with a time stamp. This time tracking process is done in order to find the idea(s) that was generated during the Creative ‘leap’ or ‘dip’ which can be identified from the plots used to track the evolution of ideation states.

The designer answers some questions that are used to determine his ideation state. Based on that, a list of appropriate methods is suggested; the user can make a

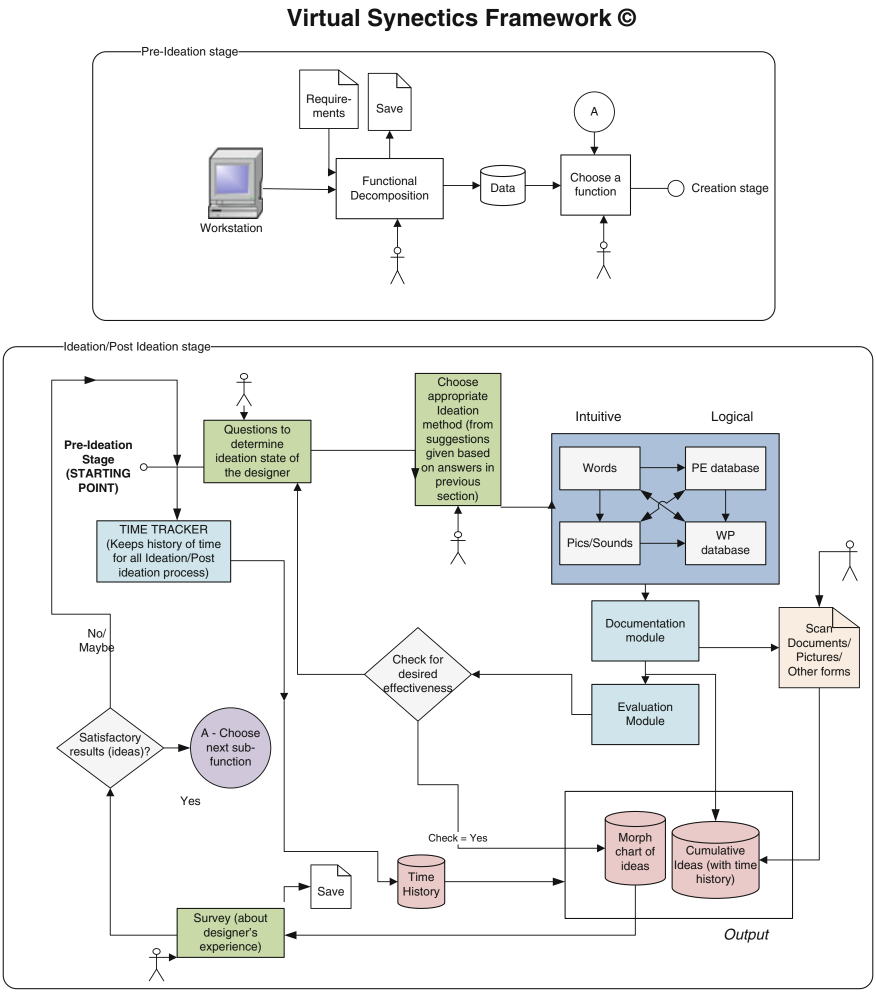
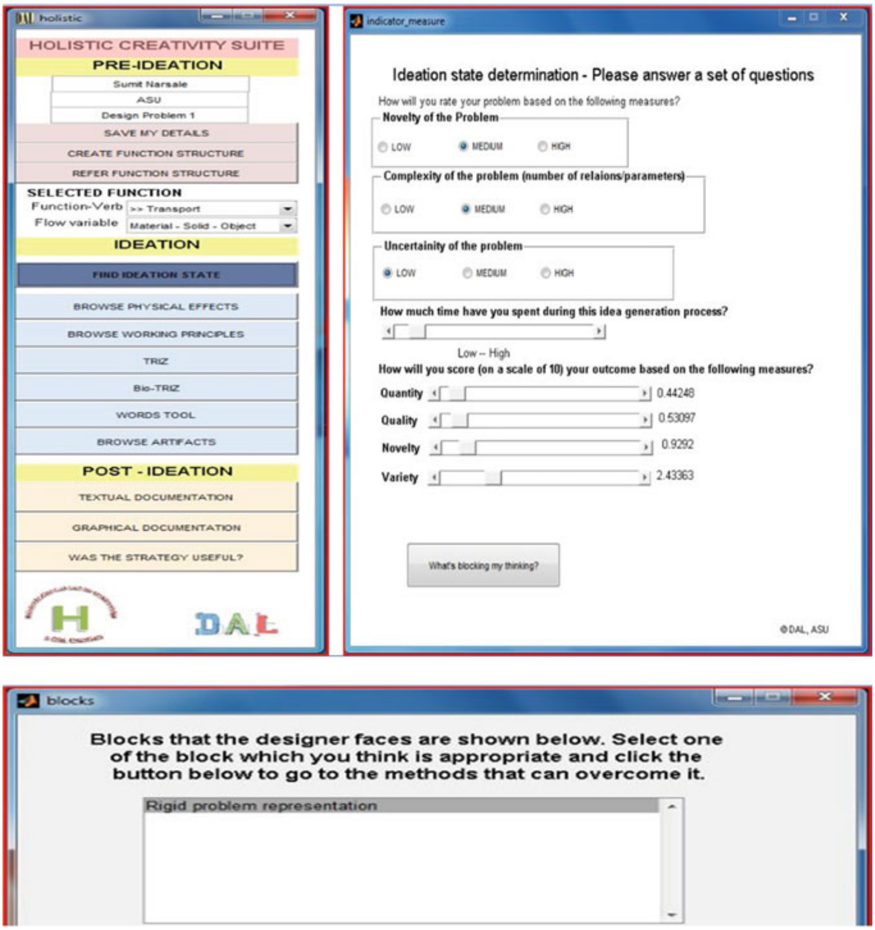


**Fig. 4** High level architecture of computer tool [15]

pick and proceed or choose a different tool. The tool instantly provides access to the chosen internal or external method selected.

Figure 6 represents one instance of the tool where the designer is using the Physical Effects method in order to find some solution where he can give input parameters depending upon his perception of the problem and can find related physical effects with description and respective formulae.

Figure 7 depicts the post ideation survey form which can be used to document various things such as how effective was the ideation method for various ideation states and their level of satisfaction with the current solutions.



Sumit Narsale
ASU
Design Problem 1

SELECTED FUNCTION
Function-Verb: >> Transport
Flow variable: Material - Solid - Object

Ideation state determination - Please answer a set of questions

How will you rate your problem based on the following measures?

Novelty of the Problem: ○ LOW ● MEDIUM ○ HIGH

Complexity of the problem (number of relaions/parameters): ○ LOW ● MEDIUM ○ HIGH

Uncertainity of the problem: ● LOW ○ MEDIUM ○ HIGH

How much time have you spent during this idea generation process?

Low -- High

How will you score (on a scale of 10) your outcome based on the following measures?

Quantity 0.44248
Quality 0.53097
Novelty 0.9292
Variety 2.43363

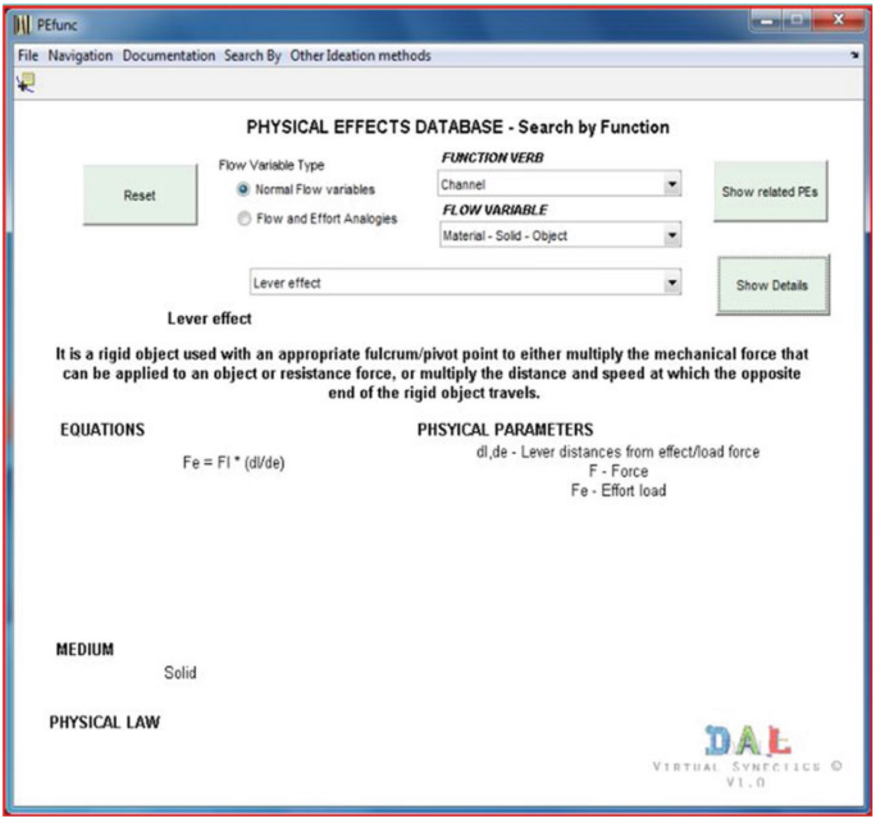
Blocks that the designer faces are shown below. Select one of the block which you think is appropriate and click the button below to go to the methods that can overcome it.

Rigid problem representation

Fig. 5 Ideation Tool GUI

## Ideation States

Ideation state means the current understanding of the design space and one's current location in that space. It tells about the current focus of the problem solver, the factors that are blocking his creativity, things that prevent him from exploring new paths and also about the reasons for not making progress. We characterize these ideation states with the use of a set of indicators. The indicators of ideation states are classified into Problem related (novelty, complexity—number of relations or variables involved and uncertainty), process related (solution time) and outcome based on evaluation of ideas (quantity, quality, novelty and variety). One



**PHYSICAL EFFECTS DATABASE - Search by Function**

Flow Variable Type
● Normal Flow variables
○ Flow and Effort Analogies

**FUNCTION VERB**
Channel

**FLOW VARIABLE**
Material - Solid - Object

Lever effect

**Lever effect**

**It is a rigid object used with an appropriate fulcrum/pivot point to either multiply the mechanical force that can be applied to an object or resistance force, or multiply the distance and speed at which the opposite end of the rigid object travels.**

**EQUATIONS**

Fe = Fl * (dl/de)

**PHSYICAL PARAMETERS**

dl,de - Lever distances from effect/load force
F - Force
Fe - Effort load

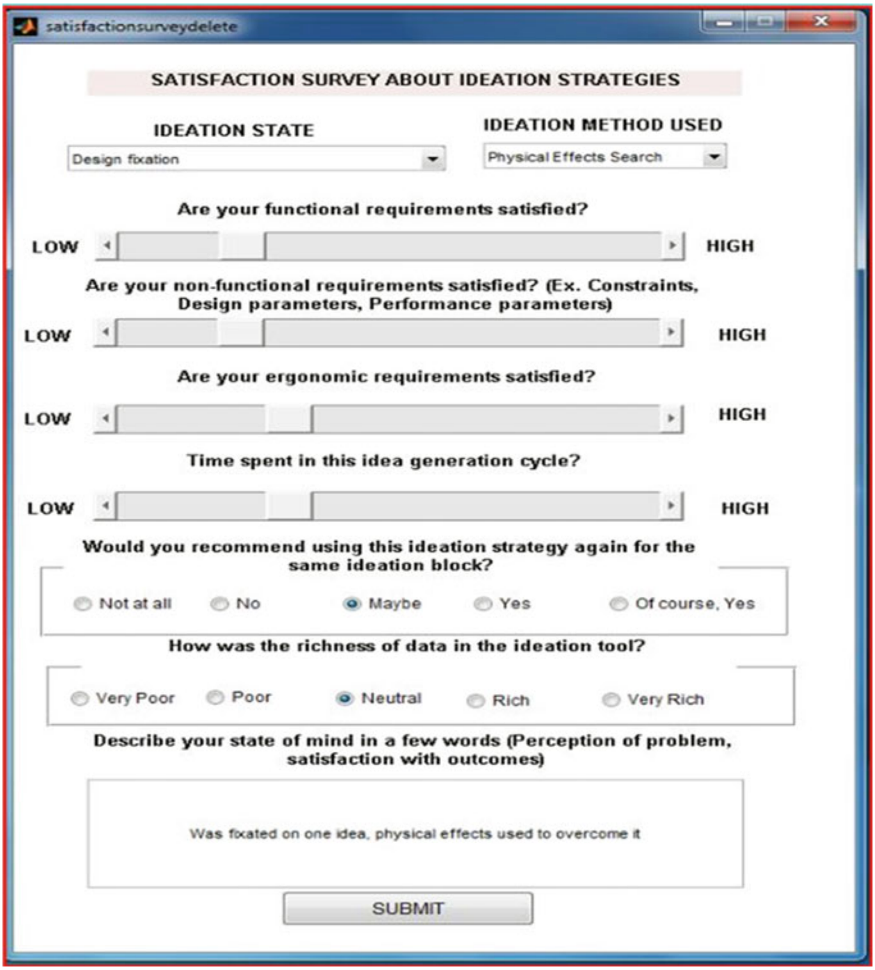
**MEDIUM**

Solid

**PHYSICAL LAW**

**Fig. 6** Application interface example: physical effects catalog

example of a blocking phenomenon is Design fixation, where the uncertainty of the problem is very less and the designer spends a lot of time in a problem and comes up with less number of ideas whose novelty and variety scores are very low. Another example is ‘Bias towards past design’ where the ideas generated are very similar to designs in the past. This phenomenon occurs mostly for less novel and more complex problems with time constraints. The overall quality may be high for those design but the novelty and variety scores are very low. Similarly, the measures for other blocking phenomena were developed. This process of finding the position of the designer in design space is called as Characterization of ideation state. This helps in finding the current focus of the problem solver, the factors that block his creativity, ways to overcome his block and make progress (also continue in the same path if the designer makes progress).



SATISFACTION SURVEY ABOUT IDEATION STRATEGIES

IDEATION STATE: Design fixation

IDEATION METHOD USED: Physical Effects Search

Are your functional requirements satisfied?
LOW HIGH

Are your non-functional requirements satisfied? (Ex. Constraints, Design parameters, Performance parameters)
LOW HIGH

Are your ergonomic requirements satisfied?
LOW HIGH

Time spent in this idea generation cycle?
LOW HIGH

Would you recommend using this ideation strategy again for the same ideation block?
○ Not at all ○ No ● Maybe ○ Yes ○ Of course, Yes

How was the richness of data in the ideation tool?
○ Very Poor ○ Poor ● Neutral ○ Rich ○ Very Rich

Describe your state of mind in a few words (Perception of problem, satisfaction with outcomes)
Was fixated on one idea, physical effects used to overcome it

**Fig. 7** Post-ideation survey

## Example Case Studies

To illustrate the differences in ideation paths taken by different designers, a few short case studies were done on the holistic ideation framework. This is not meant to be a comprehensive study and we make no claims of having found patterns of ideation that could be generalized. This is just to illustrate that the tool we have created could be used as a research test bed in future studies.

The problem posed was to design a “portable power ramp” which can transfer objects from top to bottom or from bottom to top of any vehicle (cars or trucks or buses) that can be powered by any source of choice. It is required to be lightweight,

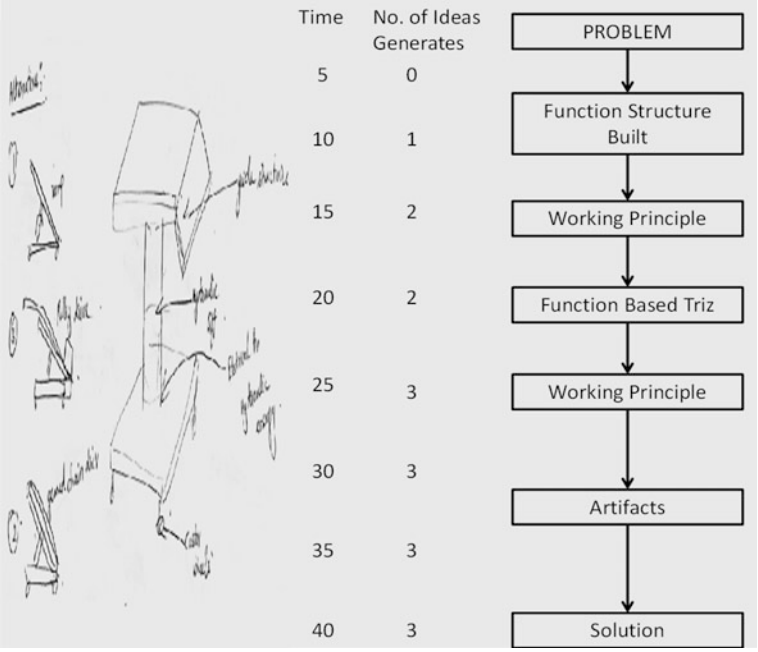


**Fig. 8** Ideas generated (*left side*) and ideation path (*right side*) of Designer 1 (Total number of ideas generated = 3), Ideation path here indicates the modules of the software that were used

easy to carry, power-efficient, and safe. Example ideation paths of three different designers are shown in Figs. 8, 9 and 10. A large variety is observed in ideas generated by different designers following different paths. While the first designer generated fewer good ideas, the second designer struggled to produce an idea that was feasible (stressed quality). The third designer adopted a different path and was able to generate a large number of alternatives.

Designer 1 used Working principles mostly and also used function based TRIZ and came up with higher quality ideas than Designer 2. He did not find TRIZ much helpful as he was concerned with generating ideas that are feasible; thus he decided to go back to Working principles method. He spent nearly half of his time searching the Working principles database and created half of his concepts using it. Even though the overall structure of all his three concepts were nearly the same and it seemed that the Working principles database was unable to inspire him to overcome fixation, by proposing alternative lowly related working principles as well as highly related ones, he was guided to change minor details of his concepts. Artifact's database was then chosen in order to search for creative concepts. Initially, the huge number of the components available in the database seemed very confusing to this designer; however in the end, even the irrelevant ones were very intuitive to him. He said that components' database reminded him of practical mechanisms and resulted in generation of concepts that are very likely to work.

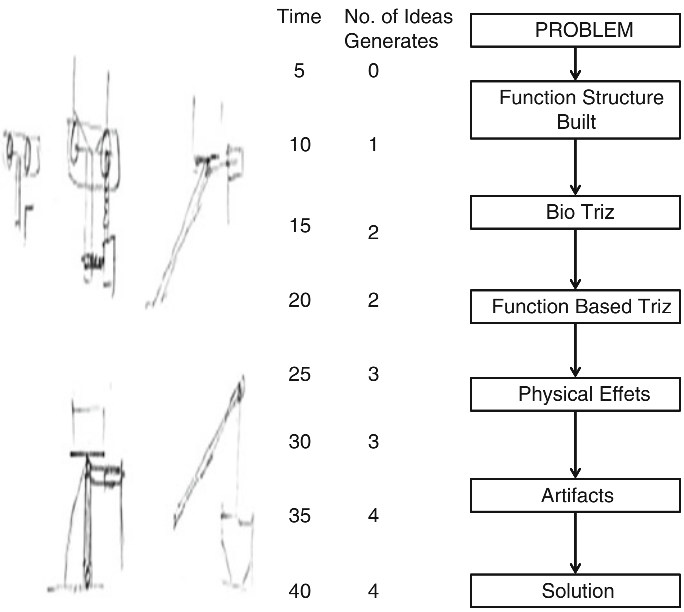


**Fig. 9** Ideas generated and Ideation path of Designer 2 (Total number of ideas generated = 4)

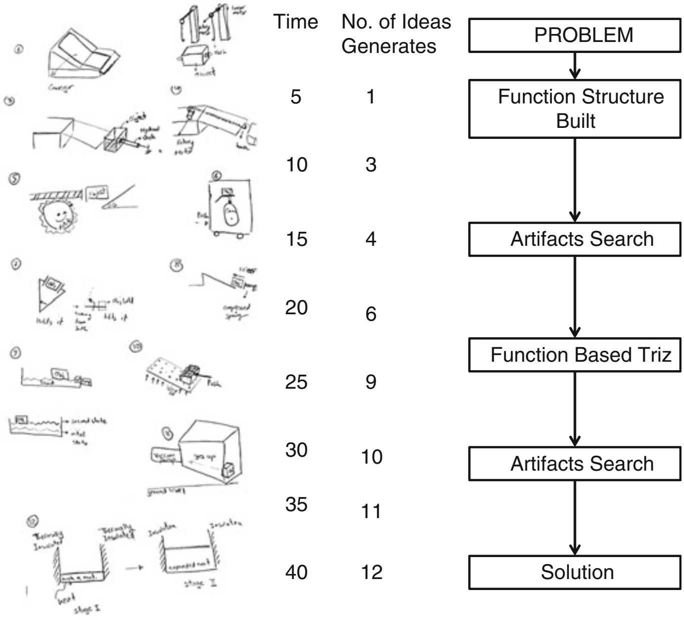


**Fig. 10** Ideas generated and Ideation path of Designer 3 (Total number of ideas generated = 12)

The most important factor for this designer was the quality [19] of the concepts; compromising to generate wild ideas for the sake of novelty—but not feasibility—was the main reason the Bio-TRIZ data base did not look much useful to him. Yet with its comprehensive information of so many mechanisms, the Artifacts database enabled him to defeat fixation.

Designer 2 started with Bio-TRIZ as it satisfied his concern for generating novel concepts. Natural solutions proposed by the data base for the natural phenomenon of "transportation", inspired this designer to generate concepts that incorporated fluids as a means of transportation. The Bio-TRIZ database enabled him to generate nearly half of his alternatives using the basic concept of transporting goods via fluids. Next, Physical effects database was chosen. Presence of numerous physical effects in the data base was so puzzling to this designer that after some time, he gave up searching the data base. According to him, the suggestions made by this module were very far from being a concept and needed much time for refinement to become practical. For the very same reason, TRIZ was not very inspirational to him either; thus he decided to use the Working principles module in order to view solution methods that are closer to being a concept than an example of a physical phenomenon. In this sense, Working principles met the designer's requirements and he spent half of his time searching its data base. Not only approved by this designer, but by all the designers who participated in this survey, the Working principles data base is very efficient with increasing creativity and fighting fixation by providing a practical link between the physical rules and working instances of the mechanisms. Extremely helpful to Designer 2 in this sense, he allocated half of his time searching its data base and ended up generating half of his concepts with its help. The search in the Artifacts database did not last for long as he did not find this module as inspirational as the Working principles.

Designer 2 was very satisfied using the software; he said that compared to his usual hard time generating very few concepts, this time with help from the software, he ended up generating more concepts. Bio-TRIZ and the Working principles were used most by him, yet in the end, he emphasized on the importance of availability of other methods as well. According to him, this capability enabled him to fight fixation and increase the novelty of his concepts.

A different path from his counterparts was incorporated by Designer 3. The Artifacts and Function based TRIZ data base were the methods of ideation used most by him. Unlike his counterparts who found re-wording very confusing, he worked most with it and tried searching the databases by using alternative functions. Enabling him to successfully expand the design space and generating twelve concepts among which patterns of fixation are not observed, he attributes his success to the availability of different methods in the data base which nurtured his imagination.

He spent most of his time skimming through the Artifacts database; with trying alternative words and looking at even unrelated artifacts, he managed to generate as many concepts as Designer 2 did in half the time. Even though he did not use Bio-TRIZ much (as he found it not much efficient), he nearly spent the entire second half of his time searching the function-based TRIZ. His main reason for not using the Bio-TRIZ was that he found connecting pure ideas received from the Bio-TRIZ a slow process requiring lots of active and passive time for refining the results; however according to him, for the Artifacts database, solutions are already there and not much effort is required for modifying them.

Overall, it could be stated that the high quality of the concepts generated by Designer 1, the novelty of the concepts generated by Designer 2 and the quantity of the concepts generated by Designer 3 as well as their variety indicate the effectiveness of the ideation paths that this software provides. Patterns of fixation are rarely observed in the generated concepts and as obvious from the results, the software is capable of meeting each designer's requirements individually.

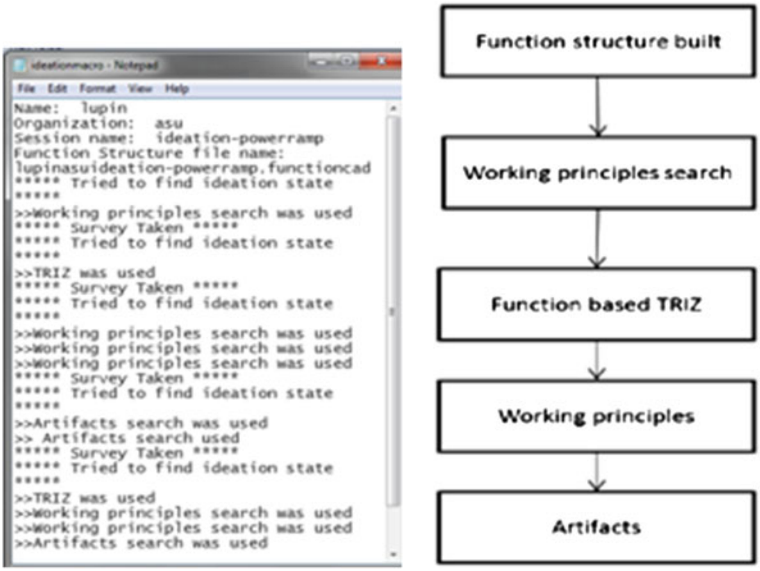
An inspiring observation made using this ideation method was that even with using the very same tool and following the very same path, two individual designers did not come up with similar concepts. Actually even though a logical ideation method was used, no trend was found between the concepts generated and the material viewed. This provides proof that use of logical ideation methods would not impose any hindrance over the creative ideation, yet it would provide an appropriate bed for generation of a wide range of concepts having high novelty, quality, quantity and variety [19].

In this small set of designer studies, we found that even though the studied designers had nearly the same amount of experience in design (all graduate students majoring in Mechanical Engineering, with specialty on Design with little industrial experience[1]), each found a different ideation path most helpful. The right sequencing of the ideation paths that they proposed after completion of the task, were also different. What they all agreed upon was the importance of pictures and videos in generating innovative concepts. And according to them, pictures were much more helpful than words in formation of new concepts. Even though they all tried to somehow look at synonymous words to modify the function structure, except Designer 3, rewording was not much helpful to others and they were unable to expand the design as much as the second designer did.

Overall, viewing the results obtained by this study, it seems that the ideation process is subjective as well as problem-dependent. Results of this study support the fact that logical ideation methods do not hinder the creative ideation process; in contrast, they encourage it. As approved by all the designers who participated in this study, the Holistic approach was found to be very helpful for persuading generation of novel concepts and providing basis for generation of a vast and creative set of ideas.

---

[1] All the designers that participated in this study received basic trainings on how to use the software; but were not guided during the study. Because of their major, they were already familiar with ideation techniques but had not prior experience using them.



```
Name:   lupin
Organization:   asu
Session name:   ideation-powerramp
Function Structure file name:
lupinasuideation-powerramp.functioncad
***** Tried to find ideation state
*****
>>Working principles search was used
***** Survey Taken *****
***** Tried to find ideation state
*****
>>TRIZ was used
***** Survey Taken *****
***** Tried to find ideation state
*****
>>Working principles search was used
>>Working principles search was used
>>Working principles search was used
***** Survey Taken *****
***** Tried to find ideation state
*****
>>Artifacts search was used
>> Artifacts search used
***** Survey Taken *****
***** Tried to find ideation state
*****
>>TRIZ was used
>>Working principles search was used
>>Working principles search was used
>>Artifacts search was used
```

**Fig. 11** Macro from Holistic Ideation represented as Ideation path

## Conclusion and Future Work

This paper discussed the importance of ideation paths and the approaches that can be used to analyze and track them. Ideation is not a monolithic process, and the designer needs different strategies at different instances as his ideation state changes. This implies that emphasis on study of single ideation strategy/method is insufficient to explore ideation paths. This paper opens up a new area of research which highlights the importance of varying ideation paths. Example case studies were presented which elucidated the changes in outcomes when the designer followed different paths. It is observed that with change in ideation path, the effectiveness measures [19], which include quantity, quality, novelty and variety, are hugely affected.

One problem in research in design thinking particularly using protocol studies is that it is impossible to collect and analyze massive amounts of data. Perhaps this difficulty can be overcome by collecting data in structured form using computer tools. Figure 11 shows the data collected by a macro that has been implemented as part of the computer tool.

It is recommended that future experiments may be done for longer durations with more variety in ideation strategies, with method variables, human variables and environment variables as the factors for factorial analysis, and the effectiveness measures as resultant measure.

**Acknowledgments** The authors gratefully acknowledge the support of the National Science Foundation Grant #1045644. Opinions expressed in this paper are of the authors and not endorsed by NSF.

## References

1. Shah JJ (1998) Experimental investigation of progressive idea generation techniques in engineering design. In: DETC-DTM conference proceeding
2. Roth K (2000) Konstruieren, Mit Konstruktionskatalogen:Band 1: Konstruktionslehre. Springer, Berlin
3. VDI-Fachbereich Produktentwicklung und Mechatronik, Konstruktionsmethodik/ 1 VDI-Handbuch Konstruktion. Methodisches Entwickeln von Losungsprinzipien 2222(1)
4. AskNature Bio-Mimicry Design Portal. http://www.asknature.org/
5. Stone RB (nd) Design Engineering Lab. Repository, Oregon State University: http://designengineeringlab.org/
6. Altshuller G (2001) 40 principles: TRIZ keys to technical innovation, Tech Innov Center
7. Zwicky P (1969) Discovery, invention, research through morphological analysis. McMillan, New York
8. Shah JJ, Vargas-Hernandez N, Summers JD, Kulkarni S (2001) Evaluation of collaborative sketching as an idea generation technique for engineering design. J Creative Behav 35(3):1–31
9. Rohrbach G (1969) Kreativ Nach Regeln—Method 635, Eine New Technik Zim Lösen von Problemen. Absatzwirtschaft 12
10. Hellfritz H (1978) Innovation via Gallerie Methode. Eigenverlag, Königstein/Ts
11. Vargas-Hernandez N, Shah JJ, Smith S (2010) Understanding design ideation mechanisms through multi-level aligned empirical studies. Des Stud 31(4):382–410
12. Hyunsoo L, Jonghyum K, Amarnath B (2010) Collaborative intelligent CAD framework incorporating design history tracking algorithm. Comput Aided Des 42(12):1125–1142
13. Fruchter R, Demian P (2002) Knowledge management for reuse. In: Proceedings of CIB W78 conference
14. Khadilkar DV, Stauffer LA (1996) An experimental evaluation of design information reuse during conceptual design. J Eng Des 7(4)
15. Mohan M, Chen Y, Shah JJ (2011) Towards a framework for holistic ideation in conceptual design. In: Proceedings of IDETC 2011
16. Reed KR (2011) Protocol analysis. Westnet learning
17. Yin RK (2009) Case study research: design and methods. Sage publications, Beverley Hills
18. Shah JJ, Smith SM, Hernandez NV, Gerkens DR, Wulan M (2003) Empirical studies of design ideation: alignment of design experiments with lab experiments. In: Proceeding of IDETC 2003
19. Shah JJ, Smith SM, Vargas-Hernandez N (2003) Metrics for measuring ideation effectiveness. Des Stud 24(2):111–134

# Field Based Behavior Regulation for Self-organization in Cellular Systems

**Yan Jin and Chang Chen**

**Abstract** A *cellular self-organizing* (CSO) approach is proposed to develop adaptive systems. The design of CSO systems however is difficult because the global effect emerges from local actions and interactions that can be specified and controlled. To achieve high level adaptability of CSO systems and acquire the capability of specifying desired global effects, we propose a *field based* regulative control mechanism, called *Field based Behavior Regulation* or FBR. FBR is a real-time, dynamical, distributed mechanism that allows CSO system cells to self-organize in complex operation environments. This paper describes the models of CSO and FBR and demonstrates their effectiveness through simulation based case studies.

## Introduction

Complexity has been recognized as an important feature and mechanism of bio-systems, societal systems and technology development processes. However, for engineered systems, the notion of complexity often points to unintended, undesirable and must-avoid system properties.

One may note that the increasing complexity of engineered systems comes from increasingly complex and highly sophisticated functional requirements. A major issue with developing complex engineered systems is that the sheer number of and sophisticated interdependencies among the system components imply *uncertainty* and *unknowns* to the engineers, making it difficult for them to ensure the valid operation range for the system to function and survive.

---

Y. Jin (✉) · C. Chen
University of Southern California, Los Angeles, CA, USA
e-mail: yjin@usc.edu

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_33,

On the other hand, it is intriguing to consider that the nature faces all the uncertainties and unknowns and yet the natural systems are "designed" to live with these uncertainties and unknowns as an inherent part of their capabilities. We observe that human design and natural "design" are very distinct from each other: human design is more purpose or function driven and takes a *top–down* approach to avoid possible complexity problems, while the nature "design" is arguably less purposeful and follows a *bottom–up* approach by making complexity as a "solution" to deal with the rising uncertainties and unknowns [1]. The research on system biology [2], self-configurable systems [3] and component-based design [4] has explored the formation of adaptive systems from both natural and man-made perspectives. In our research, we introduce a *cellular and self-organizing* (CSO) approach to building adaptive systems. In this approach, a system is composed of multiple cells, which can be either identical (for homogeneous systems) or distinct (for heterogeneous systems). Further, the formation of such systems is based on a set of bottom–up, dynamical self-organizing mechanisms. It is fully understood that such a CSO approach will not be able to compete with the traditional methods in a short term and for many applications. However, the paradigm-shift from *component-based* to *cell-based* and from *top–down* to *bottom–up* promises an alternative future for developing complex engineered systems.

Both *multi-agent* systems and *self-organizing* systems have been highlighted in many engineering fields, such as computer science, industrial engineering, and material science. Much research has been done to investigate the properties and benefits of such systems, and the ways to build them. One critical design research question is: *How can a designer connect the design of cells/agents, their interactions with functional requirements (or tasks)?* In our research, we propose a *field based behavior regulation* (FBR) approach as a basis for cells to interact with each other and with their task environment. In this approach, a cell is a *sensor-operator* unit that can perform a range of actions or behaviors. At a given time, a cell's behavior can be self-regulated based on the cell's "*field position*" at that moment. The field position of a cell is determined by the task requirements and the environmental situation that are sensible by the cell. When a CSO system is composed of multiple cells, at any given time, different cells may perform different or similar behaviors. This *cellular differentiation* is achieved locally through field based regulation, unlike conventional modular [5, 6] or component-based approaches in which differentiation of components is determined at design time and does not change during system operation.

In the rest of this paper, we first review the related work in Sect. "Related Work" and then introduce our cellular self-organizing (CSO) framework in Sect. "CSO: A Naturalistic Approach to Design". In Sect. "The Model and Concepts", we present field based behavior regulation (FBR) approach, and in Sect. "Field driven Behavior Regulation" we demonstrate the effectiveness of our approach through simulation based case studies. Section "Case Studies and Discussion" draws conclusions and points to future research directions.

## Related Work

Much research has been done to investigate multi-agent and self-organizing systems and to develop methods for designing such systems. Self-organization and emergent behavior as two major features of such systems have been popular research topics in the research field of complex systems [7–11]. Self-organization is the large scale organization through the limited local interactions of the constituent components. Emergence represents the concept of the patterns, often unpredictable ones, which are exhibited in the large scale organization. Holland [12] and Gell-Mann [13] extended the research to non-homogeneous system and pointed out the non-linearity between local and global which becomes the biggest challenge of such systems. To further address the problem, the Game of Life [14] and more Cellular Automata based fractals have been explored [10].

In the field of engineering design, design for adaptability and design of reconfigurable systems have been investigated in the past decade. In their work focusing on vehicle design, Ferguson and Lewis [15] introduced a method of designing effective reconfigurable systems that focuses on determining how the design variables of a system change, as well as investigating the stability of a reconfigurable system through the application of a state-feedback controller. This method is based on multi-objective optimization and allows systems to adjust their design variable by dynamically optimizing in response to changing conditions. The adaptability of such systems is limited by the range of change of the variables and by the pre-conceivable changing situations. Martin and Ishii [16] proposed a design for variety (DFV) approach that allows quick reconfiguration of products but mainly aims to reduce time to market by addressing generational product variation. Indices have been developed for generational variance to help designers reduce the development time of future evolutionary products [16]. In addition to developing design methods for reconfigurable systems, various reconfigurable robotics have been developed mostly by computer scientists. Fukuda and Nakagawa [17] developed a dynamically reconfigurable robotic system known as DRRS. Unsal et al. [18] focused on creating very simplistic i-Cube systems (with cubes being able to attached to each other) in order to investigate whether they can fully realize the full potential of this class of systems. PolyBot has gone through several updates over the years [19] but acquired notoriety by being the first robot that “demonstrated sequentially two topologically distinct locomotion modes by self-configuration. SuperBot [20] is composed of a series of homogeneous modules each of which has three joints and three points of connection. Control of SuperBot is naturally inspired and achieved through what the authors describe as the “hormone” control algorithm [21].

Bio-mimetic design methods allow designers to identify appropriate natural systems or mechanisms from which to draw design inspirations. The idea of using DNA and genes to capture genotype of systems is not new. Inspired by the nature’s evolution process, genetic algorithm (GA) [22] and genetic programming (GP) [23] have been established to model problems using bit string (GA) or functional tree (GP) genes and to solve problems by evolving the best solution(s) from a population through reproduction, mutation, recombination, natural selection and survival of fitness. This approach has been taken to solve various engineering problems including design optimization, configuration design, and design automation [24–27].

Our previous work on CSO generated a design DNA concept and associated system formation mechanisms [28, 29]. This research extends the previous research by first expanding the concept of design DNA from a static specification to a dynamic and probabilistic representation and then introducing a new *field based control mechanism* to utilize the potentials of such systems for increased robustness and resilience. Our field based behavior regulation approach allows agents to respond to the field of the task environment *spontaneously* and interact with other cells or agents only implicitly, as a result of their actions in the task field.

## CSO: A Naturalistic Approach to Design

The goals of our research on CSO systems are two-fold. First, we aim to develop systems that are *flexible* in responding to various known or unknown tasks, *robust* in achieving given tasks under changing environment situations, and *resilient* in dealing with partial system failures. Secondly, we are interested in understanding how nature does "design" and developing a similar *bottom–up* and *self-organizing* based design method for future complex engineered systems, i.e., we intent to *design the design* that our CSO systems can carry out by themselves through self-organizing.

Before introducing the fundamental concepts of our proposed CSO systems, we first compare engineered systems with natural systems from a *design* perspective. As shown in Fig. 1, in this comparison, we divide the natural systems into two categories, dynamical systems (e.g., planetary system) and biological systems (e.g., animals and plants), and the "design perspective" is captured by dividing analysis into four levels:

- *Physical substrate*: the physical units that constitute the system;
- *Mechanism*: the ways by which the system attain its behavior;
- *Capability*: the manifestation of external effects of the system, desired or not; and
- *Adaptation*: the ways by which the system changes itself.

As shown in Fig. 1, conventional engineered systems is designed and built based on physical components that can be structured in various ways. The mechanism is realized by organizing the behaviors of the components. The desired functions are achieved through working mechanisms of the organized component behaviors. These systems cannot change themselves in any explicit or implicit way

| **Levels** ⇩ | **Engineered Systems** | **Natural Systems** | |
|---|---|---|---|
| | | **Dynamical Systems** | **Biological Systems** |
| **Adaptation** | N/A | Strange attractors (?) | Genetic evolution: Natural selection |
| **Capability** | Function: Constrained actions | Dynamics: Attractors and stability | Survival: Live and reproduce |
| **Mechanism** | Organization of Behaviors | Self-organization | DNA guided self-organization |
| **Physical Substrate** | Components | Objects (e.g., planets, particles) | Cells |

**Fig. 1** Comparison of engineered systems and natural systems

in response to the changes of operation environment, meaning that unconventional ways to design systems is needed to achieve system adaptability.

Natural dynamical systems are formed based on objects such as planets or particles. Their mechanisms are completely self-organized based on the relations, such as gravity, between the objects. While natural dynamical systems do not perform their “functions” per se, they do exhibit their “capabilities” by reaching their attractors and maintaining stability around these attractors through spontaneous processes of the individual objects. Furthermore, the chaotic attractors of dynamical systems can be considered as the mechanism that increases the variety of the stable states of the system, hence the richness of strange attractors can be considered as a feature of adaptability. On the other hand, one may also consider the landscape of all attractors of a dynamical system is determined when the system is formed.

Common to all biological systems, *cellular* structure is indispensable for these systems to grow into complex configuration. Unlike dynamical systems where no memory or shared information is present from an object’s point of view, each cell in a biological system possesses a “description” called DNA, of the whole system and is able to *interpret* this locally shared information to generate local actions, i.e., producing adequate proteins. The self-organizing behavior is still spontaneous but guided by DNA. The separation of description of the system from the system itself makes it possible to “copy” and “vary” the description independently from changing the system. Therefore, mutation and natural selection together create an evolution framework for *open-ended* adaptation.

Learning from what nature “does” has led us to treat *self-organizing* as the key concept for adaptive engineered system. Self-organizing has profound implications in dealing with complexity. First, it is spontaneous hence does not require prespecifying “who should do what in what ways,” allowing high level uncertainty under unpredictable situations. Secondly, if arranged properly, increasing system complexity can be a solution to dealing with high level environment complexity.

| **Levels** ⇩ | **Cellular Self-Organizing (CSO) Systems** |
|---|---|
| **Adaptation** | Distributed and design-DNA based evolution |
| **Capability** | Field based "attractors" |
| **Mechanism** | dDNA guided and field based self-organizing |
| **Physical Substrated** | Mechanical Cells, Fields, |

**Fig. 2** The cellular self-organizing systems framework

The challenge, however, is how we can devise and guide self-organizing so that desired system level emergent behaviors and functions can be achieved.

In our proposed Cellular Self-Organizing (CSO) systems framework, shown in Fig. 2, three concepts are fundamental, namely, *mechanical cells*, *fields*, and *design-DNA*. Mechanical cells constitute the physical substrate for system formation. The concept of *field* is needed to bring tasks and environmental constraints into the cells' self-organizing framework. Like dynamical systems, where gravity fields are basic means for planets to self-organize, we need some fields in which our cells can self-organize. Since our *fields* are artifacts to be designed, the self-organizing behavior of our mechanical cells must be guided. The details of these concepts together with the elaborations are described in the next section.

## The Model and Concepts

In this section, we elaborated the discussion of the last section by introducing definitions of the concepts that were introduced. Through the process of describing definitions, we also introduce the field based mechanisms that are needed to realize self-organization.

**Definition 1** (*Mechanical Cell*) mCell = $\{Cu, S, A, B\}$; where $Cu$: control unit; $S = \{s_1, s_2, \ldots\}$: sensors/sensory information; $A = \{a_1, a_2, \ldots\}$: actuators/actions; $B$: designed behavior, or design information (see definition 4 below).

*Mechanical Cell* is the smallest structural and functional unit of a CSO system. Although for a CSO system design, the appearance or the structure of its *mCells* may be different, a *mCell* should be able to sense the environment and process material, energy and/or information as their actions.

**Definition 2** (*State*) State = $\{S_C, A_C\}$ where $S_C \subset S$and $A_C \subset A$ are currently sensory information and actions, respectively.

State is used to represent the situation. It is the combination of the current sensor information $Sc$ and current actions $Ac$.

**Definition 3** (*Behavior*) $b = \{S_E, A_E\} \rightarrow A_N$ where $S_E \subset S$and $A_E \subset A$ are existing sensor information and actions, respectively; and $A_N \subset A$ are next step actions.

A behavior *b* is the designed action for given situations or states. The *Cu* of the *mCell* should be able to judge the situation and make decisions on next actions. The design information of a CSO system is the fully developed behaviors for each *mCell*.

**Definition 4** (*Behaviors of System*) $BoS = \{B_1, B_2, \ldots, B_n\}$; where $B_1, B_2, \ldots, B_n$ are behavior sets of each *mCell* in the system.

The design information for such a system is the set of all the behaviors which local *mCells* should follow; also this *BoS* is supposed to be designed by a designer or designers. If all *mCells* share the same behavior set *B*, i.e., $B_1 = B_2 = \ldots = B_n = B$, then we have a homogeneous CSO system. Otherwise, the CSO system is said to be heterogeneous.

**Definition 5** (*Functional Requirement*) $FR_i = \{S_i, A_i\}$ where $S_i$, and $A_i$ form a specific state or situation.

There are two reasons why the *functional requirement* holds similar construct of *state* described above. First, this representation allows us to specify "desired states" of the system. Second, using the state construct to represent functional requirements allows *mCells* to recognize whether the function is achieved.

At present we explore CSO systems with homogeneous *mCells*. Following the stem cell analogy, the initial homogeneous *mCells* with multiple behavioral capabilities will differentiate and find their "specialty" behaviors during the period of task execution. Eventually this *self-organized* emergence may create functional blocks consisting of multiple *mCells*, as organs forming in biological systems or attractors in dynamical systems.

The high level self-organizing and redundancy ensure the flexibility, robustness and resilience of the system. From a design perspective, however, developing CSO systems means to develop mechanisms that lead to fruitful *emergence*. Two fundamental issues must be addressed. The first relates to design information representation. The second issue is to devise mechanisms to guide self-organization. In the following, we introduce a *field* based approach to allow mCells to self-regulate their behaviors.

## Field Driven Behavior Regulation

In dynamical systems, the concept of field is everywhere, e.g., gravity field, electrical field, magnetic field and electromagnetic field. Objects operating in the fields can "sense" the field and react to it by following physical principles. In the biological world, the function of an organism is realized by a collection of different types of cells working together. The distribution of the chemical signals, called

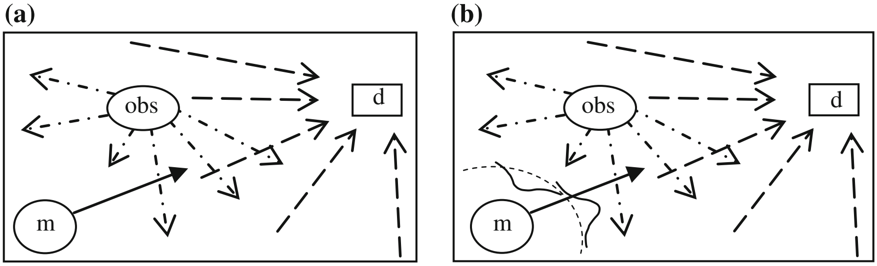


**Fig. 3** An example of task field and behavior field **a** m moves to d in tField , and **b** m moves to d in bField

*morphogen*, controls the biological regulation hence the shape and organ formation. Through the developmental process stem cells differentiate into different cell types by responding to specific *morphogen* distribution.

In our CSO systems, *mCells* need the similar differentiation capability in order to self-organize and collectively become a functional system. *mCells* produce different actions (instead of *proteins*) and be triggered by functional and environmental signals (instead of chemical signals). To realize such behaviors, we introduce a *field driven behavior regulation* (FBR) approach to guide cellular self-organization.

For a CSO system, the sensory capabilities of its *mCells* are pre-defined and given. In this case, we can define a *task field* which captures the external world to a *mCell* encompassing both task requirements and environmental conditions. We have:

**Definition 6** (*Sensory Info and Sensing*) *sInfo* = *SNS* (*FR, Env*) where, FR: functional requirements; Env: environment situation; SNS: sensory operator.

**Definition 7** (*Task Field and Field Formation*): *tField* = *FLD* (*sInfo*) where, sInfo = ($s_1$, … $s_n$): sensory information; FLD: filed formation operator.

From Definition 7 it can be seen that we define the task field relative to a specific *mCell* and its sensing capability. Figure 3a shows a simple example of *tField.* A mCell *m* is moving to its destination *d* with the potential of encountering an obstacle *obs*. In this case, the destination *d* can be considered as an attractor that creates an *attraction field,* capturing the task requirements; and the obstacle, *obs*, creates a *repelling field,* characterizing the environment. It can be seen from Fig. 3a that the *tField* serves as a "complete" context for a *mCell* to operate in this specific example.

Since *mCell* differentiation is about behavior distribution, a *mCell* must be able to determine its behavior based on the given *task field*. Therefore, we introduce a concept called *behavior field*, or *bField*, to capture the potential distribution of preferable behaviors a *mCell* can choose in a given *task field.* We further use $FBR_{FD}$ to denote the transformation from a *task field* into a *behavior field* and introduce the following definition:

**Definition 8a** (*Behavior Field and Field Transformation*) $bField = FBR_{FT}(tField, B)$ where, $FBR_{FT}$: field based regulation (FBR) operator for field transformation; *bField*: behavior field; *tField*: task field.

According to Definition 8, how behaviors should be distributed is largely dependent on the field transformation operator $FBR_{FT}$. There can be different ways to represent *bField*. One may associate "rewards", "risks", or "probability" with different "locations" for a *mCell* to perform different behaviors. The "locations" can be defined as real 2- or 3-dimension spaces or n-dimension virtual spaces depending on the task domain and *mCell* properties. Figure 3b shows a simple example of *bField*. A *mCell m* is moving in the *task field*. Based on some given field transformation operator, $FBR_{FT}$, the *mCell m* creates a *bField* around itself denoted by the curved dark line around *m*.

In this research, we associate a *mCell*'s "behavior distribution" with its surrounding "locations", and we further call this distribution *behavior profile*, or *bProfile*, which is a specific case of Definition 8a.

**Definition 8b** (*Behavior Profile and Field Transformation*) $bProfile = FBR_{FT}(tField, B)$ where: $bProfile := \{(b_1, p_1), \ldots, (b_n, p_n)\}$; and $[b_i \in B,\ 0 \le p_i \le 1,\ 1 \le i \le n]$ indicates (*behavior, probability*) pairs for a *mCell* to choose its actions, *n* is the total number of possible behaviors mCell can perform; *tField*: task field; *B*: mCell's behavior set.

The dark line in Fig. 3b indicates the "behavior profile" for *mCell m*. Given a behavior profile at a given time, a *mCell* still need to "make a decision" to select a behavior. We introduce the second field based behavior regulation operator as follows.

**Definition 9** (*Behavior Selection & Behavior Selection*) $b = FBR_{BS}(bProfile)$ where: $FBR_{BS}$: field based regulation (FBR) operator for behavior selection; $bProfile := \{(b_1,p_1),\ldots,(b_n,p_n)\};\&[b_i \in B, 0 \le p_i \le 1, 1 \le i \le n]$;*b*:selected behavior $b \in B$.

Summarizing the above definitions, for a mCell *m* at time *t* under given functional requirements *FR* and environmental situation *Env*, the behavior or action of the mCell is chosen by following self-organizing operations:

$$b_{t+1} = FBR_{BS}(FBR_{FT}(FLD(SNS(FR_t, Env_t)))) \tag{1}$$

From Eq. (1), one can see that a mCell senses the world (*SNS*), creates an information field (*FLD*), transforms it into behavior field ($FBR_{FT}$), and selects its behavior ($FBR_{BS}$), forming a complete spontaneous behavioral cycle. The system stabilities and functions are achieved around the "attractors" of the mCells. For different task domains, these capabilities should be designed and devised differently so that the overall performance of the emergent behavior is desirable. In the following, we describe examples of how these self-organizing capabilities can be designed and implemented in computer simulations.

## Case Studies and Discussion

To investigate how such our approach can be applied to CSO systems design, a set of computer simulation based case studies were performed. In the following subsections, we present two case studies. The first case study investigates the concept of *field* and the second shows FBR effectiveness.

### *Case Study 1: Single Exploration Cell*

The overall task for this case study is for one *mCell* to travel to a given destination in an unknown environment. Two functional requirements are:

FR1 = "move to destination", and FR2 = "avoid obstacle".

The *mCell* can decide the *direction* of movement, so two behaviors are:

$b_1$ = "move to the direction toward destination", and
$b_2$ = "move away from the direction to obstacle".

We further assume that the obstacles between the *mCell* and the destination can be everywhere with any density and that the *mCell* can always sense the location of the destination and can sense the locations of the obstacles only when they are within a certain range.

#### Task Field

The task field here is composed of the *attraction field* of the destination and the *repelling fields* of various obstacles. We use parameter $\theta$ to represent the *attraction field* and $\beta$ the *repelling field*, as show, in Fig. 3. Combining the two, we have task field for *mCell m*:

$tFiled_m = \{\theta; \beta_1, \beta_2, \ldots, \beta_n\}$; where, $n$ = no. of obstacles

#### Behavior Regulation

As described above, in CSO systems field-driven behavior regulation has two steps, i.e.,

Step 1: Transform *tField* into *bFiled* through $FBR_{FT}$
Step 1: Select a specific behavior/action through $FBR_{BS.}$

**Behavior field and $FBR_{FT}$**: In this example, the *bField* or *bProfile* determines the likelihood in which a *mCell* is taking its next move into direction $\alpha$, and the likelihood the *mCell* is avoiding the directions to obstacles. The distribution of these two likelihoods around the 360° circle of a *mCell* constitutes the *bField* or

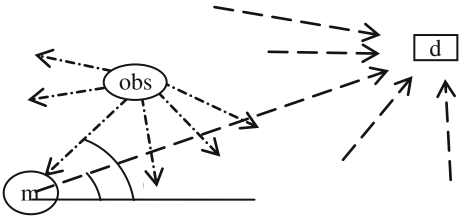


**Fig. 4** Tasks field for *mCell m*

*bProfile* of the *mCell*. Specifically, for one destination and one obstacle, we introduce the following $FBR_{FD}$:

$$\begin{aligned} bField_m(\alpha) &= FRB_{FT}(tField_m, B) = \{\alpha, p_\alpha, q_\alpha\} \\ &= \left\{\alpha, \frac{1}{\sqrt{2\pi}} e^{-\frac{(\alpha-\theta)^2}{2}}, \frac{1}{\sqrt{2\pi}}\left(1 - e^{-\frac{(\alpha-\beta)^2}{2}}\right)\right\} \end{aligned}$$

where,

$\alpha$: direction for the next move
$p_\alpha$: probability that direction $\alpha$ should be taken
$q_\alpha$: probability that direction $\alpha$ should be avoided

**Behavior selection and $FBR_{BS}$**: After the behavior field is established, a *mCell* needs a mechanism for behavior selection. In this case study, we define two types of behavior selections: "select the best" and "select any one good enough", as indicated below.

$FBR_{BS\text{-}B}$ = [Select the action with the highest probability in the *bField*]
$FBR_{BS\text{-}G}$ = [Select any action, randomly from the actions that has a bigger than threshold probability in the *bField*]

Figure 4 shows the time sequence of screen dumps of one of our simulation runs. As shown in Fig. 4, a single explorer *mCell* can travel from a randomly assigned position on the left to a given destination on the upper right. Both the *mCell*'s initial position and the positions of all obstacles are randomly generated for each simulation run.

In this case study, the *mCell* acts solely based on the task assignment (represented as FRs) and its sensory information without memory and planning. The $\text{FBR}_{\text{FT}}$ constantly transforms the perceived *task field* into local *behavior field*, allowing the *mCell* to "know" what are possible valid behaviors at each moment. Furthermore, the $\text{FBR}_{\text{BS}}$ converts behavior or action potential into specific actions.

As one may imagine, when the density of obstacles increases, the *mCell* may be trapped on its way and not be able to reach the destination. Our simulation results verified this statement. To investigate how different FBR strategies may influence the "success rate" of the simulation runs, we examined two "behavior selection" strategies, i.e., $FBR_{BS-B}$ (select the best) and $FBR_{BS-G}$ (select from good enough, i.e. top 40 %, randomly). We ran 500 test runs for each obstacle density for

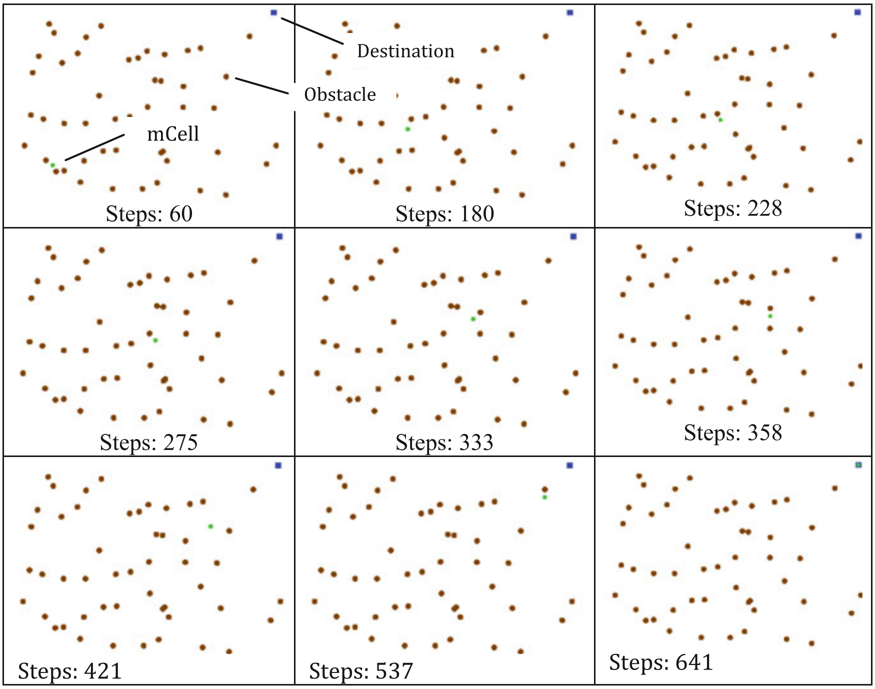


**Fig. 5** Simulation results of a single *mCell* exploring in a random obstacle field simulation results

$FBR_{BS-B}$ and $FBR_{BS-G}$, respectively, and calculate the success rate based on the 500 runs. Figure 5 shows the comparison result with 40–120 randomly assigned obstacles.

Figure 5 shows that overall the “select from good enough randomly” works better than “select the best”, especially for high obstacle density. The result indicates that behavior regulation strategies have profound impact on individual *mCell*’s performance, and the “randomness” adds “intelligence” to mechanical systems. With “select the best” strategy, a *mCell* always targets on one single best direction for its next move. When the obstacle density is low, this strategy can likely produce ideal performance in which both time and energy can be saved. When the density of obstacles increases, however, much more likely the “traps” exist in the field, resulting in lower success rate for this strategy (Fig. 6).

The “select from top 40 % randomly” strategy may not work perfectly in terms of saving time and energy. However, when the environment becomes more unpredictable and unfriendly, the *mCell* can robustly sustain the environmental change and maintain its performance. Thanks to the randomness of behavior selection, the “traps” may be overcome by the *mCell* through internal variability. Only the intrinsic variety of the system can concur the variety of the environment [1].

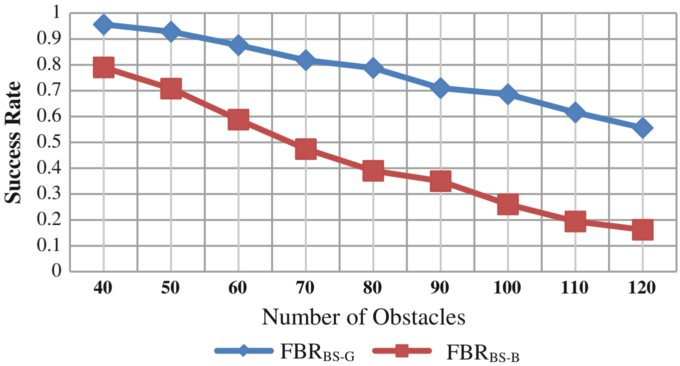


**Fig. 6** Comparison of "select the best" *($FBR_{BS\text{-}B}$)* and "select from *top* 40 % randomly" ($FBR_{BS\text{-}G}$)

## *Case Study 2: CSO Mover System*

In the single *mCell* case study, we demonstrated how *tField* can be defined, *bField* be generated and *behavior selection* be carried out through field-driven regulation (FBR). To investigate how FBR may impact on the emergence when multiple such *mCells* work together for a single task, we conducted the second case study. In Case Study 2, the task for multiple identical *mCells* is to move an object from a start point to a destination across a field with randomly distributed obstacles. At a given time, a *mCell* must decide on which direction to push the object. The movement of the object is the result of the emergent behavior of *mCells* pushing the object.

In this case study, all *mCells* can only push from their centers to the object's center with the same force. The behavior of each *mCell* is to choose a "right" location to push the object. The three functional requirements are:

FR1 = "stay close to the object",
FR2 = "push object to destination", and
FR3 = "avoid obstacles".

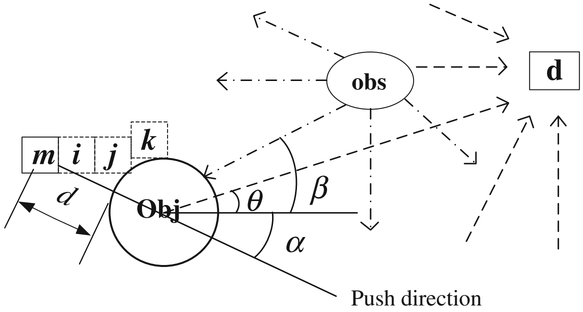
A *mCell* can choose a location to the object, so the three behaviors are:

$b_1$ = "move to locations as close as possible to the object",
$b_2$ = "push the object towards destination", and
$b_3$ = "push the object away from obstacles".

### Task Field

Similar to the previous case study, we also use parameter $\theta$ to represent the *attraction field* and $\beta$ the *repelling field*. In addition to those two, this case study introduces a new *attraction field* $d$ as the relative distance from *mCell* to the

Fig. 7 Tasks field for *mCell m*



Object. The related task field is shown in Fig. 7 and besides *mCell* m there are *mCells* i, j and k in dash line. Combining the three, we have task field for *mCell m*:

$$tFiled_m = \{d,\ \theta;\ \beta_1,\ \beta_2,\ \ldots,\ \beta_n\}; \quad \text{where}, n = \text{no. of obstacles}$$

where, $n$ = no. of obstacles

## Behavior Regulation

The two steps behavior regulation described in the previous case study is still valid in this case:

**Behavior field and *FBR*$_{FT}$**: In this example, the *bField* or *bProfile* determines the likelihood in which a *mCell* is taking its next move to either stay in the current location to push the object or move to other locations. The relative location for *mCell* is represented by α and *d*. For one destination and on obstacle, we introduce the following *FBR*$_{FT}$:

$$\begin{aligned} bField_m(\alpha, d) &= FRB_{FD}(tField_m, B) \\ &= \{\alpha, d, p_d, p_\alpha, q_\alpha\} \\ &= \left\{\alpha, d, \frac{1}{\sqrt{2\pi}} e^{-\frac{d^2}{2}}, \frac{1}{\sqrt{2\pi}} e^{-\frac{(\theta-\alpha)^2}{2}}, \frac{1}{\sqrt{2\pi}} \left(1 - e^{-\frac{(\beta-\alpha)^2}{2}}\right)\right\} \end{aligned}$$

where α: the angle to an arbitrary predefined coordinate

$d$: the related distance.
$p_d$: probability that distance d should be taken
$p_\alpha$: probability that pushing direction α should be taken
$q_\alpha$: probability that pushing direction α should be avoided

**Behavior selection and *FBR*$_{BS}$**: After the behavior field is established, a *mCell* needs a mechanism for behavior selection. In this case study, we assume that the *mCell* will change their location when the probability is below a threshold instead of choosing the "best" locations.

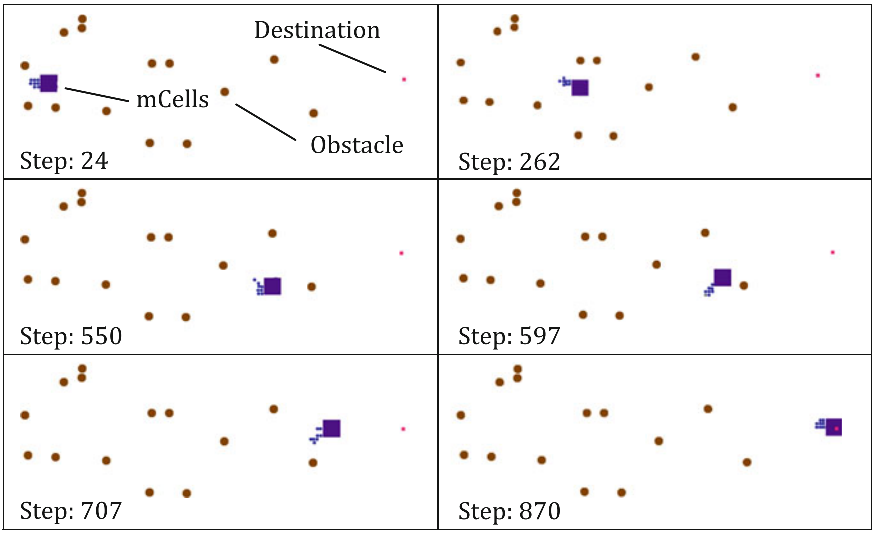


**Fig. 8** Simulation for design case 2, CSO object mover

$FBR_{BS}$ = [Select any action, randomly from the actions that has a bigger than threshold probability in the *bField*]

## Simulation Results

Figure 8 shows the time sequence of simulation screen dumps. Each *mCell* chooses a location to push the blue square Object. Each *mCell* attempts to choose a "highly" recommended zone and move into it when the zone of its current location has the probability below the threshold. There is no explicit communication between the *mCells*. However, the *mCells* interact indirectly by avoiding overlapping with each other. Our simulation results showed that with the setup of this simulation, in almost all test runs, the *mCells* were successful in pushing the square object into its destination.

One advantage of this behavior based design is that the shape of the Object and therefore the shape of the overall system are not predefined. *mCells* observe the world and decide on their behaviors locally, as the global behavior and result emerge. Based on Kolmogorov complexity measure, our CSO system of multiple *mCells* can be considered highly complex since the states of each *mCell* changes dynamically without certainty and it takes a rather long description to capture the whole system. However, using FBR makes it possible to regulate *mCells*' behaviors and to lead the emergence process to a productive direction.

Figure 9 illustrates the dynamically changing behavior field (*bField*), and how *mCells* choose their behaviors (i.e., locations) through FBR. As shown in Fig. 8 different current situations introduce two different *bFields*. Depending on the

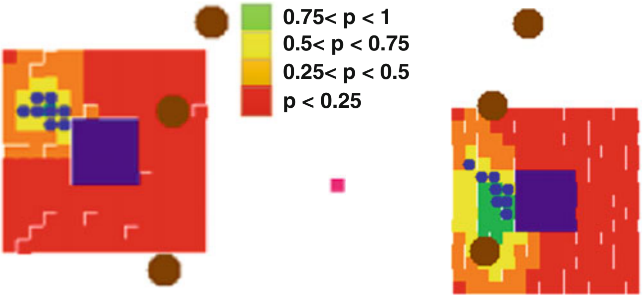


**Fig. 9** Illustration of the dynamic bField of the CSO mover in the simulated field of obstacles

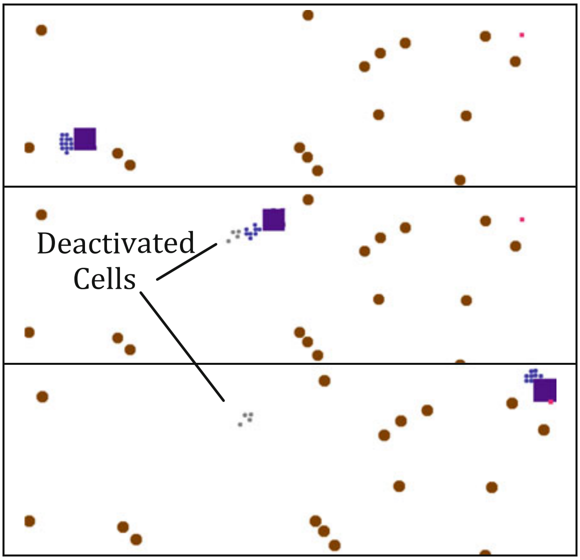


**Fig. 10** Resilience test by deactivating 4 of 12 *mCells* at step 400

relative locations of the destination, obstacles and the object, the field changes, as shown in Fig. 9 as color changes. Different colors in Fig. 9 correspond to different probabilities. The *mCells* try to choose the "green" or "yellow" zone to occupy. Through the use of field driven behavior regulations (FBR), the system dynamically adapts to its new situations even for the simple designed *mCells* of limited capability. The system can move the object in an unknown environment by *mCells* using the fields as their dynamic vision of the world. It is conceivable that the *bFields* and FBR concepts can be applied to those task situations where physical fields and chemical fields exist. We plan to expand our application example domains to assess the effectiveness of our field and FBR concepts.

In our simulation test runs, we also examined how the system might perform if some *mCells* become inactive. Figure 10 shows the resilience of the overall system when some of the *mCells* become “dead” during the simulation. There are four *mCells* that were deactivated at the step 400, since the system is fully decentralized, deactivated *mCells* had little influence to the rest of the *mCells* in the system. This way, although the system losses its performance due to the loss of *mCells*, it could still successfully accomplish the task of moving the object to its destination, showing the system resilience.

Because CSO systems are decentralized and have redundancies maintained among its *mCells*, they are more resilient than the systems with specified local functional components. When one part of the system fails, other nearby *mCells* can modify its functionality and redistribute their functions. This way, the system cannot only adapt to the environmental change but also to the system change.

## Conclusions

This paper presents a field based behavior regulation approach to designing cellular self-organizing (CSO) complex systems. The concept of CSO systems is developed based on the observations of (1) that current engineered systems are inherently incapable of dealing with variable functional requirements, changing environment situations, and possible system failures, and (2) that natural systems including dynamical systems and biological systems are formed in a bottom-up fashion and inherently equipped with capabilities to deal with uncertainties and unknowns. By combining the current engineering concepts of functions with the fundamental mechanisms of stability, self-organizing and DNA, our proposed CSO systems framework promises a different approach to developing complex engineered systems.

In our proposed CSO systems, self-organization is the key concept and mechanism. To make mechanical cells self-organize in a bottom–up fashion, a field concept is introduced that allows cells to sense the tasks and environment and formulate a task field as a model of the task world. By following the field based behavior regulation mechanism that we devised, mechanical cells can transform the sensed task field into their behavioral field in which their possible behaviors/ actions are profiled and ready to be selected. The final behavior section is carried out by the FBR behavior selection operator. It is worth mentioning that our field-based behavior regulation framework is composed of distinguishable stages of cellular operations, including sensing, field formation, field transformation, and behavior selection. These operators together with their associated variables provide a rich design space for us to explore and design CSO systems. The case studies discussed in the paper demonstrated how different FBR behavior selection strategies may yield different performances, and how transformation from task field to behavioral field determines the system behaviors and capabilities. It is expected that different domain tasks require different designs of FBR mechanisms. Future research is needed to classify the domain tasks and explore various possible FBR designs.

Our current work on this research includes expanding the case study into more sophisticated problem domains, examining trade-offs of having various combinations of *mCells* including heterogeneous ones and between swarm *mCell* structures as we presented in this paper and more structured organizations that require more tight connections, e.g., physical dockings, among *mCells*.

This paper is based on the work supported in part by the National Science Foundation under Grant No. CMMI-0943997. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

1. Ashby WR (1958) Requisite variety and its implications for the control of complex systems. Cybernetica 1(2):83–99
2. Kitano H (2002) Systems biology: A brief overview. Science 295(5560):1662–1664
3. Subramanian L, Katz RH (2000) An architecture for building self-configurable systems. In: IEEE/ACM workshop on mobile ad hoc networking and computing Boston
4. Kopetz H (1998) Component-based design of large distributed real-time systems. Control Eng Prac 6(1):53–60
5. Gu P, Hashemian M, Sosale S, Rivin E (1997) An integrated modular design methodology for life-cycle engineering. CIRP Ann—Manuf Technol 46(1):71–74
6. Gershenson JK, Prasad GJ, Allamneni S (1999) Modular product design: a life-cycle view. J Integr Des Proc Sci 3(4):13–26
7. von Neumann J (1966) The theory of self-reproducing automata. In: Burks A (ed), University of Illinois Press, Champaign, Illinois
8. Weisbuch G (1991) Complex systems dynamics: an introduction to automata networks. Santa Fe Inst. Stud Sci Complex 2
9. Butler Z, Kotay K, Rus D, Tomita K (2001) Cellular automata for decentralized control of self-reconfigurable robots. In: Proceedings of the ICRA 2001 workshop on modular robots
10. Wolfram S (2002) a new kind of science. Wolfram Media, Inc. IL, USA
11. Zouein G (2009) A biologically inspired DNA-Based approach to developing cellular adaptive systems. PhD Thesis, USC, Los Angeles
12. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge, Mass
13. Gell-Mann M (1994) The Quark and the Jaguar: adventures in the simple and the complex. Freeman WH, USA, San Francisco, CA
14. Gardner M (1970) Mathematical games: the fantastic combinations of John Conway's new solitaire game 'life'. Sci Am 223:120–123
15. Ferguson S, Lewis K (2006) Effective development of reconfigurable systems using linear state-feedback control. AIAA J 44(4):868–878
16. Martin MV, Ishii K (2002) Design for variety: developing standardised and modularized product platform architectures. Res Eng Des 13(4):213–235
17. Fukuda T, Nakagawa S (1988) Approach to the dynamically reconfigurable robotic system. J Intell Robot Syst 1(1):55–72
18. Unsal C, Kilic H, Khosla P (2001) A modular self-reconfigurable bipartite robotic system: Implementation and motion planning. Kluwer Autonom Robots 10:23–40
19. Yim M, Zhang Y, Duff D (2002) Modular robots. IEEE Spectr 39(2):30–34
20. Shen WM, Krivokon M, Chiu H, Everist J, Rubenstein M, Venkatesh J (2006) Multimode locomotion for reconfigurable robots. Auton Robots 20(2):165–177

21. Shen WM, Salemi B, Will P (2002) Hormone inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. IEEE Trans Robot Autom 18(5):700–713
22. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison Wesley Longman, Inc, New York
23. Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. The MIT Press, Cambridge
24. Maher ML (2001) A model of co-evolutionary design. Eng Comp 16:195–208
25. Fogel LJ, Owens AJ, Walsh MJ (1996) Artificial intelligence through simulated evolution. Wiley, New York
26. Parmee IC (1997) Evolutionary computing for conceptual and detailed design. In: Pélaux J, Galán M, Cuesta P (eds) Genetic Algorithms in Engineering and Computer Science. Winter G, Wiley
27. Bentley PJ (1999) Evolutionary design by computers. Morgan Kaufmann Publishers Inc, Burlington
28. Jin Y, Zouein GE, Lu S (2009) A synthetic DNA based approach to design of adaptive systems. CIRP Ann 58(1):153–156
29. Zouein G, Chen C, Jin Y (2010) Create adaptive systems through “DNA” guided cellular formation. In: Proceedings of 1st international conference on design creativity (ICDC 2010), Kobe, Japan

# Understanding Design Concept Identification

**Ivey Chiu and Filippo A. Salustri**

**Abstract** In the design literature, the term *design concept* is often used de facto, or with only a brief definition provided. Despite the cursory definition for *concept*, the design process rests heavily on concepts, e.g., brainstorming and generating multiple design concepts, and subsequently identifying design concepts for concept selection, evaluation and development, etc. Concepts and concept formation are of particular interest in psychology, as concepts play a central role in human cognition. Concepts and concept identification are also of interest in other fields such as archaeology, bioinformatics and education. In this paper, we explore the process of design concept identification and address the issue of identifying design concepts in free-form text. Our exploratory experiment uses text transcripts of verbal concept generation sessions to first investigate agreeability between human concept identifiers. Next, we perform a language analysis on the transcripts to uncover language patterns that may differentiate between text segments containing concepts and text segments not containing concepts. Our results show that humans are adept at identifying and agreeing upon concepts (average agreeability >0.70), and that there are significant language differences that may distinguish concept segments from non-concept segments (i.e., non-concept segments have significantly more verbs and borderline significantly more self-references than concept segments). In general, automated concept identification may lead to better integration of early conceptual design with more detailed and computable downstream processes, resulting in a unified design workflow.

---

I. Chiu · F. A. Salustri (✉)
Ryerson University, Toronto, ON, Canada
e-mail: salustri@ryerson.ca

I. Chiu
e-mail: ivey.chiu@ryerson.ca

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0_34,

## Introduction

Design concepts generated in early stages of the design process are key as concepts influence the rest of the design realization process and affect design success (e.g., creativity, functionality). However, we observe that the term *concept* is not well defined and is often used de facto in the engineering design literature, or with only a brief definition provided. The other observation is that the task of concept identification is difficult. This difficulty relates to the ambiguity and lack of information in early stages of design where concept generation occurs.

Despite the lack of a definition for the term *design concept*, engineers and designers appear to demonstrate an understanding of design concepts and how to identify design concepts. Better understanding of design concepts and the design concept identification process may lead to the development of methods for automated concept identification and to computable concept representations that can be used as input to downstream design processes already computable e.g., CAD, optimization, etc. This may lead to faster design and prototyping, reduction of production lead-time and a more integrated design workflow.

In this paper, we conduct an exploratory experiment to investigate concept identification based on free-form text representations of concepts, either originally presented as text (e.g., in books or websites) or transcripts of concept generation sessions of either groups or individuals. Specifically, we examine concept identification using transcripts of individual verbal concept generation sessions. Verbalizing concepts, such as in verbal protocols experiments found in the Delft Protocols and other design studies, e.g., [1, 2], etc., and other methods of verbalizing concepts such as group brainstorming sessions, are a common method of eliciting design concepts.

Our investigations occur from both a cognitive and computational approach. First, we examine the agreeability between two human coders who reviewed transcripts of concept generation sessions and identified and coded the concepts. Second, we analyzed the concept-coded transcripts and compared language patterns between transcript segments containing concepts and segments not containing concepts to determine if there are language differences that may assist in automatically identifying concepts.

In the next section, we review *concepts* in both the design literature, and the wider scientific literature.

### *Concepts in Design Literature*

First, we examine the term *concept* in engineering design textbooks, and then we turn our attention to *concept* in the design research literature. In many of the engineering design textbooks such as [3–5], only brief descriptions of *concept* are given. For example, in [5], they define concept as an *idea*, and also clarify that concepts may be called *schemes*. A more detailed description is found in [6] where concept is defined as something that can take the form of "written descriptions, sketches or preliminary calculations and need only be developed to the point in which they can be evaluated". In [7], they define concept as:

> [O]ne or several structures which could fulfill the given needs, demands and requirements and constraints. A concept can be a sketched interpretation or a proposed solution, but also an intellectual abstraction with relationships for a class of objects or phenomena

We underline the conditional "could" above to emphasize the uncertainty associated with concepts.

The research literature also confirms the uncertainly of *concept*. In C–K theory, *concepts* are explicitly acknowledged to be ambiguous. In C–K theory, concepts are "undecidable" propositions in the knowledge space containing all the true propositions. On the other hand, *concepts* in C–K theory are neither true nor false in the knowledge space [8]. Others, e.g., [9, 10], have focused on better understanding and pinpointing concept formation in designers, which can help "sharpen" the definition of *concept* [9]. Both [9, 10] investigate *concept* through examining the language use of designers.

However, the ambiguity and uncertainty associated with concepts appear not to hinder the concept identification process. Identified concepts are common in the literature. For example, in the Delft Protocols [1], there are many examples of concepts identified from the protocols. In one example, a concept for a bicycle bag pinpointed for further development is described as "maybe it's like a little vacuum-formed tray". In one analysis of the protocols, Ullman, Herling and Sinton [11] provide examples of concepts for each of the concept refinement levels, high, medium and abstract. An example of a concept at each level is provided below [11]:

| | |
|---|---|
| Highly refined example | They've got this em Batavus Buster |
| Medium refined example | It's like an old bike basket that way like the Wizard of Oz |
| Abstract refined example | It'd be cool if em this rack was used for something else like you take your backpack off and then this rack you can still put stuff on it |

While it may not be necessary, or even possible, to have an explicit definition of design concept, the ambiguity of concept may render it difficult to agree on concepts, e.g., agreeing on the number of concepts or the number of different concepts from a brainstorming session. This presents difficulties for automating concept identification, and in general, presents difficulties in early stages of design. Some, e.g., [8, 12], explicitly acknowledge the difficulty with the definition of concept and concept identification, yet the examples taken from the Delft Protocols show that humans readily identify concepts as part of design analysis tasks.

## *Concepts in Other Fields of Study*

Concepts and concept formation are a topic of interest in psychology, and research typically concerns how people form mental representations of a class of entities, or categories [13–16]. Concepts are of particular interest in psychology because concepts play a central role within human reasoning and inference. By forming a concept, and using a single word to denote a concept that encompasses an entire class of entities or a category, humans avoid having to label each and every new entity encountered [17], thus promoting "cognitive economy" [18]. Researchers in other fields, such as philosophy, language, mathematics, bioinformatics, artificial intelligence, software development and education, have also investigated topics surrounding concepts and concept identification. What follows is a brief review of concept research in these other areas.

A concept is regarded as an idea, or a thought [19, 20]. Starting from classic Aristotelian philosophy, the definition of concept is more structured. For an object to belong to a concept, it must meet necessary and sufficient membership conditions [17]. However, the Aristotelian model does not account for "fuzzy" concepts, such as those found in the natural world. For example, if a defining membership condition for the concept "bird" is "that birds fly", then is a penguin still considered a bird because it does not fly? In contemporary literature, concepts are seen as the bridge between the inner world, e.g., the mind or thought, and the outer world [16, 21, 22].

Generally, there is a strong connection between concepts and language, as it is difficult to think about concepts without a word, or label, for the concept. This is because labeling a concept with a word enables us to manipulate that concept in our mind [14, 16, 23, 24]. Concepts are often regarded as definitions, or at least something that involves a definition [14, 22, 25]. However, concepts are not necessarily identical to language or definitions. For example, experiments have shown that both pre-verbal infants and non-linguistic species, e.g., chimpanzees, have concepts [24]. Jackendoff [21] regards conceptual structure as not part of language per se, but part of thought, and conceptual structures are connected to "world knowledge" or "meaning", which go beyond language and definitions.

There are two prevailing theories of human concept formation: a similarity-based theory that relies on comparison to exemplars, and an explanation-based theory that relies on using principled rules to determine concept membership [17]. Concepts can be described with exemplars, categories and sets [15, 26], e.g., a set of triangles of different sizes defines the concept of a shape with three sides. Concepts can also be described using a function or a rule that identifies the set. For example, the rule "must have three sides," describes the concept of shapes known as triangles. Concepts are also commonly represented by images, e.g., concept maps or diagrams [27, 28].

In practice, concepts are likely represented using a combination of definitions and examples/images. Smith [14] theorizes that concepts are composed of a "prototype plus core", where prototypes are typical examples, and cores are definitions or rules. Definitions can be as brief as single words, combinations of adjectives and nouns to form conjunctions, or verb and noun combinations that correspond to units of thought [14]. Metaphors and analogies can also be used to indirectly describe concepts, e.g., “the brain is like a computer” [29]. Previously, we had seen concepts from the Delft Protocols that had relied on analogy, e.g., “it’s like an old bike basket that way like the Wizard of Oz” [11].

Researchers in various fields have different motivations for studying and understanding concepts. Archaeologists are interested in concepts because artifacts represent concepts, and concepts define culture [24]. Many are using knowledge about concepts to improve teaching and learning [26–28]. In bioinformatics, automatically identifying biomedical concepts, e.g., protein interactions, diseases, etc., from the vast amount of biomedical literature available may facilitate therapeutic and pharmacological development [30, 31]. Other applications include automatically determining the domain of human-human conversations, e.g., travel planning, for unsupervised surveillance [32], and automated analysis and reverse engineering of source code in software development [33].

This brief review of concept in the literature highlights the centrality of concepts in cognition, further supporting the importance of concepts in design. It also informs of strategies used in practical application of concept identification, specifically the application of language analysis. However, it is interesting to note that in applications such as biomedical concept identification and human–human conversation concept identification, many of the techniques used require matching of terms from a predefined vocabulary, e.g., “malaria”, “parasites” for disease concepts, and “flight”, “hotel”, for travel concepts. A challenge for design concept identification is that a standard vocabulary does not exist in design. Additionally, attempts to define terms or a vocabulary in advanced is hindered by the fact that design problems and concepts may be related to any number of topics in any number of domains.

## Experimental Method

In this section, we describe our experimental methods including participants, procedure and analyses.

### *Participants*

Four participants, all fluent English speakers, were recruited from the Department of Mechanical and Industrial Engineering at a large North American University. Participants consisted of three males and one female. Two of the male participants were fourth-year undergraduate engineering students and the remaining male participant was a master’s student. The female participant was a first-year Ph.D. student.

### *Procedure and Problems*

In individual experiment sessions, participants first completed three training problems to habituate them to verbalizing. Then, participants were instructed to verbalize all thoughts as they generated concepts addressing three design problems. The three problems consisted of the Bushing-and-pin orientation problem, the Snow Insulation problem and the Coal Storage problem and are summarized below:

Bushing problem: Parts that are automatically mated, e.g., a bushing and a pin, must be positioned so that their axes coincide. Using chamfers on mating parts does not solve the alignment problem. Develop a concept to center mating parts that does not require high positioning accuracy [34].

Snow problem: In Canada, snow is readily available in the winters and has good insulating qualities due to the amount of air in it. However, if the snow is packed to the point it becomes ice, it is less insulating due to the loss of air. Come up with a concept to enable snow to be used as an additional layer of insulation for houses in the winter.

Coal problem: Clean coal and clean coal combustion technologies make it possible to generate cleaner electricity. That, combined with the increasing cost of oil and natural gas, power plant operators may consider converting or reconverting their power plants from oil or natural gas back to coal. However, there may not be enough land area near the plant that can be used for on-the-ground coal storage. Propose alternative solutions to a conventional coal pile. Adapted from [4].

Fifteen minutes were allotted for each problem for a total experiment duration of approximately 45 min for each participant. Worksheets containing the training and design problems descriptions were provided to the participants and participants were allowed to use the worksheets to aid their concept generation sign process, e.g., by writing, sketching, calculating, etc. Sessions were recorded and fully transcribed for analysis purposes and worksheets were collected.

An independent transcriptionist was recruited to transcribe the experiment sessions. Transcripts were corrected for minor spelling errors, e.g., "pedal" for "petal", but were otherwise neither annotated nor changed. The following is a transcript excerpt representing approximately 30 s of one experiment session from the Bushing problem:

> …okay, so…chamfer is like a mini-funnel and that doesn't seem to fix the problem…um…so, I'm not exactly sure what to do because the funnel seems like a pretty good idea…maybe something like a magnet…

Transcripts for Participants 1 through 4 contained 4,678, 3,645, 2,603 and 3,759 words respectively.

## *Concept Identification and Agreeability Analysis*

To examine the concept identification process, an independent concept reviewer was recruited to identify and code concepts by reviewing the transcripts and worksheets. The independent coder was familiar with conceptual design based on coursework. To not bias the coder, he was not provided with any examples of previously identified concepts, and was only provided with typical textbook definitions of design concept for training purposes, e.g., [6]. As a minimum of two coders is required when agreeability is being examined, one of the investigators also reviewed and coded the transcripts and participant worksheets separately from the recruited coder, for a total of two coders.

Both the investigator and independent coder indicated concepts by physically marking, e.g., by highlighting or underlining, segments of the transcript in which they deemed to contain a concept. Both coders completed the coding process independently, without discussion. Afterwards, the coded transcripts were compared side-by-side to determine where coded transcript segments (i.e., highlighted text) corresponded between the two transcripts, thus showing concept agreement. Marked segments of text contain concepts, and by default, unmarked segments of text do not contain concepts. The concept identification process is illustrated in Fig. 1.

The following is an example of a concept segment and a non-concept segment from the Bushing problem, generated by Participant 2:

***Concept Segment****:…maybe if it has…if you had a conveyor belt and the part was sitting on the conveyor belt and it pass through some sort of ….ah…or passed underneath some sort of triangular shaped pin loader, pin loader, the pins could just be…be directly placed into the loader, and into the hole as these pieces pass under that. I guess there would be some sort of delay on that movement, it wouldn't just be a smooth conveyor.*

***Non-concept Segment****:…um…I dunno how I'm going to reinvent. I can think of other design courses that have had automated…uh…axis coincidement (sic) machines, I don't know if that the official term.*

On average, concept segments contained 153 words and non-concept segments contained 110 words, with no significant difference between the number of words found in concept and non-concept segments.

Both agreed-upon concepts, i.e., when both coders identified the same concept, and not agreed-upon concepts, i.e., when only one coder identified a concept, were added to the set of design concepts. The following example illustrates how we compiled the data and determined agreements and disagreements. In a transcript segment from the Coal problem, both the independent reviewer and the investigator identified an "underground storage" concept where the coal would be stored underground. Therefore, this was an agreement between the independent coder and investigator. However, the independent coder also identified a "storage pile" as a concept. The investigator did not consider "storage pile" to be a concept because

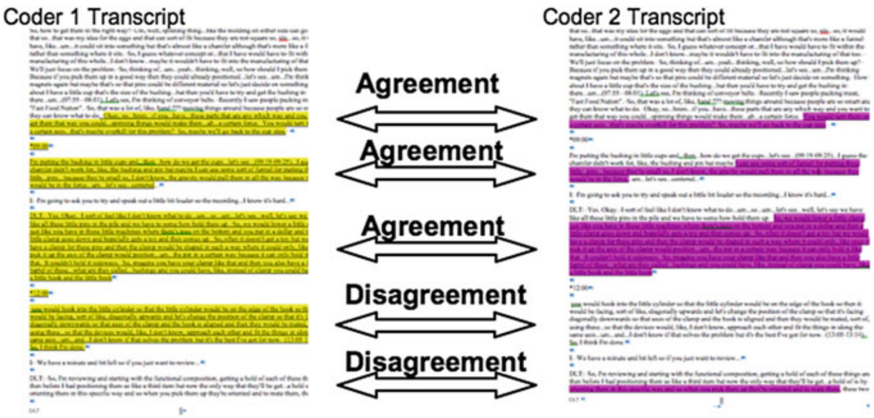


**Fig. 1** Transcript coding and concept identification. *Highlighted text* indicate concept segments identified by Coder 1 (*left*) and Coder 2 (*right*)

the problem statement specifically required concepts other than a “conventional coal pile”. Therefore, this was a disagreement. Despite the disagreement between coders, both the “underground storage” concept and the “storage pile” concept were added to the concept set.

Next, we calculated percentage agreement, or the agreeability index [35], for each participant using the following formula:

$$Agreeability = \frac{\text{Number of Agreements}}{\text{Number of Agreements} + \text{Number of Disagreements}} \qquad (1)$$

**Eqn. 1** Formula for percentage agreement or agreeability

While Eq. 1 does not account for chance agreement, this method is appropriate for exploratory experiments as it is simple and intuitive to calculate. Additionally, it is commonly used when coding protocols [35, 36].

## *Language Analyses*

Additionally, two language analyses were performed to compare linguistic differences between concept and non-concept segments. First, the syntactic parts-of-speech (POS), e.g., nouns, verbs, adverbs, adjectives, etc., in each segment were identified, or tagged, using a POS tree-tagger [37] to determine if there are POS patterns specific to concept and non-concept text segments. Then, the online version of Linguistic Inquiry and Word Count (LIWC) was also used to identify words with psychometric importance. The LIWC program analyzes text and classifies each word into one of seven psychometric categories: Articles, Big Words, Negative Emotions, Positive Emotions, Overall Cognitive Words, Self-Reference (I, me, my) and Social Words [38]. Different patterns of word use, specifically in POS and psychometric properties, may reflect participants' cognition when they are generating concepts, and when they are not generating concepts [21, 39], and may help to identify concepts.

Using the POS- and LIWC-tagged transcripts, independent t-tests were performed to compare POS and LIWC property differences between concept text segments and non-concept text segments. Independent t-tests rather than paired t-tests were used because data points are not naturally paired and there is not necessarily a defined relationship between non-concept and concept segments.

## Results

The concept identification process described previously resulted in a total of 69 concept segments from all four participants and all three problems. A total of 61 non-concept segments were identified by default.

Results of the analyses are presented below in two parts. First, results pertaining to coder agreeability are presented. Next, results pertaining to the language analyses using the POS tagger and LIWC are presented.

### *Results: Agreeability*

Using Eq. 1, an average agreeability of 0.74 was calculated across all four participants and all three problems. Specific agreeability indices are shown in Table 1 below.

### *Results: Language Analyses*

Language analyses using the POS tree-tagger and the online version of LIWC, showed that:

1. Non-concept segments contain significantly more verbs than concept segments, $t(128) = -2.86$, $p_{2\text{-tail}} = 0.0005, < 0.05$ (see Fig. 2);
2. Non-concept segments contain borderline significantly more self-references (I, me, my) than concept text segments, $t(128) = -1.66$, $p_{2\text{-tail}} = 0.100 <= 0.10$ (see Fig. 3).

**Table 1** Coder agreeability

| Participant | Problem | Total # of concepts identified | Coder agree-ability | Avg. participant agree-ability |
|---|---|---|---|---|
| 1 | 1-Bushing | 3 | 1.0 | 0.85 |
| | 2-Snow | 5 | 0.80 | |
| | 3-Coal | 4 | 0.75 | |
| 2 | 1-Bushing | 6 | 0.83 | 0.83 |
| | 2-Snow | 6 | 0.83 | |
| | 3-Coal | 6 | 0.83 | |
| 3 | 1-Bushing | 3 | 0.67 | 0.64 |
| | 2-Snow | 4 | 0.50 | |
| | 3-Coal | 4 | 0.75 | |
| 4 | 1-Bushing | 10 | 0.70 | 0.65 |
| | 2-Snow | 7 | 0.71 | |
| | 3-Coal | 11 | 0.54 | |

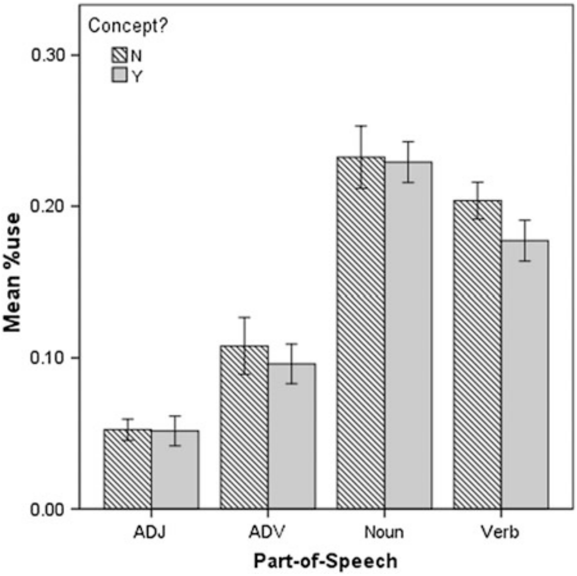
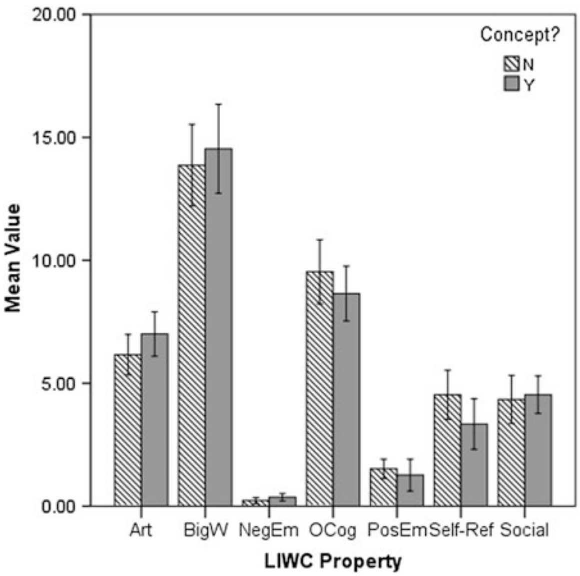


**Fig. 2** POS differences between non-concept segments and concept segments, asterisk denotes significant difference for levels of verb user

## Discussion

In this section, we first discuss agreeability and language results obtained from the experimental dataset. Next, we compare experimental results with patterns found in large, well-established corpora, or large collections of text, and discuss similarities and insights obtained from this comparison.

**Fig. 3** LIWC property differences between non-concept segments and concept segments, asterisk denotes borderline significant difference for levels of self-reference use



## *Discussion of Experimental Dataset*

In our agreeability and language analyses, we found that

1. Human coders agreed on identified concepts with an average agreeability of greater than 70 %;
2. Concept and non-concept transcript segments appear to exhibit differences in language patterns.

In terms of agreeability, for most rating and coding tasks, a 0.70 agreement is acceptable, with agreement typically increasing to 0.90 when there is additional training [35]. In this experiment, additional training was not provided and the coders did not discuss results to correlate findings. Because concept identification is such an ambiguous task, and only minimal training was provided, we consider the average agreeability of 0.74 achieved in this task to be very good. Using the concepts identified in this experiment as a training set for future concept identification tasks will likely improve concept coder agreeability.

Interestingly, agreeability indices for Participants 1 and 2 were highest (0.85 and 0.83, respectively), indicating a possibility that these two participants may exhibit more regular and detectable pattern differences between non-concept and concept segments than the other two participants. Further examination shows that Participants 1 and 2 appear to exhibit language patterns similar to each other, both in POS and LIWC patterns. Figures 4 and 5 illustrate the individual language patterns for each participant.

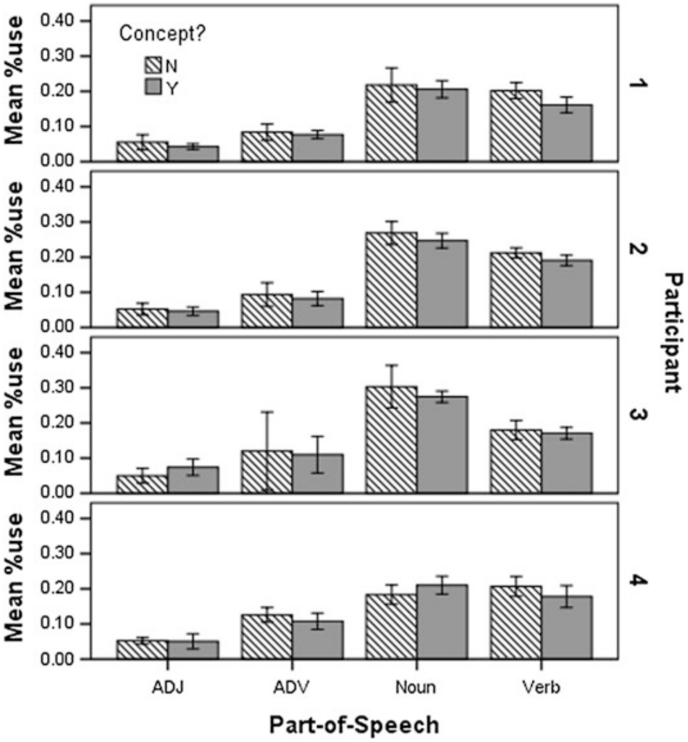


**Fig. 4** POS differences between non-concept text segments and concept text segments for individual participants. *Note* the language similarities between participants 1 and 2

Specifically, both Participants 1 and 2 use significantly more verbs in non-concept segments than concept segments, $t(21) = -2.81$, $p_{\text{2-tail}} = 0.01$ and $t(33) = -2.25$, $p_{\text{2-tail}} = 0.03$, respectively. Participants 1 and 2 also use significantly more self-references in non-concept segments than concept segments, $t(21) = -2.1$, $p_{\text{2-tail}} = 0.05$ and $t(33) = -2.03$, $p_{\text{2-tail}} = 0.05$ respectively.

## *Comparison of Experimental Results with Other Corpora*

To gain more insight into language patterns found in our experimental data, we compare results from our data with language patterns found in larger, well-established corpora. In linguistics, a corpus is a large collection of texts assembled for the purposes of studying language, e.g., word frequencies, term collocations, etc., [40]. Examining patterns found in other corpora establishes that different genres of text exhibit different language patterns.

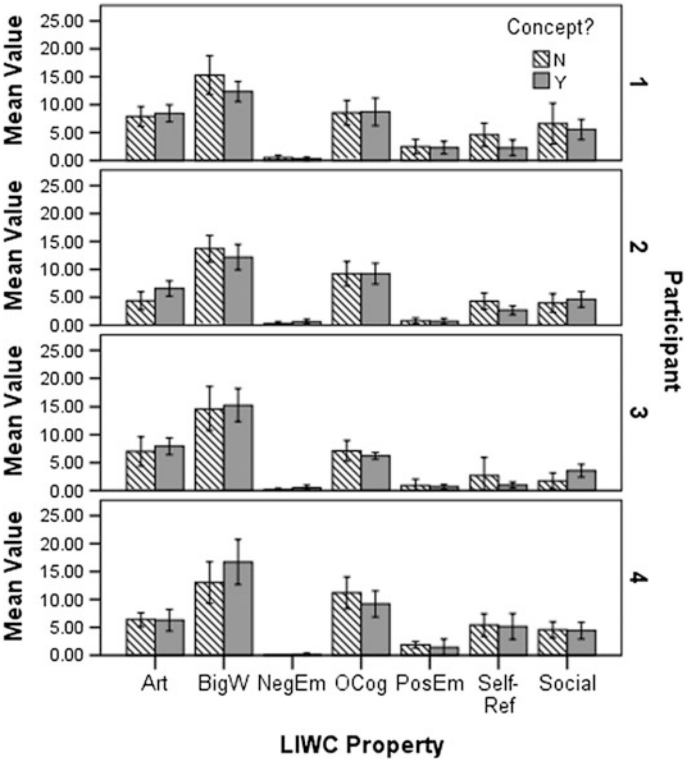


**Fig. 5** LIWC property differences between non-concept text segments and concept text segments for individual participants. *Note* the language similarities between participants 1 and 2

First, we compare POS usage rates between our dataset and POS usage rates in the combined Brown and Lancaster-Olsen-Bergen (LOB) corpora. The Brown and LOB corpora include millions of words from all domains, e.g., news reports, scientific journals, fiction, etc., and the Brown and LOB data are further separated into "imaginative" (e.g., fiction) and "informational" (e.g., newspapers) categories [41].

In the Brown and LOB corpora, informational texts use significantly fewer verbs than in the imaginative categories [41]. See Fig. 6. Similarly, we observe that in our dataset, concept text segments use significantly fewer verbs than non-concept text segments and that all four participants used fewer verbs in concept segments (see Fig. 4).

Based on this comparison, it may appear that concept segments are similar to informational texts, and that the level of verb usage may be a differentiating property between concept and non-concept text segments. A possible explanation for this difference is that in engineering design methodology, engineers are encouraged to phrase design functionality using verbs, e.g., *connect* the two parts, *move* from A to B, [3, 42].

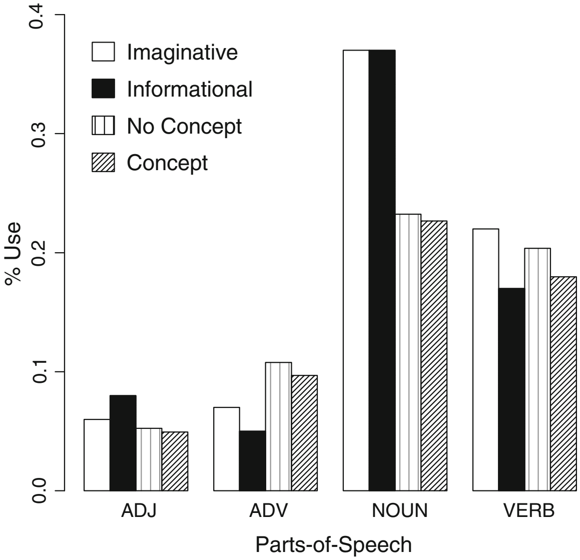


**Fig. 6** Comparison of part-of-speech usage from Brown + LOB data and experimental data

When the participants were not generating design concepts, they may be reasoning about design functionality, and thus using significantly more verbs than when they are generating and describing design concepts. More verb use in non-concept segments agrees with our intuition and the general consensus, e.g., [9, 10], that concepts are *things* and thus, verbs are less prominent in concept segments.

Next, we compare patterns of LIWC psychometric properties found in our experimental data with patterns found in the LIWC corpus. The LIWC software was developed based on analyses of over 8 million words from written texts and taped conversations to develop psychometric categories, e.g., positive/negative emotion words, cognitive words, etc., to provide insight into human cognition and emotions when analyzing text. The LIWC database is further split into personal texts, e.g., journal and diary entries, and formal texts, e.g., prepared speeches [38].

In this comparison, we see similarities between LIWC personal texts and non-concept text segments, and LIWC formal texts and concept text segments. Personal texts use fewer articles and big words than formal texts, and similarly, we observe that overall, non-concept segments use fewer articles and big words than concept segments. Both personal texts and non-concept segments use more overall cognitive words and self-reference words, with a borderline significant difference in self-references found between non-concept and concept segments. See Fig. 7 for a graph comparing property differences between the LIWC corpus and our experimental data.

Based on this comparison, non-concept segments may be similar to personal texts, and the level of self-references (found to be borderline significantly different between concept and non-concept segments) may be a differentiating property between concept and non-concept text segments. In general, pronoun use (I, me,

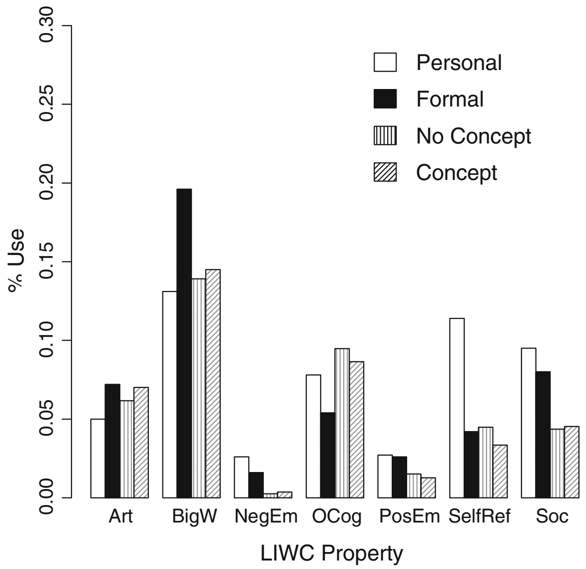


**Fig. 7** Comparison of LIWC properties between LIWC personal texts, LIWC formal texts and experimental no-concept and concept text segments

my, we, you, etc.) may indicate that the speaker/writer is thinking about, and connecting to the social world [43]. Different levels of self-references between concept and non-concept segments may indicate different levels of connectedness with the social world when the participant is either generating or not generating concepts.

In summary, a comparison between our experimental data and other well-established, large corpora found similarities between:

1. Brown and LOB informational texts and concept segments in the reduced usage of verbs as compared to imaginative texts and non-concept segments;
2. LIWC formal texts and concept segments in reduced usage of self-references as compared to personal texts and non-concept segments.

In other words, concept text segments show similarity to informational texts from the Brown and LOB corpora, and also show similarity to formal texts from the LIWC corpus.

## Limitations and Future Work

Our investigation was based on a small sample size, and thus individual language tendencies may bias results. However, we observed similar patterns between our experimental data and larger corpora based on millions of words, e.g., fewer verbs in concept segments and informational text. This may indicate that despite our small sample size, our study still provides accurate insight into language patterns found in concept and non-concept text segments from individual concept generation sessions. Limitations related to individual language tendencies can be addressed using training for concept coders and concept identification algorithms.

Future work includes:

1. Expanding this experiment to include more participants to ensure a representative sample of language use as related to concept generation;
2. Expanding the scope of language analysis to examine more language properties, e.g., verb tense, identifying potential analogies by searching for words and phrases such as “like”, or “similar to”, and also to examine patterns beyond the word level, e.g., at the phrase level;
3. Prototyping and testing a system and measuring performance based on recall and precision. Recall is defined as the number of relevant documents retried divided by the total number of existing relevant documents, and precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved [44]. The precision/recall of such a system applied to free-form text may be difficult to measure at this stage as we are uncertain of the exact number of relevant documents.

## Concluding Remarks

Despite the difficulty and ambiguity associated with the definition of *design concept* and the process of design concept identification, we demonstrate that human coders can readily identify concepts from transcripts of verbalized individual concept generation sessions, and that there is good agreement between the human concept coders. Furthermore, we uncovered significant language differences between concept and non-concept text segments that may assist in identifying concepts in free-form transcripts. These language differences can serve as the basis for an automated approach to identifying concepts. In turn, automated concept identification may result in a more integrated and efficient early design workflow.

**Acknowledgments** The authors wish to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

## References

1. Cross N, Christiaans H, Dorst K (1996) Introduction: the delft protocols workshop. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, Chichester, pp 1–16
2. Gero JS, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19:21–61

3. Pahl G, Beitz W (1996) Engineering design, a systematic approach. In: Wallace K, Blessing L, Bauert F (eds) 2nd ed. Springer London Ltd., London, UK
4. Dieter GE (2000) Engineering design: a materials and processing approach, 3rd edn. McGraw-Hill, NY
5. Dym C, Little P (2000) Engineering design a project-based introduction. Wiley, New York
6. Ullman D (2003) The mechanical design process, 3rd edn. McGraw-Hill, New York
7. Hubka V, Eder WE (1996) Design science—introduction to the needs Scope and organization of engineering design knowledge. Springer, London, UK
8. Hatchuel A, Weil B (2009) C-K design theory: an advanced formulation. Res Eng Des 19:181–192
9. Heylighen A, Martin G (2005) Chasing concepts during design: a photo shoot from the field of architecture. AI EDAM 19(4):289–299
10. Dong A (2006) Concept formation as knowledge accumulation: a computational linguistics study. AI EDAM 20(1):35–53
11. Ullman D, Herling D, Sinton A (1996) Chap. 8—Analysis of protocol data to identify product information evolution and decision making process. In Cross N, Christiaans H, Dorst K (eds) Analysing design activity, Wiley, Chichester, pp 169–185
12. Visser W (2006) The cognitive artifacts of designing. Lawrence Erlbaum Associates, Mahwah, NJ
13. Medin DL, Smith EE (1984) Concepts and concept formation. Annu Rev Psychol 35:113–138
14. Smith EE (1988) Concepts and thought. In: Sternberg RJ, Smith EE (eds) The psychology of human thought. Cambridge University Press, Cambridge, pp 18–49
15. Rosch E (1975) Cognitive representations of semantic categories. J Exp Psychol Gen 104:192–233
16. Rosch E (1999) Reclaiming concepts. J Consci Stud 6(11–12):61–77
17. Sternberg RJ, Ben-Zeev T (2001) Complex cognition the psychology of human thought. Oxford University Press, New York, NY
18. Rosch E (1978) Principles of categorization. In: Rosch E, Lloyd BB (eds) Cognition and categorization. Hillsdale, Erlbaum, NJ
19. Abate FR, Jewell E (eds) (2001) The new oxford American dictionary. Oxford University Press, New York
20. WordNet 3.0. Available online at: http://www.cogsci.princeton.edu/~wn
21. Jackendoff R (1983) Semantics and cognition. MIT Press, Cambridge, MA
22. Fodor J (1998) Concepts: where cognitive psychology went wrong. Oxford University Press, Oxford
23. Bruner JS (1964) The course of cognitive growth. Am Psychol 19:1–15
24. Gowlett JAJ (2009) Artefacts of apes, humans, and others: towards comparative assessment and analysis. J Hum Evol 57:401–410
25. Hull CL (1920) Quantitative aspects of the evolution of concepts. Psychol Monogr 28(123)
26. Tall D, Vinner S (1981) Concept image and concept definition in mathematics with particular reference to limits and continuity. Educ Stud Mathe 12:151–169
27. Safayeni F, Derbentseva N, Canas AJ (2005) A theoretical note on concepts and the need for cyclic concept maps. J Res Sci Teach 42(7):741–766
28. Weerasinghe JS, Salustri FA (2006) Use of concept maps to aid early engineering design. In: Proceedings 2006 CDEN conference, Jul 2006, Toronto, Canada
29. Gibbs RW (1997) How language reflects the embodied nature of creative cognition. In: Ward TB, Smith SM, Vaid J (eds) Creative thought: an investigation of conceptual structure and processes. American Psychological Association, Washington, DC, pp 351–374
30. Merrill GH (2009) Concepts and synonymy in the UMLS metathesaurus. J Biomed Discov Collab 4:7
31. Berlanga R, Nebot V, Jimenz E (2010) Semantic annotation of biomedical texts through concept retrieval. Procesamiento del Lenguaje Nat, Revista n° 45:247–250

32. Chotimongkol A, Rudnicky AI (2002) Automatic concept identification in goal-oriented conversations. In: 7th international conference on spoken language processing, Denver, Colorado, USA, pp 1153–1156
33. Carey MM, Gannod GC (2007) Recovering concepts from source code with automated concept identification. In: ICPC 07 15th IEEE international conference on program comprehension, pp 27–36
34. Kosse V (2004) Solving problems with TRIZ: an exercise handbook, 2nd edn. Ideation International Inc, Southfield, MI
35. Miles MB, Huberman AM (1994) Qualitative data analysis, 2nd edn. Sage Publications, Thousand Oaks, CA
36. Lombard M, Snyder-Duch J, Bracken CC (2002) Content analysis in mass communication: assessment and reporting of intercoder reliability. Hum Commun Res 28(4):587–604
37. Schmid H (1996) TreeTagger—a language independent part-of-speech tagger. Available online: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html
38. Pennebaker JW, Francis ME, Booth RJ (2001) Linguistic inquiry and word count (LIWC): LIWC 2001. Available online: http://www.liwc.net/liwcresearch07.php
39. Pennebaker JW, Graybeal A (2001) Patterns of natural language use: disclosure, personality and social integration. Curr Dir 10:90–93
40. Meyer C (2002) English corpus linguistics. Cambridge University Press, Port Chester, NY, USA
41. Hudson R (1994) About 37 % of word-tokens are nouns. Language 70:331–339
42. Stone RB, Wood KL (2000) Development of a functional basis for design. J Mech Des, Trans ASME 122:359–369
43. Campbell RS, Pennebaker JW (2003) The secret life of pronouns: flexibility in writing style and physical health. Psychol Sci 14(1):60–65
44. Witten IH, Frank E (2000) Data mining practical machine learning tools and techniques with java implementations. Morgan Kaufmann Publishers, San Francisco, CA

# Author Index

J. S. Gero (ed.), *Design Computing and Cognition '12*,
DOI: 10.1007/978-94-017-9112-0,