John S. Gero *Editor*

Design Computing and Cognition '16

Springer

Design Computing and Cognition '16

John S. Gero
Editor

Design Computing and Cognition ’16

Springer

Editor
John S. Gero
Department of Computer Science and
School of Architecture
University of North Carolina at Charlotte
Charlotte, NC
USA

ISBN 978-3-319-44988-3 ISBN 978-3-319-44989-0 (eBook)
DOI 10.1007/978-3-319-44989-0

Library of Congress Control Number: 2016950234

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Artificial intelligence, which developed as a formal discipline in the 1960s and 1970s, started to find its way into other disciplines in the 1980s. It appeared to offer a more approachable basis that mathematical modeling to model and understand designing. By the mid-1980s, the early workshops on artificial intelligence in design with published proceedings were run (e.g., Gero, JS (ed) (1985) *Knowledge Engineering in Computer-Aided Design*, North Holland, Amsterdam). We decided to inaugurate a formal conference series called *The International Conference on Artificial Intelligence in Design* (*AID*), and the first conference was held in Edinburgh in 1991 (Gero, JS (ed) (1991) *Artificial Intelligence in Design'91*, Butterworth-Heinemann, Oxford). This first conference was followed by further biennial conferences culminating in the seventh conference in the series (Gero, JS (ed.) (2002) *Artificial Intelligence in Design'02*, Kluwer, Dordrecht). Closely following and in parallel with the development of artificial intelligence, the new field of cognitive science was developing. Cognitive science research was already finding its way into the *AID* conference, when we decided to rename the conference series as *The International Conference on Design Computing and Cognition* in order to broaden its ambit, and the first conference in this new series was held in 2004 at MIT (Gero, JS (ed) (2004) *Design Computing and Cognition'04*, Kluwer, Dordrecht).

This conference, the seventh in the *DCC* series, marks 25 years since the first *AID* conference with the *DCC* series being a continuation and expansion of the *AID* series. In those 25 years, design has moved from being an arcane activity practiced and researched by only a few specialists in peripheral areas of specific fields to becoming increasingly recognized as one of the foundations of economic and social development and growth. Some aspects of designing have been packaged under the heading of design thinking and under that name have found their way into fields not normally associated with design, and fields such as management, business and marketing.

Computational models of designing are increasingly based on cognitive studies of designing (design cognition) giving them a stronger evidence based on which to found them. Cognitive studies of designing have expended beyond the idea of studying designers to using such studies to examine the effects of teaching on the development of design cognition. In a reflection on these studies, teaching is starting to be based on changing the design cognition of students. While design computing and design cognition are separate strands of design research, there are often intertwined.

The papers in this volume are from the *Seventh International Conference on Design Computing and Cognition* (*DCC'16*) held at Northwestern University, Evanston, Illinois, 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 Synthesis
Design Cognition–Design Approaches
Design Support
Design Grammars
Design Cognition–Design Behaviors
Design Processes
Design Synthesis
Design Activity
Design Knowledge

A total of 104 full papers were submitted to the conference, from which 37 were accepted and appear in these proceedings. Each paper was extensively reviewed by at least three reviewers drawn from the international panel of reviewers listed on the following pages. The reviewers' recommendations were then assessed before the final decision on each paper was taken. The authors improved their contributions based on the advice of this community of reviewers prior to submitting the final manuscript for publication. Thanks go to the reviewers, for the quality of these papers depends on their efforts.

Charlotte, USA John S. Gero

List of Reviewers

Henri Achten, Czech Technical University, Czech Republic
Robin Adams, Purdue University, USA
Saeema Ahmed-Kristensen, Imperial College, London, UK
Katerina Alexiou, Open University, UK
Janet Allen, University of Oklahoma, USA
Uday Athavankar, IIT Bombay, India
Petra Badke-Schaub, TU Delft, The Netherlands
Stefania Bandini, University of Milano-Bicocca, Italy
Lucienne Blessing, SUTD, Singapore
Nathalie Bonnardel, Aix-Marseille Université, France
Frances Brazier, TU Delft, The Netherlands
David Brown, Worcester Polytechnic Institute, USA
Janet Burge, Wesleyan University, USA
Jonathan Cagan, Carnegie Mellon University, USA
Marco Cantamessa
Hernan Casakin, Ariel University Center of Samaria, Israel
Gaetano Cascini, Politecnico di Milano, Italy
Philip Cash, Technical University of Denmark, Denmark
Amaresh Chakrabarti, Indian Institute of Science, India
Wei Chen, Northwestern University, USA
Henri Christiaans, TU Delft, The Netherlands
Graham Coates, Durham University, UK
Nathan Crilly, University of Cambridge, UK
Andrew Crooks, George Mason University, USA
Ellen Do, Georgia Institute of Technology, USA
Andy Dong, University of Sydney, Australia
Jose, Duarte, Pennsylvania State University, USA
Alex Duffy, Strathclyde University, UK
Chris Earl, Open University, UK
Claudia Eckert, Open University, UK
Athanassios Economou, Georgia Institute of Technology, USA

Tamer El-Khouly, University College London, UK
Benoit Eynard, Université de Technologie de Compiègne, France
Georges Fadel, Clemson University, USA
Katherine Fu, Georgia Institute of Technology, USA
Haruyuki Fujii, Tokyo Institute of Technology, Japan
John S. Gero, University of North Carolina at Charlotte, USA
Antonio Giretti, Università Politecnica delle Marche, Italy
Ashok Goel, Georgia Institute of Technology, USA
Gabriela Goldschmidt, Technion, Israel
Ewa Grabska, Jagiellonian University, Poland
Kazjon Grace, University of North Carolina at Charlotte, USA
Andrés Gómez de Silva Garza, Instituto Tecnológico Autónomo de México, Mexico
Sean Hanna, University College London, UK
Yuemin Hou, Tsinghua University, China
Hao Jiang, Zhejiang University, China
Yan Jin, University of Southern California, USA
Yehuda Kalay, Technion, Israel
Jeff Kan, City University of Hong Kong, China
Udo Kannengiesser, Metasonic, Germany
Akin Kazakci, Mines Paristech, France
Nick Kelly, University of Southern Queensland, Australia
Terry Knight, Massachusetts Institute of Technology, USA
Ramesh Krishnamurti, Carnegie Mellon University, USA
Ehud Kroll, Technion—Israel Institute of Technology, Israel
Djordje Krstic, Hetzel Design Inc.
Pascal Le Masson, Mines ParisTech, France
Julie Linsey, Georgia Institute of Technology, USA
Mary Lou Maher, University of North Carolina at Charlotte, USA
Anja Maier, Technical University of Denmark, Denmark
Janet McDonnell, Central Saint Martins, University of the Arts London, UK
Scarlett Miller, Pennsylvania State University, USA
Farrokh Mistree, University of Oklahoma, USA
Yukari Nagai, JAIST, Japan
Mine Ozgar, Istanbul Technical University, Turkey
Panos Papalambros, University of Michigan, USA
Morteza Pourmohamadi, Tabriz Islamic Art University, Iran
Yoram Reich, Tel-Aviv University, Israel
Luis Romão, Technical University of Lisbon, Portugal
Duska Rosenberg, Royal Holloway University of London, UK
Stephan Rudolph, University of Stuttgart, Germany
Somwrita Sarkar, The University of Sydney, Australia
Kristina Shea, ETH Zurich, Switzerland
Li Shu, University of Toronto, Canada
Vishal Singh, Aalto University, Finland

Steven Smith, Texas A&M University, USA
Tim Smithers, TS Research Consulting
Ricardo Sosa, Singapore University of Technology and Design
Martin Stacey, De Montfort University, UK
George Stiny, Massachusetts Institute of Technology, USA
Rudi Stoufffs, Delft University of Technology, The Netherlands
Joshua Summers, Clemson University, USA
Toshiharu Taura, Kobe University, Japan
Tetsuo Tomiyama, Cranfield University, UK
Irem Turner, Oregon State Universiy, USA
Barbara Tversky, Columbia and Stanford, USA
Pieter Vermaas, TU Delft, The Netherlands
Robert Woodbury, Simon Fraser University, Canada
Maria Yang, Massachusetts Institute of Technology, USA
Bernard Yannou, Ecole Centrale Paris, France
Seda Yilmaz, Iowa State University, USA
Rongrong Yu, University of Newcastle, Australia
Theodore Zamenopoulos, Open University, UK

Contents

Part I Design Synthesis

Reducing Information to Stimulate Design Imagination 3
Shiro Inoue, Paul A. Rodgers, Andy Tennant and Nick Spencer

Novelty, Conventionality, and Value of Invention 23
Yuejun He and Jianxi Luo

Characterizing Tangible Interaction During a Creative Combination Task. 39
Mary Lou Maher, Lina Lee, John S. Gero, Rongrong Yu and Timothy Clausner

Dissecting Creativity: How Dissection Virtuality, Analogical Distance, and Product Complexity Impact Creativity and Self-Efficacy 59
E.M. Starkey, A.S. McKay, S.T. Hunter and S.R. Miller

Part II Design Cognition—Design Approaches

What Can We Learn from Autistic People About Cognitive Abilities Essential to Design? An Exploratory Study 81
Andy Dong and Ann Heylighen

Solution-Oriented Versus Novelty-Oriented Leadership Instructions: Cognitive Effect on Creative Ideation 99
Hicham Ezzat, Marine Agogué, Pascal Le Masson and Benoit Weil

A Protocol Study of Cognitive Chunking in Free-Hand Sketching During Design Ideation by Novice Designers. 115
Omar M. Galil, Kirill Martusevich and Chiradeep Sen

A Systematic Review of Protocol Studies on Conceptual Design Cognition. 135
Laura Hay, Chris McTeague, Alex H.B. Duffy, Laura M. Pidgeon, Tijana Vuletic and Madeleine Grealy

Part III Design Support

Is Biologically Inspired Design Domain Independent? 157
Ashok K. Goel, Christian Tuchez, William Hancock and Keith Frazer

A Meta-Analytic Approach for Uncovering Neural Activation Patterns of Sustainable Product Preference Decisions 173
Kosa Goucher-Lambert, Jarrod Moss and Jonathan Cagan

Second Guessing: Designer Classification of Problem Definition Fragments 193
Meghna Polimera, Mahmoud Dinar and Jami Shah

The Analysis and Presentation of Patents to Support Engineering Design 209
Gokula Vasantha, Jonathan Corney, Ross Maclachlan and Andrew Wodehouse

Part IV Design Grammars

Classifications of Shape Grammars 229
Sara Garcia

From Shape Computations to Shape Decompositions 249
Djordje Krstic

Automated Induction of General Grammars for Design 267
Mark Whiting, Jonathan Cagan and Philip LeDuc

Generative Shape Design Using 3D Spatial Grammars, Simulation and Optimization 279
Luca Zimmermann, Tian Chen and Kristina Shea

Part V Design Cognition—Design Behaviors

Comparing Two Approaches to Studying Communications in Team Design 301
Hao Jiang and John S. Gero

Individual Differences in Tendency for Design Fixation 321
Song Liang Lai and L.H. Shu

Functional Thinking: A Protocol Study to Map Modeling Behavior of Designers 339
Ashwinraj Thiagarajan, Apurva Patel, Steven O'Shields and Joshua D. Summers

To Copy or Not to Copy: The Influence of Instructions in Design Fixation Experiments 359
Luis A. Vasconcelos, Chih-Chun Chen, Eloise Taysom and Nathan Crilly

Part VI Design Processes

A Self-Organizing Map Based Approach to Adaptive System Formation 379
Dizhou Lu and Yan Jin

Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks 401
Christopher McComb, Jonathan Cagan and Kenneth Kotovsky

Designerly Pick and Place: Coding Physical Model Making to Inform Material-Based Robotic Interaction 419
Daniel Smithwick, David Kirsh and Larry Sass

Knowledge Distribution and the Effect of Design Tools on the Design Process 437
Mina Tahsiri, Jonathan Hale and Chantelle Niblock

Part VII Design Synthesis

A Heuristic Approach for the Automated Generation of Furniture Layout Schemes in Residential Spaces 459
Sherif Abdelmohsen, Ayman Assem, Sherif Tarabishy and Ahmed Ibrahim

3DJ: An Analytical and Generative Design System for Synthesizing High-Performance Textures from 3D Scans 477
Sayjel V. Patel, Mark K.M. Tam and Caitlin T. Mueller

Automated Best Connected Rectangular Floorplans 495
Krishnendra Shekhawat and José P. Duarte

Individual Coffee Maker Design Using Graph-Based Design Languages 513
Claudia Tonhäuser and Stephan Rudolph

Part VIII Design Activity

Translating Analytical Descriptions of Cities into Planning and Simulation Models 537
Kinda Al-Sayed and Alan Penn

Exploring the Cognitive Dynamics of Product Appreciation 555
Niccolò Becattini, Gaetano Cascini and Francesca Montagna

A Means to Characterise and Measure the Information Processes of Engineers Using Eye Tracking 575
Duncan Boa and Ben Hicks

Personalised Specific Curiosity for Computational Design Systems 593
Kazjon Grace, Mary Lou Maher, David Wilson and Nadia Najjar

Part IX Design Knowledge

Traversing the Barriers to Using Big Data in Understating How High School Students Design 613
Robin S. Adams, Molly Goldstein, Şenay Purzer, Jie Chao, Charles Xie and Saeid Nourian

Generalizability of Document Features for Identifying Rationale 633
Benjamin Rogers, Connor Justice, Tanmay Mathur and Janet E. Burge

The Topology of Social Influence and the Dynamics of Design Product Adoption 653
Somwrita Sarkar and John S. Gero

Using Graph Complexity Connectivity Method to Predict Information from Design Representations: A Comparative Study 667
C.V. Sri Ram Mohinder, Amaninder Gill and Joshua D. Summers

Identifying Sentiment-Dependent Product Features from Online Reviews 685
Dedy Suryadi and Harrison M. Kim

Author Index 703

Part I
Design Synthesis

Reducing Information to Stimulate Design Imagination

Shiro Inoue, Paul A. Rodgers, Andy Tennant and Nick Spencer

Abstract This paper describes an experiment that is part of a larger research project that compares the visual reasoning between groups of designers and non-designers. In particular, this experiment focuses on how designers' processes of reasoning is characterized when they are given different levels of reduced information of an object in comparison to a group of non-designers. The experiment used deconstructed and scaled-down components of Gerrit Riedveld's iconic Red and Blue Chair. Three groups were given 3 different levels of information—group 1 were given components painted the same color as the original chair, group 2 were given components painted in a single (white) color, and group 3 were given unpainted (natural) components. The results suggest that the 3 levels of reduced information impacted on the designers' reasoning processes and there were clear differences in the visual reasoning processes between design and non-design participants.

Introduction

Human cognition is capable of generating a complete image of an object even if some visual elements are missing as long as visual clues are given. For example, Biederman's theory of "recognition-by-components" suggests that our perception can construct mental imagery of an object from incomplete depiction by identifying the combination of simple geometric features in the image (Biederman 1987). We can perceive meaningful objects from meaningless low-level features of information through forming patterns in our cognition (Ware 2008). Evidence also shows that even if only very small parts of an object are visible, our cognition can infer the category of it by identifying its semantic attribution (Athavankar 1989). We inherently have the ability to address incomplete visual information in order to construct a meaningful object in our imagination through reasoning.

S. Inoue (✉) · P.A. Rodgers · A. Tennant · N. Spencer
School of Design, Northumbria University, Newcastle upon Tyne, UK
e-mail: shiro.inoue@northumbria.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_1

For designers, dealing with incomplete visual information is commonplace and important for their design reasoning. In particular, during the early phases of the design process where they explore and conceptualize ideas, designers make good use of unclear and indeterminate information as clues for evolving their design ideas. Through reflective conversations with the ideas externalized on paper (Schön 1983; Goldschmidt 2003), designers discover unexpected meanings within relationships among depicted elements (Goldschmidt 1994; Schön and Wiggins 1992) when generating ideas (Suwa et al. 2000). Designers also detect unintended relationships and features even from sketches depicted for different purposes (Suwa and Tversky 2002). Goel asserted that the ambiguous nature of visual information produced during the early stages of design is not inferior but rather plays a very important role for a designer's cognitive process. The ambiguity in a designer's concept drawing can facilitate transformations of ideas (Goel 1995). The ambiguous nature of these indeterminate sketches facilitates a designer's multiple interpretations and helps to develop their design alternatives (Do and Gross 1996). Additionally, those tentative depictions that are often produced quickly and cheaply (Rodgers et al. 2000) are reinterpreted in order to transform, develop and generate new ideas (Menezes and Lawson 2006). Further, this ambiguity significantly supports the exploration of a wide variety of innovations and increases the number of ideas for designers (Tseng and Ball 2011). Thus, it is important for designers to be open to the incomplete state of their externalized images; accuracy and scale are not needed during the explorative stage (Goldschmidt 2014). The incomplete features of the pictorial representations produced at the early stage of design are the key factor for designers' reasoning and idea exploration.

Other than the designers' self-created depiction, uncertainty of visual information presented externally can potentially facilitate their design reasoning. Designers are able to find semantic meanings and develop diverse designs even from meaningless geometric forms as well. Butter's experiment, in Klaus Krippendorff's book "The Semantic Turn" demonstrates that a great number of different electronic products could be designed by combining a set of meaningless geometrical blocks (Krippendorff 2006). The participants in Butter's experiment interpreted different meanings of the parts of simplified blocks in the context of each other and constructed meaningful products. This result implies that designers are capable of carrying out design reasoning from ambiguous shapes imagining and manipulating the contexts around them.

There have been many investigations that focus on the role of internal and external representation at the conceptual stage of the design processes (e.g. Schön 1983; Goldschmidt 1994, 2003; Schön and Wiggins 1992; Suwa et al. 2000; Suwa and Tversky 2002; Goel 1995). However, little research has been done to investigate how reduced information externally presented as input impacts on design reasoning for designers. Reducing information decreases its explicitness of meaning and, therefore, provides multiple interpretations. This ambiguity might offer an opportunity to stimulate a designer's diversified imagination. This paper explores the potential impacts of how reduced information can prompt an individual's creative reasoning and compares design and non-design participants' responses.

Aim

The aim of the experiment reported here is to observe the impact of reduced information on a group of designers' visual reasoning, and how this might differ from other individuals that have different prior knowledge and experiences. This experiment was designed based on the finding of the authors' previous study (Inoue et al. 2015) described in the next section.

Previous Experiment

The authors' earlier study observed the process of visual reasoning of industrial design students when they are given images of an object whose descriptive information has been reduced (Inoue et al. 2015). The aim of this experiment was to identify what kinds of elements the design students rely on as clues for their visual reasoning within a range of different reduced images. An image of a simple chair was altered in 17 different ways (e.g. dismantled, dotted, exploded, and vandalized) and one given to each of the 17 participants in the study (Fig. 1).

Each participant was then asked to visualize the original object. They were also asked to draw a sketch and make a model of their imagined objects based on the altered (reduced) image. The experiment revealed that the design students focused mainly on the "material" and "compositional" aspects of an object as an important clue when they built the image. Additionally, the authors also found that these aspects are significantly supported by prior knowledge. Different types of prior knowledge such as associations between particular colors and materials, material processing, structural knowledge of objects, or identifying the semantic property of components suggested a hint for the participants to infer the materiality and compositional arrangement of the object they visualized.

Fig. 1 The image prompts used for the previous experiment

Research Questions

Based on the finding of the previous experiment, the authors conducted a second experiment that focused on the reduction of “material” and “compositional” information. In particular, the authors focused on the fact that color information can affect the imagination of materiality of an object. In this second experiment, the authors focus on 2 main questions:

- How are the reasoning processes of design students affected when they are given different levels of “color” and reduced “composition” information?
- How do different kinds of prior knowledge affect the participants’ visual reasoning when dealing with reduced levels of information?

Methodology

The experiment described here used a one-tenth scale-model of Gerrit Rietveld’s famous Red and Blue Chair designed in 1918 (Fig. 2). This chair consists of 2 standard flat panels (seat and backrest), 2 armrests, and 13 slats joined in the simplest possible way painted in red, blue, yellow, and black. All the geometric components ensure that no part dominates or is subordinate to the others (van Zijl 2010). The authors used this object because of its neutrality in form of the components that allows the participants to interpret in diverse ways.

Fig. 2 Red and Blue Chair designed by Gerrit Rietveld (1918)

Fig. 3 3 chair components sets (*left* to *right*: original colors, white color, natural color)

Experiment Components

In order to reduce the compositional information of the object, the components were arranged in order of size. In addition, the material information was reduced with 3 different types of color-coding (Fig. 3). In the authors' previous experiment, it was found that color information was one of the factors that prompted the materiality of the object. Thus, one set of chair components were prepared in the same colors (red, blue, yellow and black) as Rietveld's original Red and Blue Chair, one set were painted in white (obscuring material information) and, a final set of components were left in their natural color (indicating its materiality).

The components painted in the original Rietveld colors suggest some material information to the participants whereas the components painted in white give less material information to them. The focus here is on observing the impact of reduced levels of color information. The experiment sets out to detect how the difference in color across the 3 sets of components affects the participants' imagination of an object (including the material aspects). Additionally, the experiment sets out to observe how the process of visual reasoning based on the components that explicitly indicate its materiality differs from the other two painted sets.

Participants

In this experiment, 36 voluntary participants of Northumbria University were involved. 18 fourth year Industrial Design students from the School of Design and 18 third year non-design students from Newcastle Business School. The design student participants were regarded as having knowledge and experiences of industrial design as they were in the final year of their degree whereas none of the non-design participants had any knowledge or experience in design.

The 18 design students were divided into 3 groups comprised of 6 participants each. The other 18 non-design students were also divided into 3 groups of 6 participants. The experiment was conducted individually and each participant was given one set of chair components. Then, each participant was asked to complete the task of making a 3D model of his/her visualized object.

Procedure

The experiment was conducted individually in a quiet and closed room. Each participant completed their task following the instructions provided. During the process of model-making the participants were not interrupted by the instructor (first author) so that it allowed them to concentrate on their thinking and making. After the completion of the model making exercise the instructor interviewed the participants. The detailed procedure is as follows:

1. The deconstructed materials were provided.
2. The instructor informed the participant that the materials are scaled-down components of an object.
3. The participant was asked to visualize the object, and then to represent his/her idea using all the given components. Different types of glues were provided for constructing materials as well.
4. The participant was interviewed focusing on the object created and the way he/she evolved the ideas after the completion of the model-making exercise.

Semi-structured Interviews

Semi-structured interviews were conducted after the completion of model-making. The focus was on understanding the participants' visual reasoning processes and the outcomes. The participants were asked to respond to questions that focused on key themes such as:

- The objects creation and the way the object would be used.
- The generation of ideas and what clue(s) helped in imagining the object.
- The object's materiality.

After being asked these questions, the interviewer unveiled the complete scaled-down model of the Red and Blue Chair to the participants. Then, the participants were also asked to describe the difference between the object they created and the original chair.

Data Collection and Analysis

In the experiment, 2 types of data were collected: (i) 3D scale models that represent the participants' final idea, and (ii) the contents of the interviews that describe their processes of reasoning. The interviews were recorded by a sound-recording device, and later transcribed for analysis.

The analysis covered both the 3D outcome and reasoning processes. The photographed images of the 3D outcomes allowed us to analyze the variation in the participants' model visually. By comparing the differences and similarities in the outcomes, the authors were able to interpret the impact of the different reductive levels of information of the participants' final ideas. Additionally, the visual nature of the outcomes describes specific features of each participant's idea even if some of the created objects are in the same category. At the same time, focusing on the process of the participants' reasoning is important to reveal how they evolved their ideas and to identify the elements they considered as meaningful clues. Dealing with these aspects together allowed the data to be compared and constructed to derive the findings.

The analysis of the contents of the interviews was carried out focusing on the result of the design participants first. Afterwards, the contents of the non-design participants were analyzed based on the same focus. Fundamentally, the aim of this experiment was to investigate how the reduced information impacts particularly on the designers' reasoning and how their specific characteristics differ from the non-designers. Therefore, the same categories that emerged within the group of design participants were used as a basic framework for the coding process of the non-designers' transcripts. The written data of the design participants were analyzed using a general inductive approach (Thomas 2006) in order to capture the similarities and differences of the participants' thinking processes. Next, the non-designers' transcripts were searched and coded using the same focus as the design students. Details of the coding procedure are as follows:

1. All the transcribed raw data were read through until certain categories emerged within the contents of the design participants' interviews.
2. The categories identified in step 1 were revised and refined in order to find the common themes that can be applied across the groups.
3. The contents of the design participants' transcripts were reviewed over and over again and data re-collected through the refined categories until the authors gained a thorough understanding.
4. The design participants' transcripts were categorized in different color-coding groups.
5. The category system used for the design participants was then applied to the contents of the non-design participants to collect relevant data.
6. The gathered contents based on the same category system were compared between the design and the non-design participants.

Comparing the 2 groups, using the same coding system, allowed the authors to identify whether the design participants' reasoning was unique to design participants or if their reasoning was generic. This process also revealed the influences of the design participant's prior-knowledge.

Results

The results appear to suggest that there are both differences and similarities between design and non-design participants in the outcomes and in their reasoning processes.

Outcomes

All the images and the names of objects stated by the participants are provided below (see Fig. 4 and Tables 1 and 2). The result of the outcomes shows the clear distinction between the design and the non-design participants.

Fig. 4 Design student and non design student outcomes

Table 1 The name of the objects stated by the design participants

Participant	D1	D2	D3	D4	D5	D6
Original colors	School desk and stool	Table with bookshelf	Chair	Chair	Chair	Chair

Participant	D7	D8	D9	D10	D11	D12
White color	Throne to be carried	Journey of my thought process (sculpture)	Architectural sculpture/building/ pavilion	Table with sextant for a star finding device	Piano	Medieval looking chair

Participant	D13	D14	D15	D16	D17	D18
Natural color	Miniature desk	Boat/raft	Large opera house	Canopy	Rabbit/weathervane	Switch for workshop

Table 2 The name of the objects stated by the non-design participants

Participant	ND1	ND2	ND3	ND4	ND5	ND6
Original colors	Zoo cage	Entrance of restaurant	Chair and little table	Symbol	Temple	Terrace

Participant	ND7	ND8	ND9	ND10	ND11	ND12
White color	Little bench	Big white church	Monument	Table	Creative house	Reclining deck chair

Participant	ND13	ND14	ND15	ND16	ND17	ND18
Natural color	Royal chair/baby chair	Stair/portion of a big room	Brandenburger Tor	Entrance of café	House with chimney	Key holder storage device

The results from the design student participants appear to indicate that the types of the outcomes became more diverse when the multiple colors are reduced to a single white color. The types of outcome in the group of natural color were the richest in variety. In the group of the original 4 colors (top row), 4 out of 6 participants (67%) created a chair. When the painted color-pattern is reduced to 1 color, 2 out of 6 participants (33%) made chairs. Further, in the result of the group of natural color, none of them (0%) made chairs (Table 1). The participants who made an object that could be described as “furniture” were 6 (100%) in the group of 4 colors, 4 (67%) in the 1 color and 0 in the natural color (the miniature desk is regarded as a category of toy rather than furniture). The intended scale of visualized objects became more diverse in accordance with the decrease of the painted colors. In the group of 4 colors, the intended scales of outcomes were all in furniture-sized. In the group of 1 color, the assumed sizes of outcomes became more diverse. This tendency was even more explicit in the natural color group as they stated the objects from a miniature desk to a large opera house. These results appear to indicate that as the reductive level of color-coding information increases the types of outcomes become more diverse.

On the other hand, the result of the outcomes of the non-design participants shows no significant characteristics among the color-coding groups. The types of outcomes were varied regardless of the reductive levels of painted color. The participants who made a chair as outcome were only 1 (17%) in the group of 4 colors, 2 (33%) in the 1 color, and 1 (17%) in the natural color. The non-design participants who made furniture related objects were 1 (17%) in the 4 colors, 3 (50%) in the 1 color, and 1 (17%) in the natural color. Additionally, the scales of imagined objects do not seem to be affected by the amount of the painted-color information provided (Table 2). Thus, the result of the outcomes in the non-design participant groups seems to be relatively random compared to the one of the design participants. In other words, the different information of painted color has only affected the final ideas of the design participants.

Processes of Reasoning

The analysis of the processes of the participants' visual reasoning was conducted based on the content of the interviews. The total number of categories that emerged in the group of the design participants was 26. Further, these categories were subsequently used to collect data in the group of the non-design participants to compare the differences. These 4 features were identified as prominent characteristics (described in following sections of the paper):

- Thinking approach for making
- Reference objects
- Assumed materials other than wood
- Key elements as clue.

Thinking Approach for Making

The patterns of the participants' thinking approach were revealed through the analysis. The authors focused on how the different levels of color information affected the participants' thinking approach for the visualization of their ideas. The result shows that the approaches that the participants took can be divided into 2 ways: (i) top-down (image driven) and (ii) bottom-up (thinking by making). The top-down process commonly known as theory-driven or conceptually-driven processing (Galotti 2013) is the way to perceive an object depending on our prior knowledge. On the other hand, the bottom-up process is the way to form complex visual-patterns in meaningless features and then to construct a meaningful image of an object (Ware 2008). In this experiment, the approach where a participant started making an object based on his/her visualized idea was regarded as top-down. The approach where a participant started building the components to think without clear ideas was regarded as bottom-up.

The results appear to indicate that the thinking approaches of both design and non-design participants were fairly similar (Fig. 5). The participants had a tendency

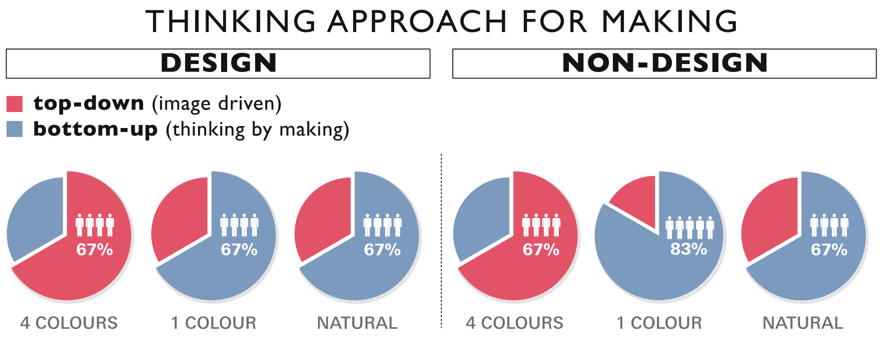


Fig. 5 Thinking approaches of the design and non-design participants

to take a top-down approach for reasoning when they were given a certain color-coding with 4 colors: 4 participants (67%) in both the design and non-design students. The participants who were given the components of both 1 color and unpaid color took bottom-up approaches: 4 design participants (67%) and 5 non-design students (83%) in the 1 color, and 4 participants (67%) for both students in the natural color.

The colors used in Rietveld's chair are very iconic and well known in a design context. Therefore, it is assumable that the original colors informed the design participants about the original chair's design so that they made a model using their design-knowledge. At the same time, although none of the non-design participants knew the original chair, the original colors still suggested concepts of an object as well. On contrary, when the painted color is reduced to white or when the material is natural colored, the components challenged the participants to visualize their ideas.

Reference Objects

The participants referred to existing objects as a source of reference in their reasoning. In the groups of design participants, the results show that the more information of color-coding given, the more participants referred to the classic chair as a clue for their reasoning (Table 3). In the 4 painted colors group, 5 out of 6 participants (83%) associated with classic design chairs such as the original Red and Blue Chair or the Charles Rennie Mackintosh Chair. In the 1 color group, 2 participants (33%) associated with classic design chairs. Just 1 participant (17%) referred to a classic chair in the group of natural color. This result suggests the types of objects that the design participants referred to can be more diverse when the painted color is reduced to a single white color. The components that indicate wooden material informed the participants about existing objects the least in the group of design participants.

Table 3 The reference objects used by the design participants

Participant	D1	D2	D3	D4	D5	D6
Original colors	Stool and desk in science lab	Red and Blue Chair	Red and Blue Chair	Red and Blue Chair	Charles Rennie Mackintosh Chair	Red and Blue Chair
Participant	D7	D8	D9	D10	D11	D12
White color	Ancient pope mobile, chairs with decoration on the back and chairs for bride	N/A	Richard Serra's sculpture	Eileen Gray's table, Lego, Tamiya models and Red and Blue Chair	Matchbox, biplane and grand piano	Platform for mountain bike, Red and Blue Chair and wheelbarrow
Participant	D13	D14	D15	D16	D17	D18
Natural color	N/A	Sail of a ship and boat	N/A	N/A	Wing of plane and Charles Rennie Mackintosh chairs	N/A

Table 4 The reference objects used by the non-design participants

Participant	ND1	ND2	ND3	ND4	ND5	ND6
Original colors	N/A	N/A	N/A	N/A	Japanese temple	N/A
Participant	ND7	ND8	ND9	ND10	ND11	ND12
White color	American trophy	Spanish and Philippine churches	Vietnamese ancient building	N/A	N/A	Framework of aeroplane, house and phone
Participant	ND13	ND14	ND15	ND16	ND17	ND18
Natural color	N/A	N/A	Old German style house	N/A	Typical house in Vietnam	N/A

On the other hand, not one participant mentioned Rietveld's iconic chair in the non-design participants groups (Table 4). Other than the design classic chairs, however, the non-design participants referred to existing objects in the 1 color group the most. The results appear to show that the particular colors used in Rietveld's chair did not prompt their association with this object. Additionally, the components that indicate the materiality of wood (natural color) informed them less about existing objects when compared with the results of the group of 1 color.

These results imply that the red, yellow, and blue colors of the components prompted the prior knowledge of the design participants. However, when the painted colors are reduced to 1 or removed, the design participants referenced random objects other than classic chairs. This feature could not be seen in the group of non-design participants. As for the types of objects that the participants referenced, both design and non-design associated with them in a variety of ways only when the painted color is reduced to a single white color.

Assumed Materials Other Than Wood

In the result of design participants, the group that mostly prompted their association of different materials other than wood was the 1 painted color (Table 5). 3 out of 6 participants (50%) stated different materials other than wood in the 4 painted-color group. When the painted color was reduced to 1 color, 5 participants (83%) referred to different materials other than wood. In the group of natural color, again, 3 participants (50%) mentioned different materials. This result suggests that when the color is reduced to a single white color, the participants imagined more material choices. In the group of natural color that indicates its materiality, half of them still stated different materials other than wood. Actually, in the natural color group, 4 participants (67%) stated that the wood material informed their reasoning. However, 5 of them (83%) explained that they attempted to avoid making an object that can be easily assumed. This result implies that the components that explicitly state its material property of wood afford some ideas about possible objects to the many participants in the group of natural color. At the same time, however, that circumstance also discouraged them to take those easy options on their decisions.

On the other hand, in the result of the non-design participants, the group of natural color seems to be the one that prompted the imagination of different materials the most (Table 6). In the other 2 painted groups, the participants who stated the different materials other than wood were more or less the same scores: 4 participants (67%) in the 4 colors and 3 of them (50%) in the 1 color.

The results of the design and non-design participants appears to indicate that the reduced information of the painted color to 1 only impacted on the assumption of materiality of the design participants. This feature was not seen in the group of non-design participants.

Key Elements as Clue

The types of information as key element that the participants used as clue were identified through the analysis. In the process of analysis, in total 4 themes that the participants frequently mentioned in terms of their reasoning process emerged:

- Shape/size
- Color
- Material
- Association with object.

In the result of the design participants, the elements that seem to be considered as clue were different among the 3 groups of the color-coding (Fig. 6). In the group of 4 colors, the identified elements are ‘Shape/size’, ‘Color’ and ‘Association with objects’. Five participants (83%) mentioned shape/size information and all of them (100%) considered color information as a clue. Additionally, 5 participants (83%) associated with existing objects. In the group of 1 painted color, the participants

Table 5 The assumed materials stated by the design participants

Participant	D1	D2	D3	D4	D5	D6
Original colors	Plywood/MDF and metal piping	Acrylic/plastic/ABS	Aluminum tubes/wood and aluminum and wood/plastic	Pinewood	Hardwood	Wood
Participant	**D7**	**D8**	**D9**	**D10**	**D11**	**D12**
White color	Plywood	Hard-board/wood and Metal	Metal	Brass/patina and wood and welded steel	Wood/metal/aluminum sheet	Hardwood/plastic
Participant	**D13**	**D14**	**D15**	**D16**	**D17**	**D18**
Natural color	Wood	Steel and copper and glass	Hardwood and bamboo and plywood	Wood/oxidized aluminum	Metal	Wood

Table 6 The assumed materials stated by the non-design participants

Participant	ND1	ND2	ND3	ND4	ND5	ND6
Original colors	Brick/metal/glass	Brick/iron/cobble stone/old broken brick	Wood	Obsidian/space fabric/wood	Wood	Wood/stone
Participant	ND7	ND8	ND9	ND10	ND11	ND12
White color	Wood	Cement/concrete/ceramic/bronze/glass/metal	Crystal	Wood/metal	Wood	Wood
Participant	ND13	ND14	ND15	ND16	ND17	ND18
Natural color	Wood/fabrics/sponge/plastic/metal	Glass/metal/wood/concrete	Stone	Wood/stone/metal	Stone/brick/bamboo/glass	MDF/plastic/acrylic

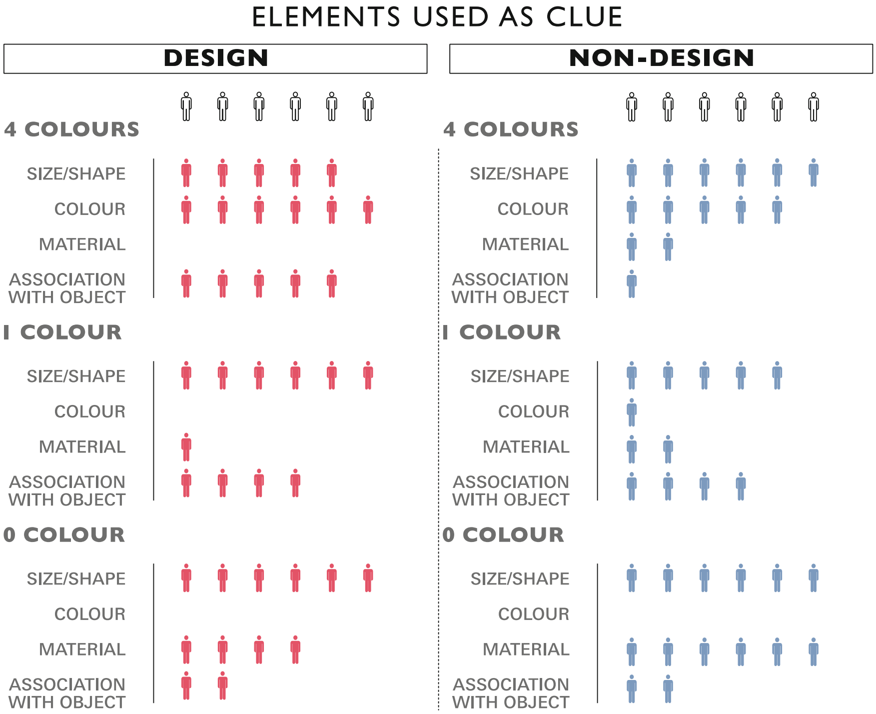


Fig. 6 Key elements used as clue

mainly considered the elements of ‘Shape/size’ and ‘Association with objects’. All the participants (100%) stated that they considered shape/size information as a clue. 4 of them (67%) associated with existing objects during the process of reasoning. In the group of natural color, they mainly considered the elements of ‘Shape/size’ and ‘Material’. All the participants (100%) mentioned shape/size information was used as a clue. 4 of them (67%) stated that material information was important.

The result of the non-design participants appeared to be similar to the one of the design participants. In the group of 4 painted color, the participants mainly focused on the elements of both ‘Shape/size’ and ‘Color’. All the participants (100%) considered the information of shape/size and 5 of them (83%) used the color information as a clue. The element of ‘Association with objects’ that was one of the elements the design participants considered was not identified in this group. This fact implies that particular color-coding encouraged only the design participants to associate with existing objects as reference. In the group of 1 painted color, the elements of both ‘Shape/size’ and ‘Association with objects’ are identified as prominent characteristics. 5 participants (83%) considered shape/size information as important clue, and 4 of them (67%) associated with objects. In the group of natural color, the identified elements were ‘Shape/size’ and ‘Material’. All the participants (100%) stated both the elements were used as an important clue.

The patterns of the identified features seem to be similar, except for the results of the 4 color groups, in the result between design and non-design participants (Fig. 6). The particular color-coding activated the prior knowledge of the design context and hence 83% of the design participants associated with the design classic chairs in the 4-color group. Other than that, the impact of the prior knowledge was not so prominent on the result of the elements the both groups of the participants used as clue.

Conclusions

This experiment has shown how a group of design participants' visual reasoning is characterized in comparison to a group of non-design participants, when they are given reduced information of an object. Additionally, it also showed how prior knowledge affects a group of design participants' reasoning processes. The results show there are some similarities and differences between the participants' use of prior knowledge in their visual reasoning.

The types and intended scales of outcomes of the design participants became more diverse when the color-coding is reduced to a single color. This characteristic became even more prominent when they were given the components that indicate the object's materiality. For the non-design participants, the types and the intended scales of outcome were just random regardless of the different levels of information given. This result suggests that giving the components reduced to a single color and the natural color led only the designers to more unpredictable and diverse outcomes. On the contrary, the multiple-colored components activated the designer's prior-knowledge and brought similar types of outcomes.

As for the process of reasoning, the multiple colors used prompted an 'image driven' approach for making models for both design and non-design participants. On the contrary, when the amount of color information is reduced or removed, the participants had a tendency to take a 'thinking by making' approach regardless of the differences of their prior knowledge. With regards to reference objects, the single color prompted the association of objects the most regardless of prior knowledge. The natural color components encouraged neither the designers nor the non-designers to reference existing objects in their reasoning processes. In terms of material, the single color components prompted the diversity of the designers' imagination whereas they did not impact on the non-designers' reasoning processes. The natural color components did not particularly prompt the imagination of materiality for the designers, as the result was the same as the one of the multiple colors. Additionally, the natural color prompted the imagination of materiality the most in the result of the non-designers. This result appears to suggest that the designers tend to stick to the idea of wood material in their reasoning process. Finally, the types of elements that the participants used as a clue were more or less the same regardless of the differences of their prior knowledge except for the group of multiple colors. Particular colors given seemed to only prompt the design participants' prior knowledge of design so that they associated with design classic chairs as a reference in their reasoning.

Thus, reduced information of both painted colors and composition certainly affected the design student's reasoning in many ways. Particularly, the reduced information to a single color brought the diversity to the designers' reasoning process regarding reference objects and materiality. Further, although prominent features were not identified in the designers' reasoning processes in the group of natural color the types and the intended scales of outcome became the most varied. Thus, the reduced information appears to assist the designers to diversify the outcomes and reasoning processes.

Discussion

This experiment was conducted under specific conditions. The components used in the experiment were from a well-known design masterpiece. Hence, if the participants were given components that have no defined answer the outcomes might have been different. However, the experiment reveals certain patterns of design participants' reasoning processes when faced with the reduction of particular parts of information.

The aim of this experiment was to observe how the designers build their imagination when they are given different levels of reduced information of an object. Also, the nature of the assignment used in the experiment is different from the actual design tasks. Accordingly, the results of the experiment do not directly contribute to the actual design practices. However, if giving less informational input to the designers can affect and facilitate their imagination, the reduced information can potentially be a useful prompt that enhances their reasoning processes to produce unexpected outcomes.

References

Athavankar U (1989) Categorization… natural language and design. Des Issues 5(2):100–111

Biederman I (1987) Recognition-by-components: a theory of human image understanding. Psychol Rev 94(2):115–147

Do EY, Gross MD (1996) Drawing as a means to design reasoning. Artificial Intelligence in Design, Palo Alto

Galotti KM (2013) Cognitive psychology in and out of the laboratory, 5th edn. Sage Publications, Thousand Oaks

Goel V (1995) Sketches of thought. MIT Press, Cambridge

Goldschmidt G (1994) On visual design thinking: the vis kind of architecture. Des Stud 15(2):158–174

Goldschmidt G (2003) The backtalk of self-generated sketches. Des Issues 19(1):72–88

Goldschmidt G (2014) Modeling the role of sketching in design idea generation. In: Chakrabarti A, Blessing LTM (eds) An anthology of theories and models of design. Springer, London, pp 433–450

Inoue S, Rodgers P, Tennant A, Spencer N (2015) Reductionism as a tool for creative exploration. In: Proceedings of 11th international european academy of design conference (EAD), Paris Descartes University, France

Krippendorff K (2006) The semantic turn: a new foundation for design. Taylor & Francis, Boca Raton

Menezes A, Lawson B (2006) How designers perceive sketches. Des Stud 27(5):571–585

Rodgers PA, Green G, McGown A (2000) Using concept sketches to track design progress. Des Stud 21(5):451–464

Schön DA (1983) Reflective practitioner: how professionals think in action. Basic Books, New York

Schön DA, Wiggins G (1992) Kinds of seeing and their functions in designing. Des Stud 13 (2):135–156

Suwa M, Tversky B (2002) External representations contribute to the dynamic construction of ideas. In: Meyer B, Narayanan NH, Hegarty M (eds) Diagrammatic representation and inference. Springer, Berlin, pp 341–343

Suwa M, Gero J, Purcell T (2000) Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. Des Stud 21(6):539–567

Thomas D (2006) A general inductive approach for analyzing qualitative evaluation data. Am J Eval 27(2):237–246

Tseng WSW, Ball LJ (2011) How uncertainty helps sketch interpretation in a design task. In: Taura T, Nagai Y (eds) Design Creativity 2010. Springer, London, pp 257–264

van Zijl I (2010) Gerrit rietveld. Phaidon Press Inc., London

Ware C (2008) Visual thinking for design. Morgan Kaufmann Publishers, Burlington

Novelty, Conventionality, and Value of Invention

Yuejun He and Jianxi Luo

Abstract Recent research has suggested that conventionality, in addition to novelty, creates value for invention. A balance between the novelty and conventionality of an invention may determine its eventual value, but is rarely understood. In this study, we use patents to approximate technological inventions, and measure the novelty, conventionality, and value of invention using patent reference and citation data from USPTO. Our empirical analyses of the patents in the 1990s reveal that medium conventionality and high novelty lead to high invention value. When conventionality is low or medium, increasing it may amplify the contribution of novelty to the value of invention. When conventionality is too high, invention value is generally low regardless of novelty. These findings provide implications and guidance to designers for enhancing the value of their potential inventions.

Introduction

In the literature, there is a common understanding that impact, and especially breakthrough impact, is a function of the novelty underlying the inventive process (Fleming 2001; Weisberg 2006). Recently, Uzzi et al. (2013) found that novelty and conventionality are not opposing factors in the production of scientific knowledge, and that the highest-impact scientific research is grounded in highly novel and exceptionally conventional combinations of prior work simultaneously. However, the design of technology and the creation of scientific knowledge are different, so how novelty and conventionality interact with each other to determine the outcomes of technological invention may be different from that in science. Our study aims to answer this question regarding technological invention, instead of scientific research.

The design literature has focused on "novelty" and "usefulness", when evaluating an invention or a new technological concept in comparison with competing

Y. He (✉) · J. Luo
Singapore University of Technology and Design, Singapore, Singapore
e-mail: yuejun_he@mymail.sutd.edu.sg

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_2

ones (Prabir and Amaresh 2007; Weisberg 1993). Novelty indicates that an invention is new, original, unexpected, and surprising (e.g. Kaufman and Baer 2004; Sternberg and Lubart 1999; Simonton 2000). Weisberg suggested that novelty is subjective to the experience of the evaluator (Weisberg 2006). Therefore, novelty may be defined with reference to the previous ideas of the individuals concerned or relative to the entirety of human history (Boden 1996). For instance, Oman et al. measured the "novelty" of a new concept as its uniqueness across all functional dimensions relative to a group of comparable ideas (Oman et al. 2013). Simonton considered an invention as the recombination of existing technologies, and the novelty of the invention as the result of unconventional combinations of prior technologies (Simonton 1999). In addition, the studies of design-by-analogy have suggested novelty may arise when the design is conceived by analogy across distant technology domains (Chan et al. 2011). However, if the domains are too distant, it is difficult to conceive a quality analogy (Fu et al. 2013). Conceptually closer rather than farther analogy appears more beneficial to design (Chan et al. 2015). Taken together, these findings suggest that conventional combinations of prior technologies, or analogies across proximate technologies, contribute to value of invention.

An invention realizes its value when or after it is endowed with utility, appropriateness, and social value, and is often measured based on usage: importance, popularity, and rate of use (Sternberg and Lubart 1999; Simonton 2004). It is naturally difficult to predict the usefulness of an invention, i.e. a new technology, when it is created, whereas ex post evaluation is possible. Regrettably, some magnificently novel inventions are so unique that they are initially rejected or ignored as irrelevant (Plucker and Beghetto 2004). Or, they are judged to be so radical and disruptive to established paradigms that they are dismissed as being outside the mainstream (Kuhn 2012). As a result, many such inventions are unsuccessful in usage and create real value for humans and society. It is a common challenge for highly novel invention to achieve a great value in usage. The analysis Uzzi et al. (2013) suggested that high conventionality may co-exist with and complement high novelty in a scientific paper to achieve high impact. Herein, we aim to explore if such a complementary relationship exists for patents.

To determine merit, new inventions must be evaluated against competing technologies. Traditionally, invention evaluation was primarily done using an expert group approach and based on experts' subjective opinions, intuitions, or experiences (Amabile 1996). Despite the existence of various procedures and techniques to facilitate expert groups and analyze their opinions, such evaluations are naturally subjective and also limited in the data sample size. As a result, it is difficult to apply rigorous mathematics for evaluation, or to test theoretical hypotheses with statistical significance. However, there have been increasing studies of data-driven evaluation of design creativity (Newell et al. 1959; Boden 1990; Grace et al. 2015; Brown 2015; Nickerson 2015). Specifically, recent studies have developed methods to analyze patent documents to evaluate patented inventions.

Patent documents contain rich details and there are also millions of patents in the public patent databases, which enable more rigorous and systematical statistical analyses on the patented inventions. For instance, Fleming (Simonton 1999) analyzed how frequent the co-classes of a patent have been assigned to the same patents in the prior history, as a way to indicate the novelty of this patent, from a knowledge combination or recombination perspective. If the classes of a patent had seldom been assigned to the same prior patents, this invention's combination of prior knowledge domains appears to be novel. Fu et al. studied the analogical distance of patented technologies, measured as functional similarity based on semantic analysis of patent documents (Fu et al. 2013). They used patents as design stimuli for new solutions to a design problem, and found that nearer field design stimuli appear more beneficial to design outcome (Chan et al. 2015).

In this paper, we aim to specify the coupled impacts of novelty and conventionality of invention on its eventual value. To investigate the relationships between novelty, conventionality, and value of invention, we analyze patents as a proxy of technological inventions, employing a set of metrics established in the study of the production of scientific knowledge based on paper publications (Uzzi et al. 2013). Our findings show the non-linear coupling of novelty and conventionality in determining value of invention. Our invention evaluation methods based on patent data and the resulting theoretical findings contribute to the growing literature on data-driven evaluation of design creativity (Newell et al. 1959; Boden 1990; Grace et al. 2015; Brown 2015; Nickerson 2015).

Data and Method

Data

Like many studies of technological invention (Fu et al. 2013; Fleming and Sorenson 2000), we use patents as a proxy of invention, with awareness of its limitations. Although companies sometimes avoid patenting to keep their inventions in secrecy and industries vary in their propensities to patent (Trajtenberg 1990; Albert et al. 1991), patent data offers a window into the development of new technologies. The rich information in patent documents allows us to characterize the inventive process and to measure the value of invention, with statistical power.

Specifically, our analysis utilizes all the US patents granted in the 1990s with 5 or more references to earlier patents (i.e. backward citations), totaling 601,715 records, in the USPTO (United States Patent and Trademark Office) database. Focusing on the 1990s ensures the data on the (backward) references and (forward) citations of the patents in our sample are complete, because prior studies have suggested that most patent-to-patent citations fall in a time lag of 10 years (Trajtenberg 1990; Hall et al. 2001). Information of references and citations in patent documents provides the basis of our analysis.

Variables

Novelty and Conventionality of Patented Invention

In patent data, each patent may be assigned to more than one patent technology classes that represent its technological fields. The fields of the referenced patents in a patent document provide the information for measuring the novelty and conventionality of the patented invention. For instance, if a patent makes references to prior patents that belong to such distant fields as "agriculture" and "computing", which are not often referenced together, it signals novelty of this patented invention. If a patent only makes references to prior patents in the fields that appear very often together in the reference lists of many other patents, it implies the conventionality of the patented invention. In reality, the reference list of a patent may include different field pairs with different co-occurring frequencies.

In this study, we use patent technology classes, specifically IPC4 classes (4-digit classes in the International Patent Classification system), to approximate the technological fields of a patent. Following Uzzi et al. (2013), we first construct 10 randomized patent citation networks and calculate a "z-score" for each IPC4 pair (Eq. 1). Such a normalized value relative to comparable random situations is often used in the network science literature (Watts and Strogatz 1998), to indicate the extent to which the empirical observation deviates from the expectation in comparable but randomized settings.

To construct a randomized patent citation network, we randomly switch the citation links with the same starting and ending years, so the randomized citation networks contain the same degree distribution and the same citation time lag as the real one. For example, if in the real citation network, a patent granted in 1999 cites 2 patents granted in 1997 and 4 patents granted in 1996, then in the randomized citation networks, it also cites 2 patents granted in 1997 and 4 patents granted in 1996. But because different patents may be assigned to different IPC4 classes, the patent's references may show different IPC4 combinations after randomization. The comparison between the real network and the randomized ones reflects the novel/conventional extents of the IPC4 pairs.

Specifically, we calculated the z-score of any pair of the 630 IPC4 classes that were assigned to the patents referred by all the US patents from 1990 to 1999. The formula for z-score is

$$z_{ij} = \frac{x_{ij} - \mu_{ij}}{\sigma_{ij}} \tag{1}$$

where z_{ij} is the relative co-occurrence frequency of the pair of classes i and j, x_{ij} is the empirically observed co-occurrence frequency of classes i and j, μ_{ij} is the average expected co-occurrence frequency of classes i and j in the comparable randomized citation networks, and σ_{ij} is the standard deviation. The average expected value μ_{ij} and the standard deviation σ_{ij} are calculated based on an ensemble of 10 randomized reference lists of the same patents, which are obtained in 10 randomized citation networks. Note that, to count the co-occurrence of IPC4 classes in the references of patents, the multiple IPC4 scheme was used.

Actually, an IPC4 pair will have a z-score higher than 0 if they co-occur in patent references more frequently than by chance. The combination of the IPC4 pair for invention is conventional. On the contrary, an IPC4 pair will have a z-score lower than 0 if they co-occur less frequently than by chance. The combination of the IPC4 pair is novel. For example, in 1999, the IPC4 pair of H04M (Telephonic communication) and H04H (Broadcasting communication) had a positive z-score of 322.415 indicating a conventional combination, whereas the IPC4 pair of H04M and A01N (Preservation of bodies of humans or animals or plants; Biocides; Pest repellants or attractants; Plant growth regulators) had a negative z-score of −75.956 indicating a novel combination.

Therefore, based on the z-scores of pairs of IPC classes, we can further measure the novel and conventional combinations of the fields of the referenced prior patents of a focal patent. Figure 1 plots the cumulative distribution of the z-scores of all pairs of IPC4 classes of the references of a patent (patent number: 5410453). This distribution provides the summary description and characterization of the space of cross-field knowledge combinations of an invention. Following Uzzi et al. (2013), we use the *median* of the z-score distribution of a patent as the measure of conventionality of the patented invention, because it describes the globally conventional level at which a patent combines prior knowledge in different fields. We use the opposite number of the z-score at the *fifth percentile* of the distribution as the measure of novelty of the patented invention, because a lower left tail in the z-score distribution indicates more novelty.

Note that, the z-scores in the distribution for a patent are calculated based on the historical data of patent class co-occurrences till the grant year of the focal patent,

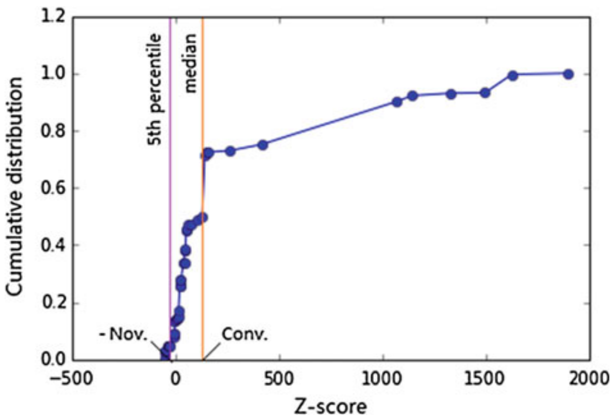


Fig. 1 Conventionality and novelty measures of a patented invention (patent number: 5410453)

because novelty and conventionality of invention are subjective to the past and present, and should vary temporarily as newer and newer technologies are developed over time (Weisberg 2006).

Value of Patented Invention

The value of an invention or new technology is realized when the new technology is endowed with utility and social significance. Prior studies have empirically shown that the count of forward citations of a patent is highly correlated with the patent's actually achieved value indicated by expert opinions, social value, awards (Trajtenberg 1990; Albert et al. 1991), and economic value (Harhoff et al. 1999; Hall et al. 2000) of the corresponding patented technologies. Thus, we follow these prior studies to approximate the value of a patented invention using the patent's forward citations. To allow analysis and comparison across fields and years, we further normalize a patent's forward citation count, by the average forward citation count of all the patents granted in the same patent class and the same year as the focal patent (Eq. 2). The formula for the invention value of a patented invention i (v_i) is

$$v_i = \frac{a_i}{\bar{\mathrm{a}}} \tag{2}$$

where a_i denotes the total count of the forward citations received by patent i, and $\bar{\mathrm{a}}$ denotes the average count of the forward citations received by all patents granted in the same IPC4 class and the same year as patent i.

We are additionally interested in the subset of inventions that achieved outstanding value and are considered breakthrough inventions. In particular, we define the patents with above top 5% value as *hit patents*, to approximate breakthrough inventions. In our analysis, the variable *hit* of a patent is 1 if the patent's value is among top 5%, and is 0 otherwise. Table 1 reports the descriptive statistics for the variables, and Table 2 displays the correlations between the variables.

Table 1 Descriptive statistics for the variables used in the following models

Variables	Mean	SD	Min	Max
Value	1	1.274	0	49.355
Hit	0.05	0.218	0	1
Conventionality	972.713	1217.711	−261.100	19,299.11
Novelty	−319.878	843.316	−19,299.11	424.734
Reference number	11.109	11.023	5	678
Inventor number	2.223	1.580	1	34

Table 2 Correlations between the variables used in the following models

Correlation matrix	Value	Hit	Conventionality	Novelty	Reference number
Hit	0.729*				
Conventionality	−0.023*	−0.016*			
Novelty	0.042*	0.029*	−0.565*		
Reference number	0.141*	0.102*	0.007*	0.111*	
Inventor number	0.080*	0.060*	0.008*	0.011*	0.075*

Legend: *$p < 0.05$

Multivariate Analysis

To investigate how conventionality and novelty determine the value of invention, we conducted one-dimensional and two-dimensional analyses with alternative focuses on mean value and hit probability. Because the values of conventionality and novelty are highly dispersed, we respectively divided the patents into 10 equally sized categories according to their conventionality and novelty values from least to greatest, and then we assigned a conventionality level and a novelty level from low to high. We analyzed the value of invention at different conventionality levels and novelty levels.

In the one-dimension analysis, we first explored the mean invention value and the hit probability (i.e. the probability for a patent to be a hit patent) at different conventionality or novelty levels. We further ran fixed effect regressions with heteroskedasticity-robust standard errors to confirm the found patterns. In the two-dimension analysis, we explored the complex mixed effects of conventionality and novelty on mean value and hit probability of patents, by plotting heat maps.

The results and discussion are provided in the next section. In fact, the results are robust when we change the analysis sample to the patents whose reference IPC4 pairs are not fewer than the quantity of 20, 30, 50, and 100, when we change the cutting line defining *hit patents* to top 10, 3, and 1% in normalized forward citation counts, and when we define novelty using the 3rd, 8th, and 10th percentile of a patent's z-score distribution.

Results and Discussion

One-Dimension Analysis

Figure 2 plots the mean invention values with the confidence intervals (±1.96 SEM) of patents equally distributed over 10 conventionality or novelty levels. On the impact of conventionality, Fig. 2a shows a parabola opening down,

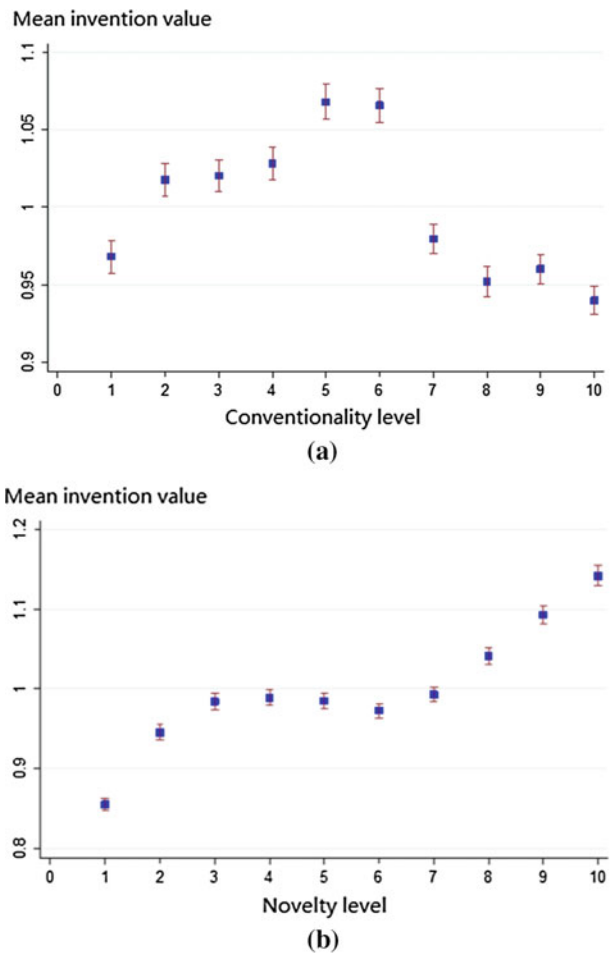


Fig. 2 Mean invention values at different conventionality or novelty levels

in which invention value increases initially till the 50th percentile, and declines after the 60th percentile onward. The highest invention value appears at a "sweet spot" of the 50th to 60th percentiles for patents according to their conventionality. This finding suggests that the highest invention value is achieved at the median level of conventionality. In contrast, Uzzi et al. (2013) found the highest value of science appears at the 85th to 95th percentiles for published scientific papers. On the impact of novelty, Fig. 2b shows a cubic curve going upwards with two peaks, one at the low novelty level and the other at the high novelty level. Clearly, the highest value of invention is achieved at the highest level of novelty. Figure 3 plots the hit probability of patents over 10 conventionality or novelty levels, and displays the similar patterns with those in Fig. 2.

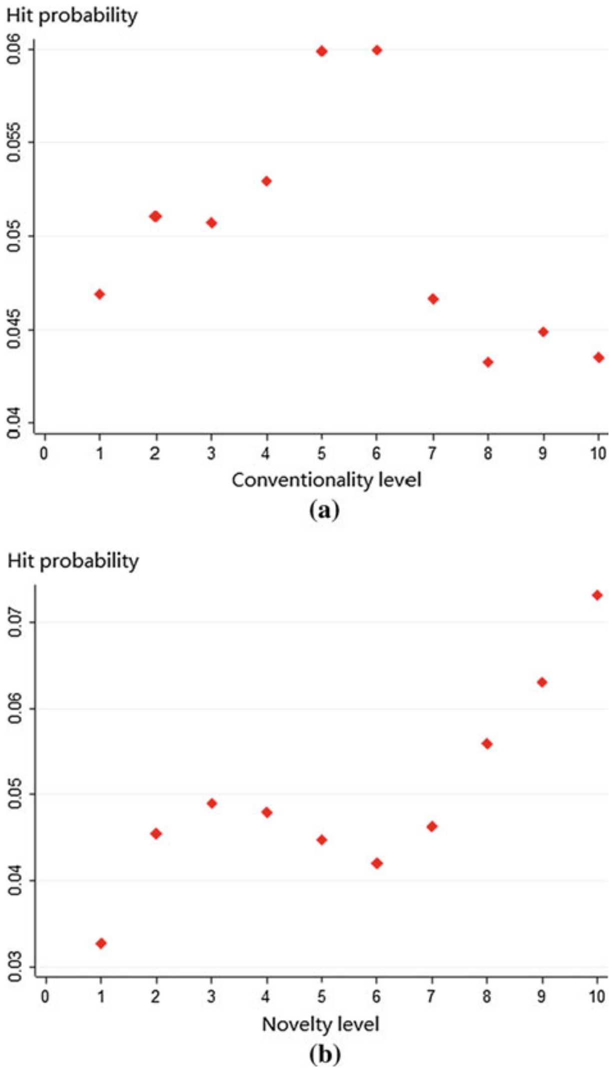


Fig. 3 Hit probability of inventions at different conventionality or novelty levels

To further confirm the patterns that were discovered in Figs. 2 and 3, we further ran multivariable regressions. The results are reported in Table 3. In the regression models, the variables *Conv_10levels* and *Nov_10levels* are 10 quantiles of the patents' conventionality and novelty levels. We also include the fixed effects of the number of references and the number of inventors of a patent, whose impact on the value of invention has been reported in the literature (Guimera et al. 2005).

Table 3 Fixed effects regressions (heteroskedasticity-robust standard errors in parentheses)

Fixed effects: *Reference_10levels*, *Team_type*				
	Value (normalized forward citation) ols regression		Hit probability logistic regression	
	Model 1	Model 2	Model 3	Model 4
Repressor				
Conv_10levels	0.0245** (0.0025)		0.0700** (0.0095)	
Conv_10levels^2	−0.0030** (0.0002)		−0.0082** (0.0008)	
Nov_10levels		0.0892** (0.0070)		0.2715** (0.0288)
Nov_10levels^2		−0.0181** (0.0015)		−0.0597** (0.0058)
Nov_10levels^3		0.0011** (0.0001)		0.0039** (0.0003)
Constant	0.7252** (0.0069)	0.6220** (0.0093)	−3.6991** (0.0289)	−3.9823** (0.0434)
Summary statistics				
$\bar{R}^2$	0.0283	0.0282		
Pseudo R^2			0.0292	0.0300
Log likelihood			−115,953.78	−115,857.34
n	601,715	601,715	601,715	601,715

Legend: *$p < 0.05$; **$p < 0.01$

The variable *Reference_10levels* is 10 quantiles of the patents' reference counts. The variable *Team_type* equals to 1 when a patent has one single inventor, 2 when the patent has two inventors, or 3 when the patent has more than two inventors. When the dependent variable is invention value (measured as normalized forward citation), we use OLS regression. When the dependent variable becomes *hit* (whose value is either 0 or 1), logistic regression is used.

In Model 1 and Model 3, the estimated coefficients of *Conv_10levels*^{2} are negative and statistically significant, which implies that the effects of conventionality on invention values follow a parabola that opens downwards. These results are consistent with the patterns in Figs. 2a and 3a. In Model 2 and Model 4, the estimated coefficients of *Nov_10levels*^{3} are positive and statistically significant, which suggests that the effects of novelty on invention values follow a cubic curve going upwards. Such results are consistent with the patterns shown in Figs. 2b and 3b.

The foregoing analyses and results show that the impacts of novelty and conventionality on invention value are clearly nonlinear. Their mixed effects are even more complex and difficult to discover.

Two-Dimension Analysis

Figures 4 and 5 plot the mean invention values and hit patent probabilities of patents in the two-dimension space constructed by the 10-quantile conventionality and novelty levels. The darkness of a cell indicates either the mean value or the hit patent probability of the patents in this cell (i.e., the likelihood for the patents to have a top 5% value among all patents in comparison). Note that the cells in the bottom left corners of both heat maps are empty, because the patent sample we analyzed here does not have any patent with extremely low conventionality and extremely low novelty at the same time. Both heat maps exhibit similar structural patterns. For instance, a "hot" area in an upside-down right triangle shape appears in both maps.

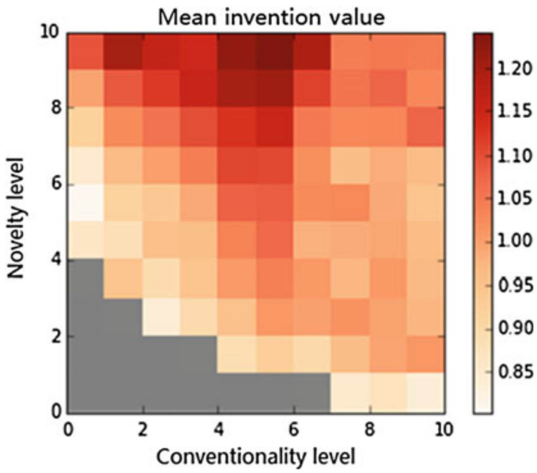


Fig. 4 Mean invention values at different combinations of conventionality and novelty levels (*grey* means no data)

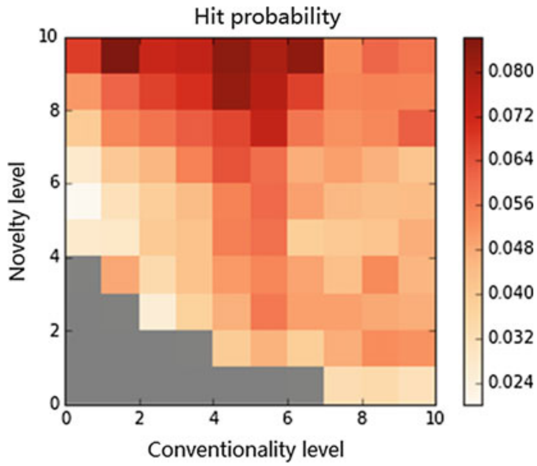


Fig. 5 Hit probabilities at different combinations of conventionality and novelty levels (*grey* means no data)

Specifically, Fig. 4 shows how the influences of novelty on invention value vary with the conventionality levels. At the low and medium conventionality levels (below the 60th percentile), the mean invention value increases with the increase in novelty. Also, the increase in the conventionality levels increases the contribution of novelty to invention value. At the medium conventionality levels compared to the low ones, relatively low novelty may lead to relatively high value of invention. This suggests that increasing conventionality may amplify the contribution of novelty to value of invention. However, when the conventionality levels are higher than the 60th percentile, the contribution of novelty to invention value is limited and non-obvious. The mean invention values are generally low, regardless of the novelty levels.

Likewise, along different novelty levels, conventionality has varied impacts on invention value. At the low novelty levels, the mean invention values are certainly low. At the medium novelty levels, the highest invention values locate at the medium conventionality levels. At the high novelty levels, the highest mean invention values appear at the low or medium conventionality levels. Figure 5 shows a similar pattern of how novel-conventionality coupling determines the hit probabilities of patents, i.e., the probabilities to have a top 5% value.

In particular, two general observations are noteworthy: (1) medium conventionality (the 50th to 60th percentile) yields relatively high invention value and hit probability at all novelty levels; (2) novelty creates significant value for invention at low and medium conventionality levels, but has little influence at high conventionality levels.

The first finding suggests that the realization of invention value relies on sufficient but not too much conventional knowledge integration of closely-related technologies. The second finding might imply the difficulty for novel technologies to be combined effectively with highly conventional ones. This relates to the coupling levels of highly conventionally integrated technologies (Fleming and Sorenson 2000) and the resulting complexity challenges to inventive efforts (Fleming 2001). Despite novel technologies being imported, it would still be difficult for them to fuse well with many conventional parts and thus create higher eventual value of the invention.

Examples

Table 4 presents a few examples of patents granted in the same field and at about the same time, but with different combinations of conventionality levels and novelty levels. All of the six patents were granted in either 1998 or 1999 with the IPC4 class “F21 V” as their primarily classification, entitled “Functional features or details of lighting devices or systems thereof; Structural combinations of lighting devices with other articles, not otherwise provided for”.

Patents 3 and 4 are built on the medium levels of conventionality, and achieved relatively high invention values. Specifically, Patent 3 is about the invention of a

Table 4 Examples of patents with different combinations of conventionality levels and novelty levels

No.	Patent ID	Year	Title
1	5967649	1999	Lighting device adapted to be positioned at any point along an electrical cord
2	5942157	1999	Switchable volume hologram materials and devices
3	5931555	1999	Background lighting apparatus for liquid crystal display
4	5857767	1999	Thermal management system for L.E.D. arrays
5	5738435	1998	Waterproof lampholder
6	5797675	1998	Spring clip for neon bulbs

No.	Conv. level	Nov. level	Value
1	2 (low)	5 (medium)	0.3568 (top 70%)
2	1 (low)	10 (high)	5.5662 (top 2%)
3	6 (medium)	6 (medium)	2.9972 (top 6%)
4	6 (medium)	10 (high)	13.9868 (top 1%)
5	10 (high)	5 (medium)	0.3987 (top 66%)
6	10 (high)	9 (high)	0.6645 (top 49%)

background lighting apparatus for liquid crystal display by integrating the technologies of displaying, lighting, measuring, optical operation, electric power, etc. Patent 4 introduces a thermal management system for L.E.D. by integrating the technologies of lighting, signaling, semiconductor, etc. The higher novelty in Patent 4, which results from combining technologies of semiconductor and lighting, may have contributed to its higher invention value, demonstrating the positive value-added of increased novelty at medium conventionality.

On the contrary, Patents 5 and 6, with high conventionality, only yielded low invention value. Patent 5 is about a waterproof lamp holder for traditional electric lamps, which conventionally combines lighting and electricity technologies. Patent 6 is about a spring clip for neon bulbs which applies pipe fitting technologies to lighting. Both patented inventions have a high conventionality level. Even though Patent 6 fancifully imports pipe-fitting technologies into lighting, it did not achieve a high value, due to the difficulty to improve the overall performance of neon bulbs. These two examples illustrate that it is difficult for inventions with too high conventionality to achieve high value, no matter how novel they are.

In contrast, Patent 2 achieved a rather high value, with low conventionality. This patented invention provides an electrically switchable material for volume holograms, which relates the technological fields of diagnosis, electric switches, optical elements, and materials together. Such a breakthrough in 1999 made a major contribution to the development of volume holography, which is still a growing field. On the other hand, Patent 1 with low conventionality but just medium novelty did not become very valuable. The invention presents a removable light assembly positioned on an electrical cord. By integrating technologies in non-portable lighting devices and electrically-conductive connections, it creates a new approach to relocate lighting devices. But this invention is not novel enough to break through the bottleneck in removable lighting systems with more convenience. The leap of value from Patent 1 to Patent 2 highlights the significance and necessity of high novelty for achieving high invention value, when conventionality is low.

Taken together, our findings in the context of technological invention are different from those in scientific knowledge production by 3, which suggested that the "sweet point" of conventionality for science is the 85th to 95th percentile and that high novelty has a positive effect at all conventionality levels (Uzzi et al. 2013). For technological invention, we have found that high conventionality limits the value of invention and also constrains the potential for increased novelty to increase the value of invention. In general, conventionality and novelty are less harmonious in technological invention than in knowledge production.

Conclusion

This study contributes to the design and creativity literature, in particular, the literature on data-driven evaluation of design creativity (Newell et al. 1959; Boden 1990; Grace et al. 2015; Brown 2015; Nickerson 2015) by revealing the non-linear and coupled impacts of novelty and conventionality on the eventual value of technological inventions. Specifically, when conventionality is low, high novelty is needed to achieve high invention value. Medium conventionality yields generally the highest invention value for all novelty levels. At high conventionality levels, invention value is generally low regardless of novelty levels. From an alternative perspective, when novelty is medium, a medium level of conventionality is desired to achieve high invention value. An invention with high novelty can achieve high invention value at low or medium conventional levels. These findings provide specific implications to technological invention, in particular, the importance of having conventionality at the medium level for realizing value of invention.

This study also contributes to design computing practices, especially in invention evaluation. Nowadays, the evaluation of invention is largely drawn upon the subjective opinions and judgements of experts based on their experience and intuition. Such evaluations are limited in scope and repeatability. This study takes a Big Data approach to provide more systematic, objective, consistent, and repeatable evaluation, which is enabled by the recent advances in complex network analysis and data sciences. The methods of invention evaluation presented here have a wide application prospect for different groups of people whose work is related to creating or evaluating inventions and new technologies, including patent lawyers, IP managers, engineers, etc.

Moving forward, this research can be further extended in a few directions. First of all, in the prior experiments, the z-scores were calculated based on an ensemble of 10 randomized reference lists of the same patents, for computational efficiency. For the next, we aim to use larger samples of randomization trials to check if the findings in the present paper hold. Second, we also aim to explore alternative measures of novelty and conventionality based on patent data, and use them to further test the findings in the present paper. Third, we also plan to develop a data-driven invention evaluation platform that automates the patent mining, measurement, and visualization procedures. So laypersons (i.e., engineers, managers, patent lawyers) can use the platform to quantitatively and visually evaluate the novelty, conventionality, and value of the patented technologies of their interests. Furthermore, we aim to put the theoretical findings here into practices, and conduct a few case studies based on the general patterns discovered by the statistical analysis in this paper.

Acknowledgements This research is sponsored by the grants from SUTD-MIT International Design Centre at Singapore University of Technology and Design and Singapore Ministry of Education Tier 2 Grant (#MOE2013-T2-2-167). We also thank Kristin Wood, Aditya Mathur, and Christopher Magee for useful comments and suggestions.

References

Albert MB, Avery D, Narin F, McAllister P (1991) Direct validation of citation counts as indicators of industrially important patents. Res Policy 20(3):251–259

Amabile TM (1996) Creativity in context: update to "the social psychology of creativity". Westview Press, Boulder

Boden M (1990) The creative mind: myths and mechanisms. Basic Books, New York

Boden MA (1996) Dimensions of creativity. MIT Press, Cambridge

Brown DC (2015) Computational design creativity evaluation. In: Design computing and cognition'14. Springer, Berlin, pp 207–224

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. J Mech Des 133(8):081004

Chan J, Dow SP, Schunn CD (2015) Do the best design ideas (really) come from conceptually distant sources of inspiration? Des Stud 36:31–58

Fleming L (2001) Recombinant uncertainty in technological search. Manage Sci 47(1):117–132

Fleming L, Sorenson O (2000) Science as a map in technological search. Available at SSRN 249578

Fu K, Chan J, Cagan J, Kotovsky K, Schunn C, Wood K (2013) The meaning of "near" and "far": the impact of structuring design databases and the effect of distance of analogy on design output. J Mech Des 135(2):021007

Grace K, Maher ML, Fisher D, Brady K (2015) Modeling expectation for evaluating surprise in design creativity. In: design computing and cognition'14. Springer, Berlin, pp 189–206

Guimera R, Uzzi B, Spiro J, Amaral LAN (2005) Team assembly mechanisms determine collaboration network structure and team performance. Science 308(5722):697–702

Hall BH, Jaffe AB, Trajtenberg M (2000) Market value and patent citations: a first look. National Bureau of Economic Research

Hall BH, Jaffe AB, Trajtenberg M (2001) The NBER patent citation data file: lessons, insights and methodological tools. National Bureau of Economic Research

Harhoff D, Narin F, Scherer FM, Vopel K (1999) Citation frequency and the value of patented inventions. Rev Econ Stat 81(3):511–515

Kaufman JC, Baer J (2004) Hawking's haiku, Madonna's math: why it is hard to be creative in every room of the house
Kuhn TS (2012) The structure of scientific revolutions. University of Chicago Press, Chicago
Newell A, Shaw JC, Simon HA (1959) The processes of creative thinking. Rand Corporation, Santa Monica
Nickerson JV (2015) Collective design: remixing and visibility. In: Design computing and cognition'14. Springer, Berlin, pp 263–276
Oman SK, Tumer IY, Wood K, Seepersad C (2013) A comparison of creativity and innovation metrics and sample validation through in-class design projects. Res Eng Des 24(1):65–92
Plucker JA, Beghetto RA (2004) Why creativity is domain general, why it looks domain specific, and why the distinction does not matter
Prabir S, Amaresh C (2007) Development of a method for assessing design creativity. Guidelines for a Decision Support Method Adapted to NPD Processes
Simonton DK (1999) Creativity as blind variation and selective retention: Is the creative process Darwinian? Psychol Inq 10(4):309–328
Simonton DK (2000) Creativity: cognitive, personal, developmental, and social aspects. Am Psychol 55(1):151
Simonton DK (2004) Creativity as a constrained stochastic process
Sternberg RJ, Lubart TI (1999) The concept of creativity: prospects and paradigms. Handb Creativity 1:3–15
Trajtenberg M (1990) A penny for your quotes: patent citations and the value of innovations. Rand J Econ 172–187
Uzzi B, Mukherjee S, Stringer M, Jones B (2013) Atypical combinations and scientific impact. Science 342(6157):468–472
Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393 (6684):440–442
Weisberg RW (1993) Creativity: beyond the myth of genius. WH Freeman, New York
Weisberg RW (2006) Creativity: understanding innovation in problem solving, science, invention, and the arts. Wiley, Hoboken

Characterizing Tangible Interaction During a Creative Combination Task

Mary Lou Maher, Lina Lee, John S. Gero, Rongrong Yu and Timothy Clausner

Abstract Tangible user interfaces change the way we interact with digital information, with physical affordances that are distinctly different from pointing and keyboard/mouse interaction. As a precursor to studying the impact of tangible interfaces on design cognition, this paper presents a coding scheme for measuring the difference between two types of user interfaces: tangible and pointing. We perform a case study, using data collected from an experiment in which participants are asked to make word combinations from a set of six nouns and give them meaning. The task is presented as a design task with references to function, behavior, and structure of the word combination meanings. The case study shows large differences in gesture and action between the two conditions. We conclude with hypotheses on how interaction modalities that afford more body movement may have an impact on creativity and design cognition.

Introduction

The affordances of design tools facilitate specific aspects of designing. As we move away from the traditional WIMP (Windows, Icons, Menus, and Pointer) interaction, we encounter new kinds of affordances in interactive digital design tools (Burlamaqui and Dong 2015). In this paper we focus on how to characterize the distinction between tangible and pointing interactions, as a precursor to studying the influence of tangible interaction on design cognition.

Tangible user interfaces (TUIs) are a type of human computer interaction design based on graspable physical objects, that shift the sorts of actions required for

M.L. Maher (✉) · L. Lee · J.S. Gero · R. Yu
University of North Carolina at Charlotte, Charlotte, NC, USA
e-mail: MaryLou.Maher@gmail.com

T. Clausner
University of Maryland, College Park, MD, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_3

Fig. 1 Sifteo™ tangible user interface cubes

interacting with digital information from pointing and clicking to holding, grasping and moving physical objects. TUIs are the coupling of physical objects and digital information, and eliminate the distinction between input and output devices, such as mouse and display (Fitzmaurice et al. 1995). For example, Fig. 1 illustrates Sifteo™ cubes, a type of TUI. Tangibles can trigger various gestures (spontaneous gestures or intentional actions), and have potential for exploring design alternatives through novel forms of interacting and discovering.

Previous studies of tangible user interfaces have shown an effect on designers' cognition during a design task (Fjeld et al. 1998; Kim and Maher 2008; Brereton and McGarry 2000). Kim and Maher (2008) found an increase of epistemic actions, and through a protocol analysis were able to observe an increase in the cognitive processes typically associated with creative design. The potential affordances of the TUIs, such as manipulability and physical arrangements, may reduce cognitive load associated with spatial reasoning, thus resulting in enhanced spatial cognition and creative cognition. Brereton and McGarry (2000) studied the role of objects in supporting design thinking as a precursor to designing tangible interaction; they found that design thinking is dependent on gesturing with objects, and recommend that the design of tangible devices consider a tradeoff between exploiting the ambiguous and varied affordances of specific physical objects.

In this paper, we study how graspable tangible devices differ from pointing, in a creative task of combining words and giving the combination a meaning. Ultimately, our goal is to study how interfaces based on physical objects (i.e. TUIs) engage human cognition differently than traditional computer interfaces that do not include grasping within a design context. We claim that the developments in user interaction toward tangible devices have significant implications for computational support for designers. In this first step, we want to better understand how graspable and pointing interactions differ. We describe the results from a case study carried out to better understand how to measure the differences in gesture and action in an existing corpus of experimental data. We show a difference in the presence of gesture and action between the TUI and control conditions; while this was an expected result of the experiment, here we are reporting a case study on how to segment and code the data.

Gesture and Thought

Gesture has been associated with thought and speech production. There is evidence that gesturing with our hands promotes learning (Cook et al. 2008), and aids problem solving (Trofatter et al. 2015), but few studies have explored actions with objects (Fitzmaurice et al. 1995; Fjeld et al. 1998) and none have compared tangible object and intangible interaction within a creativity context.

Research on gesture in psychology and cognitive science considers the different types of gestures and the roles that gestures play in communication, problem solving, and learning. McNeill (1992) describes four gesture categories when gesture is associated with speech, that he adapted from Efron (1941) and others: "iconics", when the gesture depicts an object or event; "metaphorics", when the gesture presents an image of an abstract concept such as knowledge or language; "deictics", when the gesture is a pointing movement; and "beats", hand actions linked to speech rhythm. These categories were developed assuming that the person is speaking to another person, and suggest that gesture is relevant to language and therefore important for communication.

There is evidence that gesturing aids thinking. When children are learning to count, the learning is facilitated by touching physical objects (Efron 1941; Kessell and Tversky 2006). More specifically, Kessell and Tversky (2006) show that when people are solving and then explaining spatial insight problems, gesturing facilitates finding solutions. Goldin-Meadow and Beilock (2010) summarize findings as "gesture influences thought by linking it to action", and "producing gesture changes thought" and can "create new knowledge". These studies show that gesture, while originally associated with communication, is also related to thinking. Tangible interaction design creates an environment that encourages actions on objects, and therefore affords more gestures and actions than traditional GUIs. We posit that the affordances of graspable tangible devices, when present during a creative task, may affect cognition.

Case Study

We use a case study approach as an early exploration of the differences between tangible interaction and pointing interaction in a collaborative task (Casakin and Badke-Schaub 2015). In this paper we present a subset of the data from an experiment designed to study the effect of tangible interaction on creative cognition (Clausner et al. 2015). The experiment design uses a conceptual combination task, which for the purposes of this study is similar to a design task: a synthesis of prescribed components (words) and the creation of a meaning for selected combinations. The conceptual combination task in this experiment is an adaptation of Wisniewski and Gentner (1991), for the purpose of investigating tangible interaction and its effect in a creative cognition task.

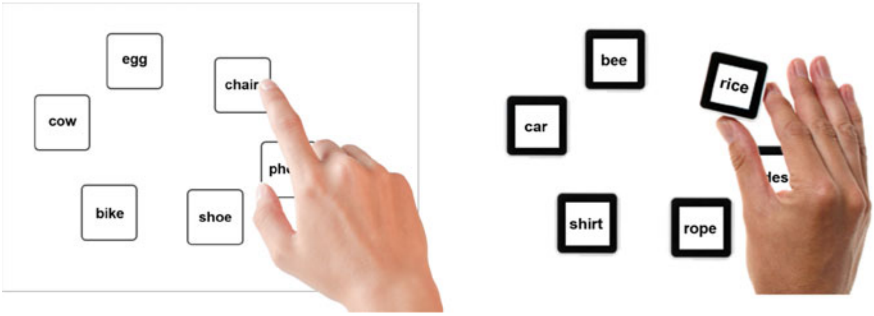


Fig. 2 Schematic of stimuli

In the experiment the participants are asked to combine words from a given set of 6 words, and then describe meanings for the combined words. In the instructions the participants are asked to think about the function, behavior, and structure of the combined word when creating its meaning. The visual display features of the words were maintained across the two conditions, but varied across conditions in terms of the affordances for interacting with the words: words were displayed on tangible user interface cubes (Cubes Condition, Fig. 2 right) or printed on a poster board (Poster Condition, Fig. 2 left).

Pairs of participants worked together in the conceptual combination task (Pauwels et al. 2015). A within-subjects experimental design enabled comparison of the graspable interaction when compared to pointing for each pair of participants.

Participants

Forty 6th-grade children (aged 11–12) participated in the experiment, providing data for 20 pairs of participants. For our case study, we arbitrarily selected results from 5 pairs of participants. We chose 6th grade children because of their ability to compose creative meanings while engaging them in a task that would be received as age-appropriate.

Words as Design Elements

Considering the conceptual combination task as a design task, the words provided the requirements of the design, and the task is similar to a design synthesis problem. The two word sets comprised six words each. Each word is a single-syllable noun representing a concrete basic-level category object (Clausner et al. 2015). Each

Table 1 Word set stimuli by semantic category

Category	Word set 1	Word set 2
Animal	Cow	Bee
Artifact-tool	Phone	Rope
Clothes	Shoe	Shirt
Food	Egg	Rice
Furniture	Chair	Desk
Vehicle	Bike	Car

noun represented one of six disparate semantic categories: food, furniture, tool, clothes, vehicle, and animal, shown in Table 1. This prevented word combinations from forming same-category meanings, thus promoting creative thinking. Each set of six nouns was constructed such that the 30 possible pair-wise combinations (e.g. rice desk) would not form conventional lexicalized N–N compounds.

Materials

Six Sifteo cubes were programmed to display one word per cube. The display did not change, and no other cube sensors or capabilities were active. The display screen of Sifteo cubes is housed in a square frame with rounded corners. The printed poster stimuli were designed to match cube displays on task relevant perceptual attributes: text font and size; words appeared centered in a rounded square; and initial spatial arrangement of the cubes and printed squares matched.

Procedures

The within-subjects procedure consisted of an instruction phase followed by two experiment conditions: Poster Condition and Cubes Condition in two counterbalanced blocks. The two Word Sets were counterbalanced with condition type (Poster, Cubes) and block order (Block 1, Block 2). The duration of the experiment was approximately 20 min, consisting of the instruction phase followed by two experimental blocks of 5 min each. Participants were instructed to "combine words and come up with as many creative meanings as you can". The instructions included an example: "the word fish and the word car are things everyone knows about, but nobody knows about a fish car". Three questions based on function, structure, and behavior (Gero 1990) encouraged creative thinking: Who can tell me what a fish car might look like?—what a fish car is for, or what it does?—how a fish car works? In the experimental phase each participant pair sat at a table with a poster paper (Poster Condition) or Sifteo cubes (Cubes Condition) (Fig. 3).

Fig. 3 Experimental design. *Left* cubes condition/*Right* poster condition

Participants self-selected their choices of word combinations and how they took turns presenting their creative ideas to their paired partner. Experimental sessions were video and audio recorded.

Exploring Hypotheses for Characterizing the Effect of Graspable Interaction on Design Creativity

In this case study we explore methods of measuring the difference in gesture and action between the two interaction conditions, in ways that are relevant to previous studies on creative cognition, a core component of creative design. As a means to defining relevant measurements, we present potential hypotheses on the impact of tangible interaction on creative cognition.

Epistemic Actions Improve Creativity

We are interested in the argument that interfaces based on physical objects may offer more opportunities for epistemic (i.e. exploratory) actions than pragmatic (i.e. direct) actions (Kirsh and Maglio 1994). Epistemic actions are exploratory motor activity aimed at uncovering information that is hard to compute mentally. An example of epistemic action is the way novice chess players find it helpful to physically move a chess piece when thinking about possible consequences. Epistemic actions offload some internal cognitive resources into the external world by using physical actions (Maher et al. 2014). In contrast, pragmatic actions are motor activity directly aimed at a final goal. If tangible interactions offer greater opportunity for epistemic actions, then they may improve creativity by affording mental exploration through physical action (Fjeld et al. 1998). This leads to hypothesis H1.

- **H1**: Tangible interaction increases epistemic actions when compared to pointing interaction.

Fluid Movement Leads to Creativity

We expect that bodily movement can influence cognitive processing, with fluid movement leading to fluid thinking. Slepian et al. (2014) explain the effect of fluid body movement on creativity. Creative thought is often contrasted with analytical thought: creative thought is associated with a more relaxed, open and playful approach where analytical thought is more rigid and precise; a fluid can move in multiple directions with ease, and the ability to fluently and flexibly generate multiple thoughts is essential for creativity. Fluid thinking, thus, is a metaphor for certain elements of creativity (Hofstadter 1995). This leads to hypothesis H2.

- **H2**: Tangible interaction encourages more fluid body movement than pointing interaction.

Collaboration Drives Creativity

Tangible interactions enable the development of sharable interfaces that provide a unique collaboration environment (Xie et al. 2008). Tangible interactions can impact children's collaboration and facilitate engagement and motivation that increase attention to tasks. We expect that the higher the interest in the experiment, the more active and collaborative their behaviors, which is expected to enhance the ability to think of creative and novel meanings. This leads to hypothesis H3.

- **H3**: Tangible interaction draws more collaborative actions than pointing interaction.

Abstract Concepts Enhance Creativity

Abstract concepts may help people solve problems in more creative ways, and encourage people to think "outside of the box." (Sternberg and Lubart 1999). We have expectations that the participants might engage in more abstract thinking when they explain function-behavior, and this abstract thinking will lead to creative meanings. This leads to hypothesis H4.

- **H4**: Tangible interaction elicits more function and behavior exploration than pointing interaction.

Bimanual Interaction Facilitates Creativity

We propose that bi-manual tangible interaction may improve creative output, because manipulating cubes with two hands may facilitate interhemispheric interaction (Bevans 2011). This leads to hypothesis H5.

- **H5**: Tangible interaction induces use of both hands.

Data Analysis for Case Study

Segmentation and Coding

Our analysis of the video stream for each session involves segmenting the video into discrete elements defined by a start time and end time, and assigning a code to each segment. We started by segmenting the verbal stream according to speaker, and then segmenting it into smaller segments so that a segment is formed around the utterance of a word combination or around the definition of a word combination. In order to associate the gestures and actions with design issues, we had an additional stage of segmentation in which each segment is associated with one “FBS” code using the FBS coding scheme described below. Segmentation and FBS coding were done simultaneously. We then coded each segment using a gesture/action coding scheme described below. We coded a gesture for each of the two children’s left and right hand for each FBS segment. Two of the authors performed the coding together for 3 of the sessions, to ensure agreement on the coding scheme. Then a single author coded all sessions twice, separated by a period of several days, followed by an arbitration process to identify and resolve differences in coding.

FBS Coding Scheme

Function-Behavior-Structure (FBS) is a schema for analyzing design activity. The definitions of FBS (Gero 1990) originally proposed are listed in Table 2.

Table 2 The FBS definitions by Gero (1990)

Function	The intentions or purposes of the design artifact
Behavior	How the structure of an artifact achieves its functions
Structure	The components which make up an artifact and their relationships

When instructing the participants to create a meaning for word combinations, they are asked to describe its function, behavior, and structure.

- R: Requirements. This is when the participant makes a verbal reference to one of the six words printed on the poster or cubes.
- F: Function. This is when the participant talks about the purpose, use, or function when describing the meaning of the word combination.
- Be: Expected behavior. This is when the participant talks about an expected behavior in the meaning of the word combination.
- Bs: Behavior from structure. This is when the participant talks about whether the structure in the meaning of the word combination can actually achieve the expected behavior.
- S: Structure. This is when the participant talks about the appearance, the form, the spatial qualities, and the material properties of the meaning of the word combination.
- O: Other. This is used when a participant repeats a phrase or talks about something that is not relevant to the task.

For example: shirt car (coded as requirements), the participants talk about how it behaves: "you wear" (coded as expected behavior), what it looks like: "car made out of shirt" (coded as structure), and what it is for: "that you can drive" (coded as function).

In addition to the FBS codes, we coded when the F, B, or S segment introduces a new or surprising idea for that pair of participants. The meaning of surprising is derived from the distinction between novel and surprising in (Maher 2010; Grace et al. 2015). Surprising ideas are unexpected issues associated with the function, behavior or structure of the meanings the participants ascribed to the word combination. For example, when explaining chair egg in a particular session, Child 2 said "a chair and an egg what if there is like a whole cracked up egg and you can just sleep on it like a vampire or just like be in the egg." A vampire in this context is unexpected and was coded as surprising.

Gesture/Action Coding Scheme

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body with the intent of conveying meaningful information or interacting with the environment. Gestures vary in form, function, and in how they relate to language. Bodily experiences play an integral role in human cognition (McNeill 1992; Goldin-Meadow and Beilock 2010). Gestures help people link words to the world (e.g. deictic gestures) or help a person organize spatial information into speech (e.g. iconic or metaphoric gestures). When we code the gestures in this case study, we distinguish between gestures and actions where actions are body movements associated with the intention to hold or move a physical object. Our intention in developing this coding scheme is to explore

Table 3 Gesture/action coding scheme

G	Grasping
P	Pointing
OG	Other gesture
NG	No gesture
NV	Not visible

potential correlations between the gesture or action, and the cognitive issue (a cognitive issue is F, B, or S). For example, does a grasping gesture correlate with thinking about structure, or thinking about behavior? We coded gestures and actions in five categories, shown in Table 3.

Although the tangible cubes afford many different actions (grasping, rotating, stacking, picking up, and rearranging), we coded all actions on the cubes as Grasping. This is distinctive to the Cubes condition and allowed us to compare the impact of the words on cubes to words on the poster. When the participant used a finger or hand to point to a word on a cube or poster, we coded that segment as Pointing. All other body movement was coded as Other Gesture (OG). An example of OG is when a participant said "it happened over and over again" while gesturing by repeatedly tracing a circle with one hand. If the participant's hand was on the table but not moving, it was coded as No Gesture (NG), and if a hand was below the table so that it could not be seen, it was coded as Not Visible (NV).

Coding Analysis

In this paper, we report on the coding and analysis of 5 of the 20 pairs of participants for a total of 10 sessions. We are using this smaller set of data to explore our hypotheses about the differences in body movement in the two conditions, and as the basis for developing hypotheses for a more complete analysis of the entire corpus of data. Our notation to refer to the session is pair number-condition, e.g. P01-POSTER is the pair of participants labeled P01, and the condition is the poster condition.

Figure 4 shows an overview of the coded data for one pair of participants. Each bar in the figure is one of the FBS segments, plus NT which we used to code a time segment in which the participants were not talking. Each code is indicated in a different color. This pair is typical of all 5 pairs: Grasping dominated in the Cubes condition and Not Visible dominated in the Poster condition.

Table 4 shows the percentage of new words and surprising words for each session along with the Total Segment number and which condition was first in the Order column. In all cases, there were more FBS segments in the Poster condition than in the Cubes condition, yet the number of new and surprising words is higher in the Cubes condition in 3 of the 5 participant pairs in this case study.

Table 5 shows the total number of gesture segments as a percentage for each pair in each condition, indicating whether the Poster or Cubes condition was larger. For

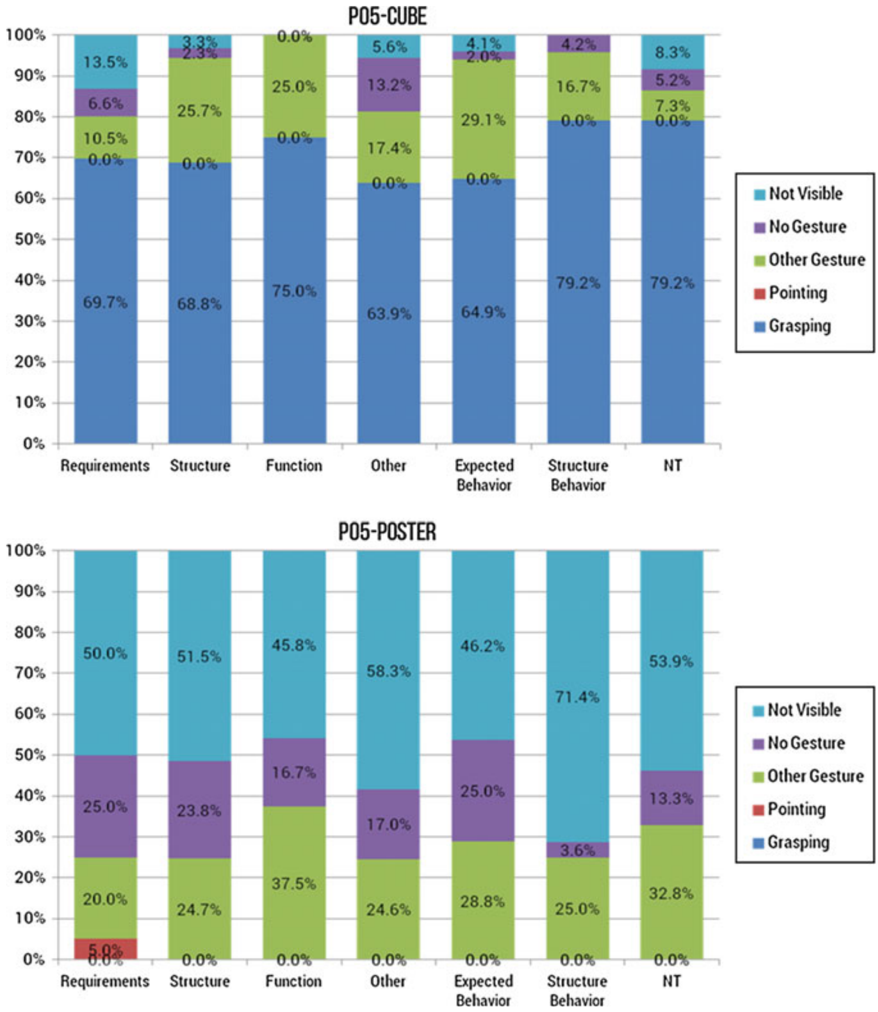


Fig. 4 Gestures on FBS codes. *Top* P05 cube condition, *Bottom* P05 poster condition

example, the NV and P code in the poster condition is larger than the NV and P code in the cubes condition for all 5 pairs, which is displayed in red column. And the G code is dominated in the cubes condition for all 5 pairs, which is displayed in blue column.

We observed the actions performed with the cubes, such as: making word combinations by changing the arrangement of cubes, holding the cubes without meaning, and actions performed to make word combinations by adding or taking out cubes from word combinations. We observed gestures not related to actions on the cubes, such as: gestures to explain meanings of a word combination, gestures to describe the behavior or the appearance of a word combination, gestures to repeat a

Table 4 New and surprising words in poster and cube condition

	Order	Poster					Cubes				
		Total segment	New (%)		Surprising (%)		Total segment	New (%)		Surprising (%)	
P01	Cubes	205	9.8		1.5		143	18.2	▲	2.8	▲
P02	Poster	194	13.9		2.6		180	16.1	▲	2.8	▲
P05	Poster	243	25.5		4.5		200	28.5	▲	6.5	▲
P04	Poster	205	24.9	▲	4.4	▲	194	21.1		1.5	
P03	Cubes	228	34.6	▲	9.2	▲	176	31.8		6.8	

▲ indicates the higher value in the two conditions

Table 5 Gesture percentages in poster and cube condition

		G (%)		P (%)		OG (%)		NG (%)		NV (%)		Total (Count)
Poster	**P01**	0		2.6	▲	7.9		11.1		78.4	▲	820
	P02	0		1.2	▲	7.5		1.4		89.9	▲	776
	P03	0		3.2	▲	52.2	▲	16.9	▲	27.7	▲	912
	P04	0		3.0	▲	13.9		4.8		78.3	▲	820
	P05	0		0.2	▲	26.5	▲	20.1	▲	53.2	▲	972
Cubes	**P01**	37.3	▲	0.7		17.0	▲	31.7	▲	13.3		571
	P02	51.7	▲	1.0		8.9	▲	12.1	▲	26.4		720
	P03	37.4	▲	0.3		40.9		12.5		8.9		704
	P04	56.4	▲	0.9		16.5	▲	16.4	▲	9.8		776
	P05	68.8	▲	0		20.9		5.0		5.3		796
G: Grasping, P: Pointing, OG: Other Gesture, NG: No Gesture, NV: Not Visible												

▲ indicates the higher value in the two conditions

certain pattern, and habitual stroking of the hair and patting around the lips. In all cases, the cubes condition had more gestures + pointing + action and the poster condition had more no gesture + not visible. This means that the participants exhibited a relatively higher proportion of using their hands in the cubes condition than the poster condition.

Table 6 shows percentages of segments based on FBS code, with an indication of when the percentage is greater in the cubes or the poster condition. For example, the segments coded R (Requirements) refer to the task to be performed. Thus, the number of segments coded R is correlated with the number of word combinations the pair came up with during the session. The remaining segments code when the pair explained meaning(s). All of the total number of requirement segments as a percentage, for each pair in cubes condition is larger than poster condition. This means that cube condition produces more word combinations per total number of segments, which is displayed in the blue column in Table 6.

Table 6 FBS segment percentages in poster and cube condition

		R (%)		S (%)		F (%)		O (%)		Be (%)		Bs (%)		NT (%)		Total (Count)
Poster	**P01**	19.5		52.7	▲	2.4		8.3		11.2	▲	0.5		5.4		205
	P02	17.0		20.6		0.5		39.7	▲	7.2	▲	6.2	▲	8.8		194
	P03	5.3		44.3	▲	3.9		17.1		23.2	▲	6.1		0.0		228
	P04	13.2		46.3		3.9	▲	15.6	▲	11.7		4.4	▲	4.9		205
	P05	4.1		34.2		2.5	▲	27.2	▲	16.0		2.9		13.2	▲	243
Cubes	**P01**	25.9	▲	45.5		3.5	▲	8.4	▲	4.2		3.5	▲	9.1	▲	143
	P02	21.1	▲	23.9	▲	1.7	▲	28.9		5.6		4.4		14.4	▲	180
	P03	15.9	▲	33.0		4.5	▲	19.9	▲	20.5		6.3	▲	0.0		176
	P04	16.5	▲	47.9	▲	3.6		11.9		11.9	▲	0.0		8.2	▲	194
	P05	9.5	▲	38.0	▲	0.5		18.5		18.5	▲	3.0	▲	12.0		200

R: Requirements, S: Structure, F:Function, O:Others, Be: Expected Behavior, Bs: Behavior from Structure, NT: No talking

▲ indicates the higher value in the two conditions

Not Talking (NT) is not typically included in the FBS coding scheme, but is in this case study. It is meaningful to code NT because the participants were gesturing or grasping while not talking. Figure 4 shows percentages for gestures based on FBS. It shows a high portion of gestures using cubes during 'no talking' time. This result can be used as a clue to how a series of gestures which appear while they do not talk, develop thoughts in performing design tasks and what effect they have on cognition.

Analysis of Table 6 shows that, of all FBS codes, structure has the highest proportion. Structure issues address the object(s) and their relationships while explaining the meaning of word combinations. Table 6 also shows that in all 5 cases, the percentage of Requirements segments is always higher in the Cubes condition.

H1: Tangible interaction increases epistemic actions when compared to pointing interaction.

Epistemic actions are counted as the number of segments in which the participant is pointing at a word or grasping a cube. As seen in Table 7, the percentage of segments coded as epistemic actions is higher in cubes condition than poster condition for all 5 pairs, which is displayed in the blue column. In the case of P05, epistemic actions (grasping) are 59.7% of the segments in the cubes condition. In the cubes condition, participants are changing or realigning the positions of the cubes but in poster condition, percentage of epistemic action (pointing) is negligible at 0.2%. Participants rarely took pointing actions.

Table 7 Epistemic actions in poster and cube condition

	Poster				Cubes			
	Pointing		Total (%)		Grasping		Total (%)	
	C1 (Count)	C2 (Count)			C1 (Count)	C2 (Count)		
P01	0	17	2.1		51	137	33.3	▲
P02	9	0	1.2		96	206	41.9	▲
P03	16	13	3.2		98	165	37.4	▲
P04	25	0	3.0		188	214	51.8	▲
P05	0	2	0.2		259	216	59.7	▲

▲ indicates the higher value in the two conditions

H2: Tangible interaction encourages more fluid body movement than pointing interaction.

Where H1 focuses on epistemic actions, considering only those segments in which the participants are searching the words for a word combination to give meaning, H2 focuses on body movements during all phases of the design task. Hence all the gestures with the cube and other gestures were included in this comparison. There is diverse range of fluid body movements. Movements made with cubes show continuously repeating patterns of behavior. For example, there is the pattern of back-and-forth movement on the surface on the desk with a cube held in one hand, or the pattern of keeping on turning the cube in one direction.

Overall, the percentage of segments with body movements is higher in cube condition than poster condition. In Table 8, P03 and P05 show the highest proportion of body movements in both conditions. When compared with Table 4, P03 and P05 expressed more surprising meanings. Though the data is not sufficient to

Table 8 Body movement in poster and cube condition (number of segments)

	Poster						Cubes					
	P		OG		Total (%)		G		OG		Total (%)	
	C1 (Count)	C2 (Count)	C1 (Count)	C2 (Count)			C1 (Count)	C2 (Count)	C1 (Count)	C2 (Count)		
P01	0	22	20	45	10.6		54	157	52	96	62.9	▲
P02	9	0	3	55	8.6		119	253	25	64	64.0	▲
P03	16	14	184	292	55.5		98	165	133	288	97.2	▲
P04	25	0	76	38	17.0		207	231	83	128	83.6	▲
P05	0	2	163	95	26.7		300	252	60	166	97.7	▲

▲ indicates the higher value in the two conditions say this increase in body movement and increase in new and surprising ideas are related, it highlights that this is a hypothesis to explore further.

H3: Tangible interaction encourages more collaborative actions than pointing interaction.

In order to determine if the tangible objects create a more collaborative environment in performing tasks, we counted how many segments a participant speaks before being interrupted by the other participant in each condition. It was assumed that if the two participants speak alternately, it means that they cooperate to make developments by sharing thoughts to perform the tasks, and that both participants are engaged in the activity.

In Table 9, the number for each participant in the row labeled 1 is a count of how many times that participant spoke for 1 segment before the other participant spoke, row 2 is how many times that participant spoke for 2 segments before the other participant spoke, and so on. We indicate when the cube condition is greater than the poster. Our analysis shows that subjects generally spoke more times alternately in the poster condition. In Table 9, by this analysis, P03 is the most collaborative. In the cube condition, child 1 spoke 33 times and child 2 spoke 24 times while in the poster condition, child 1 spoke 41 times and child 2 spoke 27 times.

The percentage of abstract concepts for each pair under each condition is shown in Table 10. For participants P03 and P05 function and behavior segments were a higher percentage of the segments when explaining meanings compared to the other

Table 9 Hold the floor in cube and poster condition

Poster																				
Seg	P01				P02				P03				P04				P05			
	C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)	
1	24	▲	20	▲	31	▲	20	▲	41	▲	27	▲	29	▲	33	▲	23	▲	24	▲
2	4		6	▲	8	▲	7	▲	3		13		6	▲	1		9	▲	6	
3	1		1	▲	2		9	▲	1		4		1		3	▲	6	▲	2	
4	-		1		-		5	▲	-		2	▲	1		1	▲	-		-	
Cubes																				
Seg	P01				P02				P03				P04				P05			
	C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)		C1 (Count)		C2 (Count)	
1	13		17		27		18		33		24		23		17		14		9	
2	6	▲	3		6		6		7	▲	13		4		8	▲	8		8	▲
3	1		0		2		5		-		4		3	▲	1		2		2	
4	1	▲	1		-		4		-		-		1		-		-		2	▲

▲ indicates the higher value in the two condition

Table 10 Abstract concept in poster and cube condition

	Poster				Cubes			
	F (%)	B (%)	Total		F (%)	B (%)	Total	
P01	2.4	11.7	14.1	▲	3.5	7.7	11.2	
P02	0.5	13.4	13.9	▲	1.7	10.0	11.7	
P03	3.9	29.3	33.2	▲	4.5	26.8	31.3	
P04	3.9	16.1	20.0	▲	3.6	11.9	15.5	
P05	2.5	18.9	21.4		0.5	21.5	22.0	▲

▲ indicates the higher value in the two conditions

pairs of participants. When compared with Table 4, participants expressed more surprising meanings in P03 and P05.

We observed that there may be a correlation in the participants explaining function or behavior in the segments before and after coming up with a surprising meaning. For example, in the word combination car rice, a concrete representation refers to the shape and color or something that describes structure related to its most common use. An abstract representation refers to car rice as a food like "if you are hungry, you can eat the car while you are driving, they cheap too because they're edible." This meaning is about function and behavior. These more abstract thoughts might lead the participants to contemplate other, less common uses or structure descriptions.

H4: Tangible interaction elicits more function and behavior exploration than pointing interaction.

H5: Tangible interaction induces use of both hands.

We measured bimanual interaction in two ways: alternate uses of each hand and simultaneous usage of both hands. Table 11 shows how many times hand gestures were used for each participant. In order to investigate whether participants used both hands when making explorative gestures, we counted the number of segments with gasping and pointing. As there were no pointing or grasping gestures in P01 child 1, P05 child 1, P02 child 2 and P04 child 2, they were excluded from comparison. In the remaining sessions, the number of times when both hands are alternately used was larger in the cubes condition. In the poster condition, participants did not use pointing gestures at all except in P03, and even when using pointing gestures they used only one hand.

Table 12 shows the number of segments for each child where the LH and RH are both coded as G, P, OG. Bimanual interaction occurred more frequently in the cubes condition than the poster condition.

Table 11 Bimanual interaction, number of times, in poster and cube condition

	Poster						Cubes					
	P						G					
	C1		Index[a]	C2		Index[a]	C1		Index[a]	C2		Index[a]
	RH (Count)	LH (Count)		RH (Count)	LH (Count)		RH (Count)	LH (Count)		RH (Count)	LH (Count)	
P01	0	0	–	22	0	0	49	5	0.1	89	68	0.7
P02	0	9	0	0	0	–	100	19	0.2	114	139	0.8
P03	4	12	0.25	8	5	0.6	65	33	0.5	73	92	0.7
P04	0	25	0	0	0	–	116	91	0.8	118	113	0.9
P05	0	0	–	2	0	0	151	149	0.9	142	110	0.7

[a]Bimanual index: 0 if they only used 1 hand and 1 if they used both hands an equal amount of time. Index = smaller divided by larger

Table 12 Bimanual interaction, number of segments, in poster and cube condition

	Poster				Cubes			
	C1	C2	Total		C1	C2	Total	
P01	4	4	8		13	80	93	▲
P02	0	16	16		25	119	144	▲
P03	53	127	180		78	151	229	▲
P04	14	11	25		108	116	224	▲
P05	21	24	45		169	166	335	▲

▲ indicates the higher value in the two conditions

Discussion and Future Work

This paper presents a coding scheme used in a case study, to measure the influence of the grasping affordances present in tangible user interfaces, in the context of a design task. Our gesture/action coding scheme connects gesture to the segmentation and coding of the verbal utterances of FBS issues. Our coding scheme includes Grasping, Pointing, Other Gesture, No Gesture, and Not Visible. The case study reported in this paper shows that the cubes condition served as an environment that has more gesture and action than the poster condition. Although this is to be expected given the different affordances in the two conditions, our coding scheme identifies a way to measure this difference in a design context.

Using our gesture/action coding scheme applied to 5 case participants, we measured the differences between actions in the cubes condition (tangible interaction with the design elements) and poster condition (no tangible interaction) to explore 5 hypotheses, derived from the literature on creativity, that relate body movement and creativity. Of the 5 hypotheses, our case study shows that the 3 that are directly related to body movement (H1, H2, and H5) should be further explored. The two hypotheses that relate to collaboration (H3) and abstract thinking (H4) are not strongly associated with tangible interaction when compared to pointing interaction. Specifically, our case study supports further analysis of the following hypotheses; alternately, non-tangible interaction my have a negative influence on these hypothesized factors.

1. Tangible interaction encourages more epistemic actions.
2. Tangible interaction encourages more fluid body movement.
3. Tangible interaction encourages the use of both hands.

This exploration of the data and hypotheses is a starting point for understanding the influence of TUIs on gesture and action in a design task, with the potential for developing similar studies that show the impact of this kind of body movement on creativity and design. Our study results are consistent with the idea that participants produce more gestures when allowed to manually manipulate physical objects.

The contributions of this paper include: a coding scheme specifically for gesture and action in a design task, and the exploration of hypotheses relating gesture and action to creativity. While the affordances of different kinds of user interfaces are taken into consideration in designing new interaction technologies (Kim and Maher 2008), few studies consider their impact on cognition (Cook et al. 2008; Trofatter et al. 2015; McNeill 1992). In this study we focused on characterizing TUI affordances on cognitive design issues (FBS), that ultimately can inform the design and analysis of new TUI technologies and applications. Currently we are coding and analyzing the full set of experimental participant data in two coding scheme methods: FBS (as in this paper) and psycholinguistics and semantics (Clausner et al. 2015). In our future work, we plan to apply the gesture/action codes to the remainder of the participants in the experimental data, and to further explore hypotheses that posit correlations between our gesture/action codes and cognitive issues in designing.

Acknowledgements This research was funded by NFS Grant No. IIS-1218160 to M. L. Maher, T. Clausner, and A. Druin. The author contributions of this paper are: Clausner led the design of the experiment, which yielded data for coding and analysis both by cognitive scientific methods (in preparation), and the FBS coding and analysis (this paper) were led by Maher and Gero, with assistance by Lee and Yu.

References

Bevans A (2011) Investigating the effects of bimanual multi touch interaction on creativity. In: Proceedings of the 8th ACM conference on creativity and cognition, pp 451–452

Brereton M, McGarry B (2000) An observational study of how objects support engineering design thinking and communication: implications for the design of tangible media. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 217–224

Burlamaqui L, Dong A (2015) The use and misuse of the concept of affordance. In: Gero JS, Hanna S (eds) Design computing and cognition'14. Springer, Berlin, pp 298–311

Casakin H, Badke-Schaub P (2015) Mental models and creativity in engineering and architectural design teams. In: Gero JS, Hanna S (eds) Design computing and cognition'14. Springer, Berlin, pp 155–171

Clausner TC, Maher ML, Gonzolez A (2015) Conceptual combination modulated by action using tangible computers. Poster presented at The 37th annual meeting of the Cognitive Science Society, Pasadena

Cook SW, Mitchell Z, Goldin-Meadow S (2008) Gesturing makes learning last. Cognition 106(2):1047–1058

Efron D (1941) Gesture and environment. King's Crown Press, Oxford, England

Fitzmaurice GW, Ishii H, Buxton WAS (1995) Bricks: laying the foundations for graspable user interfaces. In: The proceedings of CHI, pp 442–449

Fjeld M, Bichsel M, Rauterberg M (1998) BUILD-IT: an intuitive design tool based on direct object manipulation. Gesture and sign language in human–computer interaction. Springer, Berlin, pp 297–308

Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36

Goldin-Meadow S, Beilock SL (2010) Action's influence on thought: the case of gesture. Perspect Psychol Sci 5(6):664–674

Grace K, Maher ML, Fisher D et al (2015) Modeling expectation for evaluating surprise in design creativity. In: Gero JS, Hanna S (eds) Design computing and cognition'14. Springer, Berlin, pp 189–206

Hofstadter DR (1995) Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought. Basic Books, New York

Kessell A, Tversky B (2006) Using diagrams and gestures to think and talk about insight problems. In: Proceedings of the meeting of the cognitive science society

Kim MJ, Maher ML (2008) The impact of tangible user interfaces on designers' spatial cognition. HCI 23(2):101–137

Kirsh D, Maglio P (1994) On distinguishing epistemic from pragmatic action. Cogn Sci 18(4):513–549

Maher ML (2010) Evaluating creativity in humans, computers, and collectively intelligent systems, DESIRE'10: creativity and innovation in design. Aurhus, Denmark

Maher ML, Clausner TC, Gonzalez B, Grace K (2014) Gesture in the crossroads of HCI and creative cognition. In: Proceedings of CHI 2014 workshop on gesture-based interaction design: communication and cognition

McNeill D (1992) Hand and mind: what gestures reveal about thought. University of Chicago Press, Chicago

Pauwels P, Strobbe T, Derboven J et al (2015) Conversation and critique within the architectural design process: a linkograph analysis. In: Gero JS, Hanna S (eds) Design computing and cognition'14. Springer, Berlin, pp 135–152

Slepian ML, Weisbuch M, Pauker K, Bastian B, Ambady N (2014) Fluid movement and fluid social cognition bodily movement influences essentialist thought. Pers Soc Psychol Bull 40(1):111–120

Sternberg RJ, Lubart TI (1999) The concept of creativity: prospects and paradigms. Handb Creat 1:3–15

Trofatter C, Kontra C, Beilock S, Goldin-Meadow S (2015) Gesturing has a larger impact on problem-solving than action, even when action is accompanied by words. Lang Cogn Neurosci 30(3):251–260

Wisniewski EJ, Gentner D (1991) On the combinatorial semantics of noun pairs: minor and major adjustments to meaning. In: Simpson GB (ed) Understanding word and sentence. North Holland, Amsterdam, pp 241–284

Xie L, Antle AN, Motamedi N (2008) Are tangibles more fun?: comparing children's enjoyment and engagement using physical, graphical and tangible user interfaces. In: Proceedings of the 2nd international conference on Tangible and embedded interaction, pp 191–198

Dissecting Creativity: How Dissection Virtuality, Analogical Distance, and Product Complexity Impact Creativity and Self-Efficacy

E.M. Starkey, A.S. McKay, S.T. Hunter and S.R. Miller

Abstract Product dissection has been adopted in engineering education as a means to benchmark existing products and inspire new design ideas. Despite widespread adoption of dissection practices, however, little is known about the effectiveness of the widely varying approaches of dissection for encouraging creativity. Therefore, the purpose of this study was to identify the impact of dissection virtuality, analogical distance, and product complexity on creativity and self-efficacy through a factorial experiment with 30 engineering students. The results show that virtual dissection can be used to increase student creativity over physical dissection but this increase is moderated by the complexity and analogical distance of the product being dissected. CSE was not significantly impacted in the study. These results are used to derive implications for dissection practices in engineering education and drive future research that explores the multifaceted role of analogical distance in example-based design practices.

Introduction

Product dissection—the systematic disassembly and analysis of a product and all of it's parts—has become a staple in engineering design courses due to its ability to help students understand (Sheppard 1992), benchmark (Lamancusa et al. 1996), and redesign (Grantham et al. 2010; Toh and Miller 2013) existing products. Although recent work have shown that student participation in team-based product dissection activities increases novelty (Toh et al. 2012) and student design variety over other forms of physical product interactions (Toh and Miller 2014), limited attention has been given to the varying approaches to deploying product dissection activities. For example, while virtual dissection has increased in popularity in recent years due to costs associated with physical dissection practices e.g., material costs and space requirements (Borrego et al. 2010; Devendorf et al. 2010) and the scalability

E.M. Starkey · A.S. McKay · S.T. Hunter · S.R. Miller (✉)
The Pennsylvania State University, University Park, PA, USA
e-mail: scarlettmiller@psu.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_4

problem associated with physical dissection (Doyle et al. 2011; McKenna et al. 2008), limited research has been conducted on the impact of dissection virtuality on student creativity. Universal application without systematic assessment of dissection variants is problematic because we do not know which forms of product dissection promote or inhibit creative idea generation.

Exploring the impact of the virtuality of dissection is particularly important because of the reduced time and effort required to virtually dissect complex products may actually be beneficial for design creativity. For example, previous research on the Sunk Cost effect has suggested that the time and effort a designer invests in a design activity can significantly impact their creative abilities during idea generation (Viswanathan and Linsey 2011). In addition, research in psychology suggests that the medium of interaction can lead to variations in cognitive load requirements and working memory limitations (Ayres et al. 2003), which is important given research findings that have linked idea generation activities to short-term and long-term memory structures (Finke et al. 1992). However, the impact of the virtuality of the product dissection activities on design creativity has yet to be explored. Thus, it is open to question how virtual dissection activities help or hinder engineering creativity.

In addition to the virtuality of dissection, the way a product is selected for dissection varies between classrooms. This is an important factor to explore given research that has revealed that the type of example used during the design process can impact an examples ability to inspire or constrain design thinking (Sio et al. 2015; Agogué et al. 2014). Specifically, researchers have identified that the analogical distance between the example product and the design domain can impact design creativity, with analogically far examples (Campbell 1960) and surface dissimilar examples (Wilson et al. 2010) being the most beneficial for helping designers develop original ideas. The complexity of the product dissected may also impact idea generation ability following a dissection task. Complex products require more cognitive effort to analyze and understand (Ayres et al. 2003), which places greater strain on memory structures. While it makes sense that a stimulus further from the target would lead to a more creative solution, this is not always the case with stimulus too far from the desired space (Fu et al. 2013). In addition, previous research has largely explored the impact of 2D, relatively simple design examples outside of a dissection context leading to questions of how analogical distance and product complexity impact creativity during dissection activities of 3D products.

Finally, research in engineering design has largely focused on how interventions like product dissection impact design creativity through idea generation activities. Few studies have explored how these interventions impact creative self-efficacy (CSE), or “one’s belief in their ability to produce creative alternatives” (Tierney and Farmer 2002, pg. 1138). This is important given research in psychology that has related CSE to creative success (e.g., Beghetto 2006; Choi 2004; Gong et al. 2009; Tierney and Farmer 2011). Additionally, Bandura (1997) noted that CSE might develop through product dissection activities. Thus, understanding how CSE is impacted through design activities like product dissection may provide an idea of how these activities impact creativity overall, rather than in respect to a specific design task. However, no studies have been performed in engineering design.

The purpose of the current study was to explore the effects of product dissection on engineering design creativity and CSE. Specifically, we sought to examine the effects of dissection virtuality, analogical distance, and product complexity on student creativity and CSE through an experimental study. While analogical distance, virtuality and complexity have been investigated previously; the interaction of these factors has not been explored. The results of this study are used to derive implications for dissection practices and drive future research that looks at the complexity of the relationship between example-based design practices and creativity in engineering education.

Related Work

Although not studied in the context of product dissection, there have been several studies exploring the impact of design examples and self-efficacy gains on design creativity. This section serves to summarize the main findings in these research areas and provide a framework for the current study.

Impact of Examples on Design Creativity

Investigations on the impact of product dissection on creativity has been limited, yet there has been extensive research conducted that has explored the impact of design examples, or known solutions to the design problem (e.g. existing products), on design creativity. This research has shown that these example products can serve as a 'jumping off' point and inspiration for designers (Herring et al. 2009). However, researchers have also found that viewing examples of prior solutions to a problem can cause design fixation, or a "blind and sometimes counter-productive adherence to a limited set of ideas in the design process" (Jansson and Smith 1991, p. 4). While many studies have shown that examples can cause fixation effects, (see for example Pertulla and Sipila 2007; Marsh et al. 1999; Linsey et al. 2010), a recent meta-analysis revealed that the type of example presented can impact the novelty and quality of an individuals creative output (Sio et al. 2015). In order to understand how to mitigate the constraining effects of examples and maximize their inspirational impact, researchers have devoted much time and attention to identifying the impact of example types selected and the type of interactivity a designer has with the example on design creativity.

Specifically, analogical reasoning has been studied in the context of creativity due to its ability to highlight important commonalities, identify obstacles, and suggest new ways to characterize the problem domain (Gentner and Markman 1997). In analogical transfer, the distance between the source (e.g., example product) and target (e.g., design domain) can be conceptualized as ranging over a continuum from far-field, where there are little to no surface shared features with the target domain, to near-field, where there are a significant number of similar surface features (Chan et al. 2011). Although increases in the superficial similarity between near-field analogies make them easier to access (Gentner et al. 1993), researchers have reported that far-field analogies (Campbell 1960) and surface dissimilar examples (Wilson et al. 2010) are most beneficial for helping designers develop novel ideas. Similar effects have been found on exploratory (N = 8) investigations of product dissection; dissecting analogically far products resulted in increases in design novelty compared to analogically near products (Toh and Miller 2013). This study also found that quality was not impacted by analogical distance (Toh and Miller 2013). Importantly, researchers have also found that analogically distant examples can help designers minimize fixation effects (Campbell 1960; Simonton 1999).

While this research highlights that distantly related examples may support design creativity, the complexity of the product being dissected may impact this relationship due to cognitive load requirements (Sweller et al. 2011). Cognitive load refers to the amount of working memory required to process information (Mayer 2005). Increases in cognitive load have been attributed to the length or complexity of an activity (Mayer 2005; Schweppe and Rummer 2014; Mayer and Chandler 2001; Tabbers et al. 2004), as well as how closely the content matches procedural-motor knowledge (Höffler and Leutner 2007). Specifically, this load can vary based on the complexity of the content to-be-learned (intrinsic load) and the cognitive demands imposed by the way the information is presented (extraneous load). When a complex task is completed (e.g. complex product dissection), intrinsic load may be increased due to the reduction in mental schema development (Van Merriënboer et al. 2006) and increase in mental resource requirements (Feinberg and Murphy 2000), which may interfere with idea generation (Diehl and Stroebe 1991). On the other hand, the way material is presented (e.g. virtual versus physical dissection) can impact extraneous cognitive load, providing a positive, or negative, impact on learning (Feinberg and Murphy 2000).

Creative Self-Efficacy

In addition to exploring the impact of dissection practices on design creativity, we also explored its impact on creative self-efficacy (CSE). Importantly, CSE—a belief in ones creative ability—has been linked to creative success (Beghetto 2006; Gong et al. 2009). The mechanisms that efficacy operate are straightforward—individuals who believe they are capable of successfully generating novel ideas are the most likely to engage, persist, and seek out the potentially risky acts of creativity (Beghetto 2006). This is important because creative ideas, by definition, are those that differ from those that came before them and have the potential to be met with criticism as well as being most likely to fail given their untested nature (Hunter and Cushenbery 2014).

Although creative self-efficacy is linked to creative outcomes, precisely how to encourage CSE is less clear given that work here has been somewhat limited. Fortunately, in the few studies that have examined interventions aimed at improving CSE, there is indication that is possible to improve CSE with targeted approaches. For example, Mathisen and Bronnick (2009) utilized three samples including undergraduate students and employees receiving either a five-day or one-day training course, and found that CSE increased for all groups. Similarly, Robbins and Kegley (2010) found that using an online creativity task (i.e., tinkertoy software) increased CSE. This study highlights that CSE can be improved in short-term interventions and that virtual approaches show promise as tools for developing CSE. Finally, although not studied in the context of CSE, prior work on CSE has suggested that the complexity of the dissection activity may increase CSE. Specifically, prior work has found that task complexity is an important factor in self-efficacy (see for example Bandura 1986) as complex tasks typically require higher cognitive facilities of the task performer (Bandura 1997).

Taken together, the above studies suggested that CSE is malleable. Specifically, it identifies that short-term interventions may be used to improve CSE. It also indicates that the virtuality of the dissection activity may increase CSE due to differences in the engagement of the activity. However, no study to date has explored these effects. Thus, we sought to identify how CSE is impacted by product dissection activities.

Research Design and Methodology

In light of this prior work, the purpose of the current study was to explore the effects of product dissection on creativity and CSE in an engineering classroom. Specifically our 2-h experimental study was developed to understand how dissection virtuality (physical and virtual), analogical distance (near and far), and product complexity (simple and complex) impacts participant: (1) Design Creativity and (2) CSE. Specifically, our hypotheses were:

1. *Design Creativity will be significantly higher in the virtual dissection condition compared to the physical condition*. This hypothesis is based on prior work that has shown that has shown that sunk cost plays an important role in design fixation (Viswanathan and Linsey 2011). Since virtual dissection indirectly expedites the dissection process by reducing the time and effort required to dissect complex products in comparison to physically, the reduced ‘cost’ of virtual dissection may serve to support creativity.
2. *Design Creativity will be significantly higher in the analogically far condition than in the analogically near condition*. However, this relationship may be moderated by the complexity of the product dissected with simple products

increasing design creativity. This hypothesis is based on prior work that has shown that the use of analogically far examples result in increases in design novelty (Chan et al. 2011), and research that has related task complexity to cognitive load (Mayer 2005; Schweppe and Rummer 2014; Mayer and Chandler 2001; Tabbers et al. 2004).

3. *CSE will be significantly improved following the dissection activity.* However, CSE will be significantly higher in the dissection of the simpler products. This hypothesis is based on prior research that has shown that short and active interventions (1-day) can improve undergraduate student CSE (Beghetto 2006; Mathisen and Bronnick 2009) and research in self-efficacy that has identified the importance of task complexity (Stajkovic and Luthans 1998).

The remainder of this section outlines the procedures followed for our experimental investigation aimed at addressing these research questions.

Participants

Participants were recruited from a first-year undergraduate engineering design course at a large Northeastern university. The course used in the study was an introductory course that has received numerous national awards for its ability to encourage hands-on engagement of first-year students through two in-depth design projects throughout the semester. In all, 30 students (22 males and 8 females) participated in the study.

Procedure

At the start of the experiment, a brief overview was provided and informed consent was obtained. Participants then completed a CSE survey developed by Tierney and Farmer (2002). After completing the survey, participants were introduced to the goal of the study through the following design prompt that was read allowed to participants as they followed along on provided papers:

> Upper management has put your team in charge of developing a concept for a new innovative product that froths milk in a short amount of time. Frothed milk is a pourable, virtually liquid foam that tastes rich and sweet. It is an ingredient in many coffee beverages, especially espresso-based coffee drinks (Lattes, Cappuccinos, Mochas). Frothed milk is made by incorporating very small air bubbles throughout the entire body of the milk through some form of vigorous motion. The design you develop should be able to be used by the consumer with minimal instruction. It will be up to the board of directors to determine if your project will be carried on into production.

Participants were also provided with samples of frothed milk so they could see the consistency and appearance of the liquid. The design task chosen in the current study was selected to represent a typical project in a first year engineering design course, or corner stone class. In these courses, students typically redesign small, electro-mechanical consumer products that are equally familiar, or unfamiliar, to the students (Simpson and Thevenot 2007; Simpson et al. 2007). To confirm our participants had equal familiarity with the product being explored, we performed pilot testing with first-year students prior to this study (Toh and Miller 2015).

Next, participants were randomly assigned to one of eight experimental conditions in the 2 (dissection virtuality) × 2 (analogical distance) × 2 (product complexity) factorial design experiment (see "Experimental Design"). Participants were then introduced to the purpose and goals of the product dissection activity. Specifically, participants were instructed that:

> Product dissection is often done in industry and academia to uncover opportunities for re-design. Designers take apart and analyze all components of a product to understand its structure and properties, and thus, find ways to improve the product. Therefore, the goal of dissection is to improve the functionality, maintainability, and reliability of a product through the examination, study, capture, and modification of existing products.
>
> During this activity, you will perform a product dissection on the provided product by taking it apart and analyzing the function of each component. Your goal is to understand strengths and weaknesses of the product in order to develop new innovative concepts that satisfy the design goal.

They were then asked to complete their assigned dissection activity for 30 min using tools such as screwdrivers, pliers, and table clamps for the physical condition and SolidWorks eDrawings for the virtual condition. Specifically, during the dissection activity participants were asked to take apart the product until they could not take it apart anymore, analyze the function of each component, complete a bill of materials, and draw a functional layout diagram, as is typically done following dissection in engineering design (Doyle et al. 2011), see Fig. 1. Participants were also instructed to use the full 30 min allotted for the dissection activity and to continue adding detail to their functional layout diagram until the activity ended.

After completing of the dissection activity, participants completed a 20-min brainstorming activity for the task described above. The participants were instructed that the individual with the most creative ideas would receive a $10 gift certificate. Once the idea generation session was over a post-study CSE survey was administered.

Experimental Design

The study was of a 2 (dissection virtuality) × 2 (analogical distance) × 2 (product complexity) factorial design, and participants were randomly assigned to a condition prior to the study. The levels are described below.

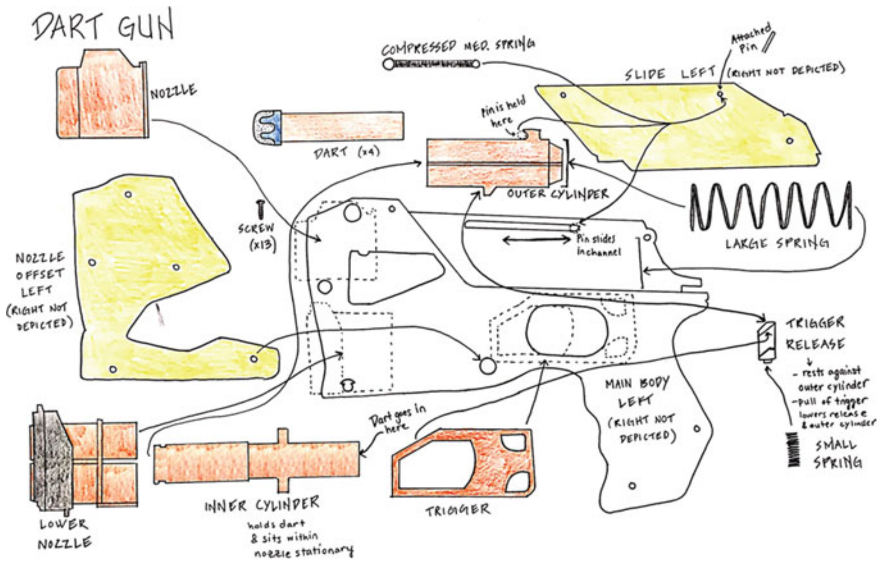


Fig. 1 A sample functional layout diagram of the SharpShot Nerf Gun (complex, far)

Dissection Virtuality

Participants were instructed to dissect each product either physically (N = 16), using tools such as pliers and screwdrivers, or virtually (N = 14) using an animated exploded view of a detailed 3D model (see Table 1). The virtual dissection activity was completed in SolidWorks eDrawings 2015 (64 bit edition version 15.4.0.0012) on A Dell computer with Intel core i5 3.33 GHz CPU and 8 GB of ram running 64 bit Windows 7 on service pack 1. The Computer was outfitted with a 17 Dell monitor (1280 × 1024 resolution, part number c2jmk), a Dell USB scroll 3-button optical mouse (part number 0xn967), and a Dell Quiet Keys USB keyboard (part number KB1421). While students were shown the full capabilities of the application through a video (section views, part transparency, exploded view, etc.), they were not specifically instructed which tools to use during the dissection activity and thus each student used the tools they felt most appropriate for the dissection task.

Product Analogical Distance

Participants were provided with a product that was either analogically near (N = 16) or far (N = 14) from a milk-frothing product (the design goal). To ensure the internal and construct validity (Campbell and Fiske 1959) of analogical distance as a manipulated variable, three steps were taken. First, a team of subject matter experts (two faculty members and two PhD students with at least two publications in related work) generated a list of potential products that were perceived as

Table 1 Products dissected in the experimental study

		Complexity	
		Simple	Complex
Analogical Distance	Near	Oral-B 3D white power Toothbrush (battery operat-ed)  # parts – 16	Proctor Silex Mixer (Model: 62509RY)  # parts – 43
	Far	Tombow Mono Correction Tape hybrid style  # parts - 9	Sharp Shot Nerf Gun (Model: 38123)  # parts – 31

differing in analogical distance from the design task along three dimensions: general appearance, mechanical movement, and use and application. The team discussed these products until consensus was reached on a final list of 10 possible handheld electromechanical and mechanical products. These products were then given to a sample of 10 engineering students along with pictures and descriptions of the products (e.g. French press, electric drill, spray bottle). Specifically, participants were provided with pairs of the ten products and asked to rate perceived analogical distance between the products on a 5-point Likert scale from "very dissimilar in [e.g. mechanical movement]" to very "very similar in [e.g. mechanical movement]", as has been done in previous research (Lopez et al. 2011).

Importantly, these ratings of analogical distance were completed along three dimensions: general appearance, mechanical movement, and use and application. Of specific interest to the current study were the pairwise comparisons between the milk frother product and the nine remaining products. These data were evaluated once again by the team of subject matter experts and used to select the four products deemed analogically near and far from the milk frothing domain, see Table 1 for product list.

Product Complexity

Participants were provided with a simple (N = 15) or complex (N = 15) product to dissect. The complexity of the products was identified through the same pilot survey used for analogical distance. However, this time on the survey, participants were provided with each of the 10 products and were asked to rate them on a 5-point Likert scale from "very low in complexity" to "very high in complexity". The four products selected from this data represented high levels (complex) and low levels of perceived complexity (simple) that were also analogically near and far in nature, see Table 1.

Metrics

Several metrics were used to compare the creativity and self-efficacy during the study. These metrics are described in detail below.

Design Creativity

The ideas developed were evaluated for creativity using Amabile's (1996) Consensual Assessment Technique. Three quasi-expert raters (Kaufman et al. 2013; Kaufman and Baer 2012) with a background in creativity studies (Baer et al. 2009) independently rated the ideas generated using a 6-point scale (very uncreative to very creative). It should be noted that creativity was defined to the raters as an idea that was both novel and useful (Runco and Jaeger 2012; Sternberg 1999). However, studies have shown that while raters often consider both novelty and usefulness when rating overall creativity, they often place a larger emphasis on the idea's novelty (Diedrich et al. 2015). Thus, novelty is a first-order criterion for creativity and usefulness is a second-order criterion. Importantly, using this methodology there was high agreement among the raters for creativity, $\alpha = 0.85$. Creativity ratings were aggregated across the three raters and then were averaged across all of the ideas generated by each person following recommendations by Silvia (2011).

Creative Self-Efficacy

Differences in CSE were measured by comparing responses from the 3-item pre- and post-study survey developed by Tierney and Farmer (2002). Specifically, students were asked to provide responses on items such as "I am good at coming up with new ideas". Potential ratings ranged from 0 (low self-efficacy) to 100 (high self-efficacy) and were used to compare changes in students' CSE among the dissection conditions.

Results

During the study, 153 ideas were developed by participants with an average of 5 ideas developed by each participant (SD = 2.40). These ideas ranged in creativity scores from 1.33 to 5.5 with a mean score of 2.86 (SD = 1.09). The average change in CSE during the study was 0.27 (SD = 1.45). The remainder of this section outlines our results with reference to our hypotheses. These results present one between-person ANOVA and one mixed-design ANOVA using SPSS v. 22 and a significance level of 0.05. For all analyses, pairwise comparisons using least significant difference tests and simple main effects are reported.

Design Creativity

Our first hypothesis was that virtual dissection would significantly increase idea creativity compared to physical dissection. In addition, our second hypothesis was that dissecting analogically far products would increase design creativity but this increase would be moderated by the complexity of the product. In order to test these hypotheses, a three-way between-person ANOVA was calculated to examine differences in the creativity of ideas generated, see Table 2 for means and standard deviations. The dependent variable of this analysis was idea creativity and the independent variables were Product Complexity (Simple, Complex), Analogical Distance (Near, Far), and Dissection Virtuality (Physical, Virtual). The results revealed a significant main effect for Virtuality, $F(1, 23) = 5.11$, $p = 0.03$, $\eta^2 = 0.18$. The remaining two main effects were not significant, $Fs < 2.7$, $ps > 0.11$.

There was a significant Complexity × Virtuality interaction effect, $F(1, 23) = 5.06$, $p = 0.03$, $\eta^2 = 0.18$ and a marginally significant Complexity × Analogical Distance interaction effect, $F(1, 23) = 4.19$, $p = 0.052$, $\eta^2 = 0.15$. The Analogical Distance × Virtuality interaction effect was nonsignificant, $F(1, 23) = 3.05$, $p = 0.09$, $\eta^2 = 0.12$.

Table 2 Means and standard deviations for idea creativity

Analogical distance	Virtuality	Complexity	
		Simplex	Complex
Near	Physical	$2.38\ (0.67)_{a1}$	$2.70\ (1.21)_{a1}$
	Virtual	$2.30\ (0.63)_{a1}$	$3.08\ (0.35)_{a1}$
Far	Physical	$2.05\ (0.56)_{a1}$	$2.95\ (1.21)_{a1}$
	Virtual	$4.80\ (0.79)_{a2}$	$2.57\ (0.73)_{b1}$

N = 30. Simple main tests were used to compare mean differences across the eight experimental conditions. Different subscript letters indicate a significant difference in means on the horizontal at $p < 0.05$. Different subscript numbers indicate a significant difference in means on the vertical at $p < 0.05$

Notably, the three-way Complexity × Analogical Distance × Virtuality interaction effect was significant, $F(1, 23) = 9.20$, $p = 0.006$, $\eta^2 = 0.29$. When analogical distance was far and the method of dissection was virtual, creativity was significantly greater when complexity was simple (M = 4.80, SE = 0.41) compared to when it was complex (M = 2.57, SE = 0.48), $F(1, 23) = 12.61$, $p = 0.002$, $\eta^2 = 0.35$. When analogical distance was far and complexity was simple, creativity was significantly greater when the method of dissection was virtual (M = 4.80, SE = 0.41) compared to when it was physical (M = 2.05, SE = 0.41), $F(1, 23) = 22.46$, $p < 0.001$, $\eta^2 = 0.50$. When complexity was simple and the method of dissection was virtual, creativity was significantly greater when analogical distance was far (M = 4.80, SE = 0.41) compared to when it was near (M = 2.30, SE = 0.41), $F(1, 23) = 18.61$, $p < 0.001$, $\eta^2 = 0.45$. The remaining pairwise comparisons were significant. These results support our hypothesis that while dissecting products virtually can increase student creativity over physical dissection, this change in idea creativity is moderated by the analogical distance and complexity of the product being dissected.

Creative Self-Efficacy

Our second research question sought to understand how CSE was impacted by our experimental factors. We hypothesize that CSE would be significantly improved following dissection activities, but the complexity of the dissection activity would impact CSE gains. In order to test this hypothesis, a four-way mixed design ANOVA was calculated to examine changes in creative self-efficacy. The within-person variable was CSE from the pre- and post-survey and the between-person variables were Complexity (Simple, Complex), Analogical Distance (Near, Far), and Virtuality (Physical, Virtual). See Table 3 for means and

Table 3 Means and standard deviations for creativity self-efficacy

Complexity	Analogical distance	Virtuality	Pre-assessment	Post-assessment
Simple	Near	Physical	$7.75\ (1.77)_a$	$8.17\ (1.84)_a$
		Virtual	$5.83\ (2.19)_a$	$7.00\ (0.72)_a$
	Far	Physical	$7.42\ (0.50)_a$	$7.75\ (0.96)_a$
		Virtual	$6.89\ (1.17)_a$	$7.17\ (1.89)_a$
Complex	Near	Physical	$6.67\ (1.05)_a$	$6.42\ (1.07)_a$
		Virtual	$4.92\ (1.64)_a$	$4.83\ (3.38)_a$
	Far	Physical	$5.67\ (1.56)_a$	$5.00\ (2.76)_a$
		Virtual	$5.89\ (3.02)_a$	$6.56\ (3.20)_a$

N = 30. Simple main effect tests were used to compare mean differences across the experimental conditions with an emphasis on pre- and post-test assessment of CSE. Different subscript letters indicate a significant difference on the horizontal at $p < 0.05$

standard deviations. The ANOVA results revealed a significant main effect for Complexity with CSE being significant higher in the simple condition ($M = 7.25$, $SE = 0.47$) than it was in the complex condition ($M = 5.74$, $SE = 0.47$). The remaining main effects, two-way interactions, three-way interactions, and the four way interaction were nonsignificant, Fs < 1.81, ps > 0.19.

Discussion

The main goal of this study was to explore the effects of product dissection on idea creativity and creative self-efficacy in an engineering classroom. The main findings from our experiment were as follows:

- Students in the virtual dissection condition created ideas that were more creative than students in the physical dissection condition
- Students who dissected simple products were more creative when the product was also analogically far, regardless of dissection virtuality
- Student CSE did not significantly increase or decrease from pre to post dissection

These results are discussed in the following sections in relation to our research hypotheses.

The Impact of Dissection Virtuality on Design Creativity

The results of this study support our hypothesis that creativity would be significantly higher in the virtual dissection condition compared to the physical condition. This finding may be related to the sunk cost affect associated with dissecting products physically; since physical dissection takes more time and effort to complete, participants may be more likely to become fixated on the central ideas of the dissected product reducing the relative creativity of their ideas. This finding is aligned with Viswanathan and Linsey (2013) who found that sunk cost can significantly impact design fixation during idea generation activities.

While sunk cost effects may offer one possible rationale for this finding, the result may also be attributed to the way people process information during virtual dissection activities. Specifically, virtual dissection can be viewed as a more transient activity than physical dissection as students must rely on exploded views and animations of the parts as opposed to physically taking apart an analyzing how each product functions. The transient nature of virtual dissection may cause an increase in the cognitive load of the user because it requires them to process more information in their limited working memory (Chandler 2009; Wong et al. 2012). Because of this increase cognitive load, participants may not be encoding, or storing, the information in memory in as meaningful of a way as participants who physically dissected the product, leading participants to forget aspects of the product or not really understand how a part functions. This is important given research that has shown a linkage between forgetting and design creativity (Smith 1995; Smith and Blankenship 1991).

While future studies are needed to understand what facets of virtual dissection activities make it beneficial for design creativity, our study highlights its utility for aiding in creative idea generation. The use of virtual dissection activities is highly desirable due to their reduced costs (Borrego et al. 2010; Devendorf et al. 2010) and scalability (Doyle et al. 2011; McKenna et al. 2008). The results from this study indicate that virtual dissection may be better than physical dissection, the more expensive and less scalable option, for certain activities, indicating that their use is not sacrificing usability.

The Impact of Analogical Distance and Product Complexity on Design Creativity

Our results also support, in part, our second hypothesis that analogically far products would serve to inspire design creativity, but this relationship would be moderated by the complexity of the product dissected. Specifically, our results show that dissecting analogically far products increased creativity compared to analogically near products only when the dissected product was simple. These findings are supported by prior work that showed that far-field analogies (Campbell 1960) and surface dissimilar examples (Wilson et al. 2010) are most beneficial for helping designers develop original ideas. However, analogical distance in and of itself was not a significant main effect for design creativity in our study. This may be due to increases in the cognitive load required to dissect the complex product while at the same time identifying the products analogical relationship to the target domain. In other words, analogically distance, complex products required the most cognitive load from the study sample of the products dissected. This finding is in line with prior research that has shown cognitive load increases with the length or complexity of an activity (Mayer 2005; Schweppe and Rummer 2014; Mayer and Chandler 2001; Tabbers et al. 2004) and that increases in cognitive load may interfere with idea generation (Diehl and Stroebe 1991).

However, this relationship is moderated by the virtuality of the dissection activity, as evident by the significant three-way interaction. Specifically the interaction showed that participants who dissected analogically far, simple products were more creative than those who dissected analogically near, simple products in the virtual dissection condition. Again, this findings may be due to the increase in cognitive load (extraneous load) required to dissect products virtually (Feinberg and Murphy 2000).

While future work is needed to understand cognitive load requirements and its impact in dissection environments, the current study points to the impact of example complexity in idea generation tasks. Future research should be directed at understanding this relationship and uncovering how analogical distance and complexity impact cognitive load.

The Impact of Analogical Distance and Product Complexity on Creative Self-Efficacy

Our results refute our hypothesis and contradict prior work that has shown that short exercises can be used to improve CSE (Mathisen and Bronnick 2009). However, this prior work showed improvements over a 1-day rather than a 2-h period as explored in this study. Therefore, these results indicate that there may be a 'sweet spot' for seeing increases in CSE through short hands-on activities. This finding may also be impacted by the sample used in the current study, as there was a main effect of complexity that revealed that CSE was significantly higher in the simple condition than in the complex condition. In other words, students in the simple condition had a higher starting CSE than those in the complex condition calling for more research on these effects. Finally, while not studied in the context of CSE, researchers have repeatedly found that task complexity is an important factor in self-efficacy (see for example Bandura 1986) as complex tasks typically represent multifaceted constructs with different implications for information processing and cognitive facilities of the task performer (Bandura 1997). This study highlights that further investigation is needed to determine the influence of dissection on CSE.

Conclusions and Future Work

This study has shown that virtuality, analogical distance, and product complexity impact the creativity of ideas generated in a first year engineering classroom. The main contributions of this work are: (1) Virtual dissection has shown to be a superior tool to physical dissection for inspiring creativity in an engineering classroom, and (2) Analogical distance in product dissection in and of itself was shown not to impact idea creativity. Instead, this relationship was moderated by the complexity of the product and the medium of dissection.

While the current study identified important insights into the impact of dissection virtuality, analogical distance and product complexity on design creativity and self-efficacy, there were several limitations of the study. First, the study was conducted with first year engineering students due to the nature of dissection activities in the engineering classroom. While the findings can be used to provide insights into dissection practices in the engineering classroom, future work is needed to test the generalizability of these findings outside of an educational context in industry. Second, the products selected for the current study were identified based on a comparison between ten potential products in a similar manner to Lopez et al. (2011). While a pilot-study was used to identify analogical distance and perceived product complexity, future studies are needed to test the generalizability of these findings when wider variations of analogical distance and product complexity are used. In addition, the design task in the current study was an electro-mechanical task and the results may be dependent on the type of problem domain explored. Therefore, future work should look into how the problem domain impacts these factors. CSE has been linked to one's ability to be creative, choosing products for dissection such that CSE gains are higher is desirable and needs further investigation. Future work should investigate CSE in studies of differing length and with larger samples to further investigate CSE in an engineering education setting.

Acknowledgements This material is based upon work supported by the National Science Foundation under Grant No. 14630009. We would also like to thank our undergraduate research assistant Clayton Meisel for his help in this project.

References

Agogué M, Kazakçi A, Hatchuel A, Masson P, Weil B, Poirel N, Cassotti M (2014) The impact of type of examples on originality: explaining fixation and stimulation effects. J Creat Behav 48 (1):1–12

Amabile T (1996) Creativitiy in context. Westview Press, Boulder

Ayres P, Chandler P, Sweller J (2003) Expertise reversal effect. Educ Psychol 38:23–31

Baer J, Kaufman JC, Riggs M (2009) Brief report: rater-domain interactions in the consensual assessment technique. Int J Creat Probl Solving 19(2):87–92

Bandura A (1986) Social foundations of thought and action. Prentice Hall, Englewood Cliffs

Bandura A (1997) Self-efficacy: the exercise of control. Macmillan, New York

Beghetto RA (2006) Creative self-efficacy: correlates in middle and secondary students. Creat Res J 18(4):447–457

Borrego M, Froyd JE, Hall TS (2010) Diffusion of engineering education innovations: a survey of awareness and adoption rates in U.S. engineering departments. J Eng Educ 99(3):185–207

Campbell DT (1960) Blind variation and selective retentions in creative thought as in other knowledge processes. Psychol Rev 67(6):380

Campbell DT, Fiske DW (1959) Convergent and discriminant validation by the multitrait-multimethod matrix. Psychol Bull 56(2):81

Chan J, Fu K, Schunn C, Cagan J, Wood KL, Kotovsky K (2011) On the benefits and pitfalls of analogies for innovative design: ideation performance based on analogical distance, commonness, and modality of examples. J Mech Des 133(8):1–11

Chandler P (2009) Dynamic visualizations and hypermedia: beyond the "Wow" factor. Comput Hum Behav 25:389–392

Choi JN (2004) Individual and contextual predictors of creative performance: the mediating role of psychological processes. Creat Res J 16(2–3):187–199

Devendorf M, Lewis K, Simpson TW, Stone TW, Stone RB, Regil WC (2010) Evaluating the use of digital product repositories to enhance product dissection activities in the classroom. J Mech Des 9:1–8

Diedrich J, Benedek M, Jauk E, Neubauer AC (2015) Are creative ideas novel and useful? Psychol Aesthet Creat Arts 9(1):35

Diehl M, Stroebe W (1991) Productivity loss in idea-generating groups: tracking down the blocking effect. J Personal Soc Psychol 61(3):392
Doyle TE, Baetz BW, Lopes B (2011) First-year engineering bicycle dissection as an introduction to sustainable design. In: Canadian engineering education association, St. John’s
Feinberg S, Murphy M (2000) Applying cognitive load theory to the design of web-based instruction. In: Proceedings of the IEEE professional communication society international professional communication conference. IEEE Educational Activities Department, pp 353–360
Finke RA, Ward TB, Smith SM (1992) Creative cognition: theory, research and application. MIT Press, Cambridge
Fu K, Chan J, Cagan J, Kotovsky K, Schunn C, Wood K (2013) The meaning of “near” and “far”: the impact of structuring design databases and the effect of distance of analogy on design output. J Mech Des 135(2):021007
Gentner D, Markman AD (1997) Structure mapping in analogy and similarity. Am Psychol 52:45–56
Gentner D, Rattermann MJ, Forbus KD (1993) The roles of similarity in transfer: separating retrievability from inferential soundness. Cogn Psychol 25(4):524–575
Gong Y, Huang J-C, Farh J-L (2009) Employee learning orientation, transformational leadership, and employee creativity: the mediating role of employee creative self-efficacy. Acad Manag J 52(4):765–778
Grantham K, Okudan GL, Simpson TW, Ashour O (2010) A study on situated cognition: product dissection’s effect on redesign activities. In: Proceedings of the 2010 IDETC, ASME, pp 617–626
Herring SR, Chang C-C, Krantzler J, Bailey BP, Greenberg S, Hudson S, Hinkley K, RingelMorris M, Olsen D (2009) Getting inspired! understanding how and why examples are used in creative design practice. ACM CHI, Boston, pp 87–96
Höffler TN, Leutner D (2007) Instructional animation versus static pictures: a meta-analysis. Learn Instr 17(6):722–738
Hunter ST, Cushenbery L (2014) Is being a jerk necessary for originality? examining the role of disagreeableness in the sharing and utilization of original ideas. J Bus Psychol 30:621–639
Jansson D, Smith S (1991) Design fixation. Des Stud 12(1):3–11
Kaufman JC, Baer J (2012) Beyond new and appropriate: who decides what is creative? Creat Res J 24(1):83–91
Kaufman JC, Baer J, Cropley DH, Reiter-Palmon R, Sinnett S (2013) Furious activity vs. understanding: how much expertise is needed to evaluate creative work? Psychol Aesthet Creat Arts 7(4):332
Lamancusa JS, Jorgensen JE, Fridley JL (1996) Product dissection-a tool for benchmarking in the process of teaching design. In: Proceedings of the frontiers in education conference, IEEE, pp 1317–1321
Linsey J, Tseng I, Fu K, Cagan J, Wood K, Schunn C (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Mech Des 132:41003
Lopez R, Linsey JS, Smith SM (2011) Characterizing the effect of domain distance in design-by-analogy. In: Proceedings of the 2011 IDETC, ASME, pp 141–151
Marsh R, Ward T, Landau J (1999) The inadvertent use of prior knowledge in a generative cognitive task. Mem Cogn 27:94–105
Mathisen GE, Bronnick KS (2009) Creative self-efficacy: an intervention study. Int J Educ Res 48 (1):21–29
Mayer R (2005) Principles for reducing extraneous processing in multimedia learning: coherence, signaling, redundancy, spatial conguity, and tempora contiguity principles. In: Mayer RE (ed) The Cambridge handbook of multimedia learning. Cambridge University Press, Cambridge, pp 183–200
Mayer RE, Chandler P (2001) When learning is just a click away: does simple user interaction foster deeper understanding of multimedia messages? J Educ Psychol 93(2):390

McKenna AF, Chen W, Simpson T (2008) Exploring the impact of virtual and physical dissection activities on students's understanding of engineering design principles. In: ASME DETC, Brooklyn, pp 359–368
Pertulla M, Sipila P (2007) The idea exposure paradigm in design idea generation. J Eng Des 18 (1):93–102
Robbins TL, Kegley K (2010) Playing with Thinkertoys to build creative abilities through online instruction. Think Skills Creat 5(1):40–48
Runco MA, Jaeger GJ (2012) The standard definition of creativity. Creat Res J 24(1):92–96
Schweppe J, Rummer R (2014) Attention, working memory, and long-term memory in multimedia learning: an integrated perspective based on process models of working memory. Educ Psychol Rev 26(2):285–306
Sheppard SD (1992) Mechanical dissection: an experience in how things work. In: Engineering education: curriculum innovation and integration. Santa Barbara
Silvia PJ (2011) Subjective scoring of divergent thinking: examining the reliability of unusual uses, instances, and consequences tasks. Think Skills Creat 6(1):24–30
Simonton DK (1999) Origins of genius: Darwinian perspectives on creativity. Oxford University Press, Oxford
Simpson T, Thevenot H (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(1):120–130
Simpson TW, Lewis KE, Stone RB, Regli WC (2007) Using cyberinfrastructure to enhance product dissection in the classroom. IERC, Nashville
Sio UN, Kotovsky K, Cagan J (2015) Fixation or inspiration? a meta-analytic review of the role of examples on design processes. Des Stud 39:70–99
Smith SM (1995) Fixation, incubation, and insight in memory and creative thinking. Creat Cogn Approach 135:156
Smith SM, Blankenship SE (1991) Incubation and the persistence of fixation in problem solving. Am J Psychol 104:61–87
Stajkovic AD, Luthans F (1998) Self-efficacy and work-related performance: a meta-analysis. Psychol Bull 124(2):240
Sternberg R (1999) Handbook of creativity. Cambridge University Press, New York
Sweller J, Ayres P, Kalyuga S (2011) Cognitive load theory. Springer, New York
Tabbers HK, Martens RL, Merriënboer JJ (2004) Multimedia instructions and cognitive load theory: effects of modality and cueing. Br J Educ Psychol 74(1):71–81
Tierney P, Farmer SM (2002) Creative self-efficacy: its potential antecedents and relationship to creative performance. Acad Manag J 45(6):1137–1148
Tierney P, Farmer SM (2011) Creative self-efficacy development and creative performance over time. J Appl Psychol 96(2):277
Toh C, Miller SR (2013a) Product dissection or visual inspection? the impact of designer-product interactions on engineering design creativity. ASME DETC, Portland
Toh C, Miller SR (2013b) Exploring the utility of product dissection for early-phase idea generation. Portland, ASME DETC
Toh CA, Miller SR (2014) The impact of example modality and physical interactions on design creativity. J Mech Des 136(9):1–12
Toh CA, Miller SR (2015) How engineering teams select design concepts: a view through the lens of creativity. Des Stud 38:111–138
Toh C, Miller S, Okudan Kremer G (2012) The impact of product dissection activities on the novelty of design outcomes. In: ASME 2012 IDETC, Chicago
Van Merriënboer JJ, Kester L, Paas F (2006) Teaching complex rather than simple tasks: balancing intrinsic and germane load to enhance transfer of learning. Appl Cogn Psychol 20 (3):343–352
Viswanathan V, Linsey JS (2011) Design fixation in physical modelling: an investigation on the role of sunk cost. DETC, Washington

Viswanathan VK, Linsey JS (2013) Role of sunk cost in engineering idea generation: an experimental investigation. J Mech Des 135(12):121002
Wilson JO, Rosen D, Nelson BA, Yen J (2010) The effects of biological examples in idea generation. Des Stud 31(2):169–186
Wong A, Leahy W, Marcus N, Sweller J (2012) Cognitive load theory, the transient information effect and e-learning. Learn Instr 22(6):449–457

Part II
Design Cognition—Design Approaches

What Can We Learn from Autistic People About Cognitive Abilities Essential to Design? An Exploratory Study

Andy Dong and Ann Heylighen

Abstract This paper proposes to contribute to our understanding of the fundamental cognitive processes essential to design by exploring the experiences of people who have different information processing behaviors to those found in most people. In particular, we focus on people with autism spectrum conditions (ASC), because they are known to have information processing behaviors that are both maladaptive and exceptional. Central to our exploratory study is the question: what can we learn from people with ASC about cognitive processes essential to design? The scholarship on cognitive behaviors associated with the autism spectrum and narratives on the experiences with design of individuals with ASC are discussed in relation cognitive processes associated with design. We conclude that the weak central coherence theory of autism provides a useful prediction of the cognitive processes necessary for expertise in design practice.

Introduction

At least since Herbert Simon first postulated particular forms of reasoning associated with design practice (Simon 1969, 1995), design cognition has been a central theme of design studies (Chai and Xiao 2012). Scholars have been searching for the possibly unique and essential cognitive processes at the foundation of design practice, regardless of design discipline (Goel and Pirolli 1992). Part of the challenge in this search is the very definition of design. For the purposes of this paper, design is defined as the act of conceiving an object, environment, or situation for an intended purpose. Methodologically, disproving a null hypothesis about a cognitive process essential to design is nearly impossible: a hypothesis such as "Analogical reasoning is essential to design cognition" is hard to confirm, as rejecting the null

A. Dong (✉)
The University of Sydney, Sydney, Australia
e-mail: andy.dong@sydney.edu.au

A. Heylighen
KU Leuven, Leuven, Belgium

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_5

hypothesis that it is inessential would require a population of people who are impaired in this cognitive process and yet are able to design. Finding a population that lacks or is deficient in analogical reasoning is extraordinarily difficult, if not impossible.

To circumvent this methodological challenge, scholars have taken a comparative psychology approach by examining evidence of cognitive processes in nonhuman animals that may be relevant in the evolution of design cognition in humans. A review of cognitive processes found in the great apes identified representation, recursion, and curiosity as likely to be essential cognitive processes associated with the conceptual part of design cognition (Dong et al. 2017).

This paper proposes another approach: to examine evidence on the cognitive behaviors associated with people with autism spectrum conditions (ASC). People with ASC are known to exhibit both cognitive impairments and exceptional abilities for imagination. As imagination is considered the cognitive ability sine qua non for design practice, cognitive behaviors in the ASC may shed light on the cognitive processes essential to design. As cognitive scientists clarify the specific cognitive impairments, exceptional abilities, and information processing biases in ASC (e.g., Frith 2003; van der Lugt 2005), it may become possible to compare them with our understanding of design cognition to identify a set of essential cognitive processes.

The rest of the paper continues as follows. First, we briefly review the literature on design cognition to establish a provisional set of cognitive processes associated with design. We then proceed to review the literature on cognitive behaviors associated with ASC in light of the scholarship on design cognition. Narratives of individuals with ASC and their experiences with design practice are described and analyzed based upon the provisional set of cognitive processes associated with design. We conclude by discussing how this analysis helps us understand the nature of design cognition.

Cognitive Processes Associated with Design

One challenge in identifying a cognitive process associated with design is that there is no agreed upon exhaustive list of cognitive processes. Cognitive scientists propose a high-level list of cognitive processes: Attention, Perception, Learning, Remembering, Speaking, Problem-solving, Reasoning, and Thinking (Eysenck 1993). This list is not exhaustive. In addition, the terms cognitive process, cognitive function, cognitive skill, and cognitive ability are polysemous and often used interchangeably. In this paper, a cognitive process refers to an intrinsic ability of the mind. A cognitive behavior is the outward, observable manifestation of a cognitive process. A cognitive skill refers to a task-oriented ability that must be learned through observation and practice, e.g., solving a differential equation or designing a four-bar linkage. Design is likely to involve a set of cognitive processes. Design is an adaptive function of the mind's broad capability for imagination, i.e., generating

Table 1 Design cognition articles found in web of science

Journal	Articles
AI EDAM	15
Design studies	58
Journal of engineering design	4
Research in engineering design	7

novel mental representations of content that is not available to the senses (Suddendorf and Dong 2013). Like language, design is a cognitive skill that must be learned even though the cognitive processes associated with it are intrinsically available to us. As it takes time to learn a particular language, such as English, Japanese, or Hindi, so too it takes time to learn to design a four-bar linkage. The interest in this paper is to identify those cognitive processes essential to designing independent of the object of design. Cognitive skills associated with designing specific objects and environments transcend the scope of the review.

To identify a candidate set of cognitive processes, articles published in *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (AI EDAM), *Design Studies*, *Research in Engineering Design*, and *Journal of Engineering Design*, all four considered influential journals of design research (Gemser et al. 2012), were searched using Web of Science. Articles having the topic of cognition were identified (TS = cognition). In total, 84 articles were identified, as shown in Table 1. Articles about end-user cognition, history, theory, and criticism, literature reviews, and computational simulations were excluded. This resulted in a set of 67 articles.

One author read each article and noted the cognitive processes studied therein. The list of cognitive processes was then analyzed to identify the set of most frequently studied cognitive processes. To create this set (see Table 2), a key phrase count of one to three-word phrases (with appropriate stemming) was performed over this list of cognitive processes identified. Single words counted were not allowed to be part of a phrase.

Table 2 Identified cognitive processes in design cognition articles

Cognitive process	n	Representative article	Definition
Analogical reasoning	12	Ball et al. (2004)	A reasoning process entailing the transfer of properties from a source domain to a target domain
Problem solving	11	Cross and Cross (1998)	The mental act of finding a process to create a desired goal from the present situation
Creativity	6	Kim et al. (2007)	The generation of novel ideas, objects, environments or situations
Decision making	4	Christiaans and Almendra (2010)	The cognitive selection of an object or course of action from a range of choices

(continued)

Table 2 (continued)

Cognitive process	n	Representative article	Definition
Framing	5	Kleinsmann et al. (2012)	The inclusion and exclusion of facts about an object, environment or situation to produce a schema for its interpretation
Ideation	4	Hernandez et al. (2010)	The creation of an idea for an object, environment or situation
Team mental model	4	Dong (2005)	A mental representation of an object, environment or situation that is shared by a group
Mental imagery	3	Bilda and Gero (2007)	A mental representation of the perceptual experience of an object, environment or situation
Mental simulation	3	Wiltschnig et al. (2013)	The mental modification of a mental model of an object, situation or environment
Reflecting	3	Petre (2004)	A consideration of the theories and beliefs that underlie actions
Conceptual blending	2	Nagai et al. (2009)	The integration of properties from multiple objects or situations into a single object or situation
Fixation	2	Youmans (2011)	The repeated transfer of inappropriate or irrelevant properties from an object or situation to another object or situation
Naming	2	Khaidzir and Lawson (2013)	A description for a set of important elements in a design problem that require explicit attention
Visual reasoning	2	van der Lugt (2005)	The manipulation of a mental image in order to attain desired knowledge about an object, environment or situation

Cognitive Behaviors in the Autism Spectrum

Autism—from the Greek word 'autos' meaning self—is clinically classified as a 'pervasive developmental disorder', manifesting itself in a set of behavioral characteristics, mainly with regard to social interaction and communication, and the presence of restricted, repetitive and stereotyped patterns of behavior (American Psychiatric Association 2013; Happé 1999; Wing 1997). The term was first used by Kanner (1943) and Asperger (1944) to describe a group of children, seemingly living in their own private worlds and combining a great ingenuousness with a fundamental lack of knowledge about social interaction (Delfos 2005). In the mid-1960s, well-founded theories started to explain what autism is about: a difference in information processing with a neurobiological cause (Rajendran and Mitchell 2007), involving a more consistent, more detailed, and rather literal interpretation of perceived information (Frith 2008; Noens and van Beckelaer-Onnes 2004). Today researchers talk about a much wider autism spectrum (American Psychiatric Association 2013), as autism-related conditions have been found to occur in a continuum of different forms and gradations. At one end of the spectrum, the so-called "Kanner/Asperger types" have rigid, concrete thinking patterns and definite problems with certain kinds of cognitive processes; at the other end, the so-called "low-functioning" or regressive/epileptic types, have sensory processing problems (Grandin 1995a: 138). Below we discuss cognitive behaviors associated with ASC from both diagnostic and theoretical angles.

From a Diagnostic Angle

In order to understand the characteristics associated with ASC, we start from those that are determinative in the diagnosis of autism. Note that these are not necessarily all present and do not always hold for all subgroups to the extent described here. As starting from these diagnostic determinants may sketch an extreme picture, it is important not to reduce people to their diagnosis and continue considering them in the first place as people who, just like others, have important qualities.

Although Kanner's original interpretation of the term 'autism' underwent major changes, difficulties in social interaction are still acknowledged as one of the most important, if not *the* most important aspect of autism (Davis and Carter 2014), which can express itself in a lack of *social* or *emotional reciprocity*. The diagnosis lists multiple aspects such as a lack of spontaneous urge to share with others certain experiences, good or bad, or difficulties in putting oneself in someone else's shoes, and related difficulties in being aware of someone else's emotions.

A second important characteristic associated with ASC relates to *communication* (Kim et al. 2014), both *verbal and non-verbal*. Some autistic individuals never learn to speak; others develop language only later in a sometimes-unusual way, both form- and content-wise (e.g., repeating collocutors' words or complete sentences). Some autistic people, however, speak in a very refined way or do not speak but write brilliant literary works. Other characteristics relate to difficulties in understanding directions or interpreting language non-literally. As far as non-verbal communication is concerned, visual thinking is often evident at an early age in high-functioning Kanner-type people (Park 1992). Yet, even those who consider themselves as 'visual thinkers' have difficulties in understanding body language, facial expressions, or eye-contact (Grandin 1995a).

The *limited ability to imagine* things expresses itself probably most strongly in holding on to specific–not necessarily functional–rituals or routines and resistance to change (American Psychiatric Association 2013; Ahrentzen and Steele 2009; Sánchez et al. 2011). Trivial changes, such as a small modification in a furniture arrangement or use of a new tea set, might suffice to make autistic people lose their hold and cause major stress. This category of characteristics also focuses on their intense and sometimes limited interests. Both the intensity and focus are remarkable. Interests may range from questions of origin over dissecting the concrete structure of appliances to certain movements of processes, like a toy car's turning wheel or a sliding door that opens and closes continuously.

From a Theoretical Angle

The explanatory theories of autism that have been developed offer another angle to consider the characteristics associated with ASC by going into the underlying mechanisms.

The first theories about autism situate its cause in perception (Rajendran and Mitchell 2007). An example is the *sensory theory* of autism (Delacato 1974). As suggested in Kanner's (1943) earliest reports, autistic people process sensory information in a special way, which can manifest itself in sometimes unusual reactions to stimuli–hyposensitivity, hypersensitivity, or inability to distinguish certain stimuli. According to the sensory theory, these difficulties in processing stimuli from the surrounding physical environment underlie the atypical behavior in autism (Iarocci and McDonald 2006). Rocking to and fro or biting one's fingers are explained as attempts to compensate for these differences in sensory perception (Mostafa 2007). Sometimes an autistic person has to make so much effort to process sensory information that other elements enter all the more strongly, which in turn may trigger strange behavior and frustrations. What may come across as maladjusted behavior results from an imbalance between the environment and a person's abilities to adjust to it (Sánchez et al. 2011). These problems are not unique to autism (Rajendran and Mitchell 2007), however, suggesting that they might result from a deeper problem.

In the mid-1980s, the characteristics of autism were starting to be understood as the consequence of a primary cognitive condition. As a result, several cognitive theories were developed. Most discussed is the *Theory of Mind* (ToM) hypothesis (Baron-Cohen et al. 1985), also referred to as the 'mind-blindness' theory (Lombardo and Baron-Cohen 2011; Baron-Cohen 1990). ToM is the ability to attribute to and recognize in oneself and others the entire spectrum of mental states like convictions, desires, intentions, imaginations, and emotions. This ability allows individuals to think about what is going on in their own and others' minds (Baron-Cohen 2001). According to the ToM hypothesis, autistic people develop only to a limited extent an inner theory of how people think and feel (Delfos 2005). They show impairments in representing or attributing mental states to both *self* and *other* (Lombardo and Baron-Cohen 2011; Baron-Cohen 1990). This hypothesis considerably furthered the understanding of the core social-communication impairments in ASC (Lombardo and Baron-Cohen 2011; Baron-Cohen 1990): to an autistic person, social interaction is a difficult process as it requires understanding what is going on in the other (Sánchez et al. 2011).

Another theory is that of the *planning and executive functions* (Ozonoff et al. 1991). Originally it was developed as a general theory regarding behavior occurring in people with, e.g., a brain injury. After similar characteristics had been observed in autistic people, executive functions were investigated also in the context of explaining autism (Rajendran and Mitchell 2007). These functions are defined as the capacity to hold on to a fit set of problem-solving activities and reuse it in the future to obtain a desired goal state (Welsh and Pennington 1988). This includes the intention to curb a reaction and delay it to the right moment, a strategic plan of successive actions or a mental representation of a task, including storing in memory the relevant information and desired state. These functions resemble domains such as attention, reasoning and problem-solving thinking, and differ from cognitive domains such as sensation, perception, language and memory. Together they allow for flexibility in thought and action. Within the autism spectrum it is precisely these central regulatory processes that are said to be sometimes disturbed. The behaviors are not correctly maintained by an 'executive supervisory system'. This might explain the often observed stereotypical and repetitive behavior in autistic people and their sometimes major resistance to change (Delfos 2005; Ozonoff et al. 1991).

The final theory of autism that has been thoroughly investigated is that of the *weak central coherence*. Frith (2003) explains autism by a specific imbalance in integrating information from different levels. 'Normal' information processing is characterized by an urge for 'central coherence', i.e., assembling diverse information in order to create meaning at a higher level that subsumes as many contexts as possible (Frith 2003). Autism is characterized by a weak or absent urge for this central coherence (Frith and Happé 1994). This tends to be translated as follows: "people with autism often do not see the wood for the trees. Sometimes they even do not see the trees, but they do see the grooves in the bark, the leaves and even the veins in the leaves." (Cross 2003) Therefore the theory is sometimes wrongfully reduced to the idea that autistic people are detail thinkers and cannot oversee the whole (Vermeulen 2010). In advancing this theory, however, Frith (2003) clearly distinguishes between local coherence at a low level and central coherence at a high level. Local coherence refers to seeing wholes. This occurs in an early stadium of information processing and can be done by autistic people equally well. Central coherence at a higher level addresses the need to fit information into a broader context and in this way assign meaning to the whole (Frith 2003). This is precisely where the difficulty for autistic people lies: in distinguishing details that are important and deserve attention from other details. Most people do not consider a situation detail by detail. They first try to gain a general impression of the whole, the essence, and, based thereon, decide subsequently what details are worth their attention. Autistic people, by contrast, may perceive and store all details as a kind of compensation strategy (Frith and Happé 1994) because they do not see the essence (cf. Grandin 1995a). The theory thus points at the sharp eye for some details and the difficulty in distinguishing essentials from side-issues (Vermeulen 2010). As a result difficulties arise in recognizing faces and emotions, which require uniting aspects of intonation, body language, and expression, and in interpreting out of context (Sánchez et al. 2011).

In his hypothesis about *context blindness*, Vermeulen (2010) builds on the theory of weak central coherence, in particular on the aspect of context. Context does not refer to the whole as such but to the whole of contextually relevant elements, in both environment and memory, which may also include details. "Context blindness is a deficit in the ability to use this context spontaneously and unconsciously in assigning meaning." It concerns 'not using' the context rather than 'not seeing' it. Moreover, autistic people are able to consciously use the context when someone points out to them that multiple possibilities exist: "Context blindness is a deficit in the ability to use context spontaneously and unconsciously in assigning meaning, when information is vague, unclear or ambiguous" (Vermeulen 2010). When the meaning is not directly clear, when information is unclear, vague or incomplete, context thus plays a role.

This is also important in case of 'open connections'. Research suggests that autistic people are able to see connections and reason logically. They are even good at fixed one-on-one relations whereby meaning is to be assigned to simple stimuli. When a stimulus may relate to multiple meanings, however, difficulties arise. In this kind of one-on-many relations context becomes important to decide which relations you select out of the many possibilities. Nothing of what humans observe necessarily has an absolute or fixed meaning: meanings change depending on the context. The world with all its plural and changing meanings is thus very confusing for autistic individuals. "The sense for context therefore plays a vital role in numerous cognitive functions, from observing simple objects over understanding language and human behavior to developing world knowledge with a fair share of common sense. This reduced context sensitivity leads to difficulties that are highly characteristic for autism." (Vermeulen 2010).

Experiences with Design of Individuals with ASC

So far we identified a set of cognitive processes associated with design and discussed cognitive behaviors observed in the autism spectrum. To further our understanding of which cognitive processes are essential to design, we now describe the experiences with design of two individuals with ASC. Both were selected because of (a) their diagnosis on the autism spectrum, (b) their experiences with design, and (c) the available documentation of these experiences.

The first is Temple Grandin, a professor of animal science and successful 'humane' livestock facility designer. Grandin wrote extensively about her life with autism, including her way of designing (Grandin 1995a, b, 2009). The second is Roland, who is schooled in chemical engineering, works as a self-employed consulting engineer, and is a certified inventor. Roland redesigned the interior of his house (Baumers and Heylighen 2015). Grandin was diagnosed as a child and describes herself as a Kanner/Asperger type; Roland was diagnosed with Asperger's at age 50.

In what follows, we analyze their experiences with designing based on their own accounts. In the case of Grandin, our analysis relies on her own accounts of her way of designing, which we found in several publications she authored (Grandin 1995a, b, 2009). In the case of Roland, our analysis relies on an account of his way of designing by Baumers and Heylighen (2015), which is based on two weblogs in which Roland documented his design process, a guided tour through the house under renovation, and an in-depth interview with Roland on the location of the design project. These accounts are analyzed in light of abovementioned

(a) cognitive behaviors within the autism spectrum outlined above and (b) cognitive processes associated with design to draw a connection between both.

Analogical Reasoning

When designing, Temple Grandin seems to rely heavily on analogical reasoning. To understand interactions with people during design, she compares them to something she has read or experienced: "For instance, in my equipment-design business, an argument between myself and one of my clients is similar to something I have read about the United States and Europe fighting over trading right." (Grandin 1995b) She describes the acquisition of case examples as adding to a "video" library over which she searches to find solutions to analogous problems. "Since my mind works similar to an Internet search engine, my ability to solve problems got better and better as I had more and more experiences and read more and more books and journal papers. This provided lots of images in my memory for the search engine in my mind to search." (Grandin 2009: 1438).

To some extent, analogical reasoning can be found also in Roland's design approach. The source he relies on is Alexander et al.'s book *A Pattern Language* (Alexander et al. 1977), containing 253 'problems' each of which describes a problem related to towns, buildings or construction and offers a 'solution' (Baumers and Heylighen 2015: 335). Among these patterns he identifies appropriate experiences that could be related to the interior's redesign, such as 'Sitting Circle', 'Built-in-Seats' or 'Intimacy Gradient'. "And the clever thing is", according to Roland, "that Christopher Alexander managed to establish exactly these experiences and linked them to concrete solutions." It is these solutions Roland draws upon to redesign the interior of his house.

From their recollections it is impossible to determine whether Grandin and Roland engage in mostly case-driven analogizing, invoking concrete elements of prior design solutions as the basis for the present solution, or schema-driven analogizing, drawing on the principles of prior solutions (Ball et al. 2004). Yet, their direct references to specific cases (Grandin) and concrete prior solutions (Roland) would suggest a stronger reliance on case-driven analogizing, a pattern attributed to novice designers (Ball et al. 2004).

Problem Solving

Both Grandin and Roland take linear and apparently fixed approaches to design problem solving. In drawing on her "video" library, Grandin describes her design approach as almost algorithmic. She is able to combine fixed, structural elements from previous design solutions to generate a new solution. When designing livestock equipment she takes bits and pieces of other equipment she has seen in the past and combines them to create a new system: “My mind works like the computer programs that are used to make high-tech special effects in movies; I can take many different bits and pieces and combine them into new images. I use this method when I design equipment in my livestock equipment-design business. The more bits and pieces I have in my “video library”, the better I can design equipment. I have videos of many things, such as metal posts, sheet metal, bearings, cattle, motors, gates and so on. To create a new design I pull the bits out of memory and combine them into a new piece of equipment” (Grandin 1995a).

Similarly, Roland redesigned his house’s interior by drawing on multiple design patterns (Baumers and Heylighen 2015: 335). Yet, rather than integrating properties from different patterns into a new solution, he seems to rely on one pattern at a time, redesigning and completing the interior in a gradual way: he designs something, builds it (or commissions to build it), and then proceeds with the next part of the design; there is only a rough master plan. Their descriptions of rather rigid ‘bottom-up’ design approaches contrasts sharply with the recollections of expert designers, who have been reported to take a ‘systems approach’ starting from ‘first principles’ (Cross 2003, 2004).

Mental Imagery

Grandin’s writings suggest that she can imagine herself being an animal: “I am successful in designing livestock systems because I can imagine myself as an animal, with an animal’s body shape and senses. I am able to visualize myself as an animal going through one of my systems. This “video” is complete with touch, sound, visual and olfactory sensations. I can play this “video” from two perspectives. The perspective is either I am watching the animal, or I am inside the animal’s head looking out through its eyes” (Grandin 1995a: 149–150). According to Grandin, many systems used in meat plants are designed poorly because the engineers never thought about what the equipment would feel like when it contacted the animal’s body. She attributes the fact that she does to her ability to set her own emotions aside: “I can imagine realistically what the animal would feel because I do not allow my own emotions to cloud the picture” (Grandin 1995a: 149–150). Other factors that might help explain Grandin’s complete “videos” of being an animal are her outspoken interest in the way subjects of her work see their world, evidenced by the fact that she gets in a chute herself, and her hypersensitivity to noise and other sensory stimuli herself, illustrated by her description of her roommate’s hair dryer as “a jet plane taking off” (Grandin 1995b). Her recollection suggests that she can create a mental image of herself inside a livestock system. Through this image, she develops empathy for the animals. Her ability to create this type of mental imagery is similar to humans’ general ability to imagine themselves in a previous time and place. We can imagine and play a video of ourselves walking through our childhood home as an adult today. Grandin’s recollections suggest a limited ability to imagine how something might work when she has not yet previously experienced this object. This limitation is compensated by a heightened ability to store and retrieve details of objects from prior experience and then to imagine her using or being in the object she is designing. The level of detail in Grandin's mental images may be higher than in most people's.

Mental Simulation

In redesigning the interior of his house, a major challenge for Roland was to anticipate how the design would be experienced (Baumers and Heylighen 2015: 335). Therefore he made for each partial design a mock-up (a simulation to the actual size) on site, using material at his disposal. Once this mock-up was made, he considered it important "to take the time to sit down, and to feel whether the design induces the right emotion." If this turned out to be the case, Roland drew that part or piece of furniture, and built it (or commissioned to build it). The advantage of this method, which Roland learned from Alexander, is "that you can all the time optimally experience how the design feels" (Baumers and Heylighen 2015: 335). Yet, despite building a mock-up and taking the time to experience it, Roland seemed to have difficulty imagining how it would be, as the quotes on his blog suggest: "I wonder whether it will work as I thought"; "I'm really curious what this will actually look like"; "I can't wait to see whether it will all end as I imagine it now"; "I don't think it's beautiful, but you only really know when it's finished". Striking is also that, each time a pattern had been applied, he attached a label with its number to the piece of furniture. By explicitly referring to Alexander's patterns, these labels sought to name the intended experiences to all people interacting with the interior so that he could recall their experience.

In exploring and evaluating her own design, Grandin will "run simulations under many different conditions and rotate the machine in my head. I don't need computers with fancy graphics programs, because I have a sophisticated drawing and graphics computer in my head. In my imagination I can duplicate the most sophisticated, computerized virtual-reality systems" (Grandin 1995a: 149–150). Grandin's mind works slowly, however: drawing a detailed three-dimensional drawing takes several hours. Moreover, while her account would imply a complete ability for mental simulation, her mental simulation is limited to objects similar to those she has previously experienced. Grandin seems to have difficulties simulating the operation of an object she has not yet experienced. She says she needs to experience things to store them literally in her "video" library: "To obtain a good concept of cats or churches, *I need to experience many different ones to fill up my "video" library*. There is no generalized cat concept. I can manipulate the car or church "videos". I can put snow on the roof and imagine what it would be like during different seasons" (Grandin 1995a: 142) (emphasis added). Once Grandin has a "video" of how something works, she can then replay it in her mind. Yet, when she has not personally observed the object she is designing, she struggles to understand its possible operation.

Grandin relies on images of direct experiences and Roland relies on external prototypes as strategies for understanding how a product might work. They do not report mentally simulating the mechanical and functional properties of a product by altering their mental representations of it.

Discussion

The lived experiences with design recounted by individuals with autism have important implications for the interpretation of cognitive processes associated with design. The paper started out with a putative definition of design as conceiving an object, environment, or situation for an intended purpose. Based on this definition, high-functioning individuals with ASC clearly can design. By a conventional understanding of expertise, Grandin could even be described as an expert in the area of designing livestock systems; she has designed livestock equipment across the world, has received numerous industry awards, and is one of *Time* magazine's 100 most influential people. The profiles of expertise in design (Lawson and Dorst 2009) would not neatly describe Grandin's and Roland's capabilities though. We therefore frame our discussion on what we can learn about the fundamental cognitive processes essential to design through our understanding of individuals with ASC in relation to expertise in design rather than the duality of being able/unable to design. In using the term expertise from hereon in, we will mean expertise as described in the framework proposed by Lawson and Dorst (2009) and in a number of empirical studies of expert designers.

According to Lawson and Dorst (2009) competent designers should be able to select the elements of a design problem that are relevant and devise a plan to address them. Proficient designers perform this task almost immediately. With expert designers, this task may not even be evident since the reasoning is argued to be intuitive. Grandin's design process would not appear to satisfy this criterion of design expertise. Her problem-solving strategy is purely bottom-up. "All my thinking is bottom-up instead of top-down. I find lots of little details and put them together to form concepts and theories. (…) When I solve the problem, it is not top-down and theory driven. Instead I look at how all the little pieces fit together to form a bigger picture" (Grandin 2009: 1437). Rather than presenting a design approach that is contingent upon the problem, Grandin describes an approach that appears relatively fixed. The same can be said of Roland's adherence to Alexander's patterns and concrete solutions. Expert designers tend to mix top-down and bottom-up approaches. The rigidity means neither Grandin nor Roland report any reframing of design problems, a key indicator of design expertise.

However, none of the differences in information processing behaviors associated with ASC necessarily limit their ability to design, as the narratives of Grandin and Roland portray. Instead, both Grandin and Roland have found compensatory strategies. In this regard, the weak central coherence theory of autism (Frith and Happé 1994) provides a useful prediction about the sorts of cognitive processes essential to *expertise* in design rather than design per se. This theory describes individuals with ASC as having a processing bias: using fixed procedures as the basis for solving design problems, rather than abstracted principles, may hamper the development of design expertise, as defined by the abovementioned profiles. A well-accepted axiom of design practice is that designers construct meaning from representations by engaging in a reflective conversation with the design situation (Schön and Wiggins 1992). The lack of a tendency to find higher order meaning in experiences sharply contrasts with Schön's description of design as a reflective practice and the importance of framing therein: "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" (Schön 1988: 182) (emphasis added). The ability to frame and reframe successfully is attributed though to productive design practices (Paton and Dorst 2011) found in experts (Kleinsmann et al. 2012) and is not an essential element of design cognition if by design we mean the definition stated at the start of the paper.

Second, analogical reasoning in expert designers is characterized by an ability to refer to principles underlying prior design solutions rather than their structural elements (Ball et al. 2004). Both Grandin and Roland describe that they do not generally attempt to find thematic relations in their experiences. For example, for Grandin, there is no general concept of cat, only experiences of various kinds of cats. Therefore, it is unlikely that they would engage in schema-driven analogizing. While their technique of case-driven analogizing is likened to novice designers, their narratives suggest that an inability for schema-based analogizing does not diminish their potential to create award-winning designs. Providing designers with a large and detailed database of designs can compensate for this information processing bias.

Lastly, Grandin and Roland report an inability to imagine how a design that they have never experienced before would work. Neither of them report any challenges in creating mental images of their designs though; Grandin reports having highly detailed mental images. Mental models that contain relationships between objects support mental simulation. Identifying the relevant relationships in a particular situation requires that individuals be able to reason about the principles underlying the objects; the weak central coherence theory predicts that this reasoning is impaired in individuals with ASC. Second, the actual mental simulation is intended to reduce uncertainty where gaps in information and knowledge exist. So far as we can tell from Grandin's accounts, her memory is vivid, like a video. If scenes from the video are missing in her memory, it is not clear whether she could perform the mental simulation to fill the gaps. Mental simulation in design is generally attributed to fillings gaps in knowledge about a design where exact realization through other means (e.g., sketching, prototypes, computational simulations) is impractical or impossible (Christensen and Schunn 2009). However, the compensatory strategies taken by Grandin (having more videos of existing designs) and Roland (making prototypes) suggest that mental simulation is not essential to design cognition. It might only be essential in situations of 'breakthrough' designs where prior case examples do not exist and the construction of any prototype is impractical during ideation.

In summary, in connecting the narratives of individuals with ASC to the set of cognitive processes associated with design, we find that their information processing biases with regard to certain cognitive processes associated with design cognition can be explained by the weak central coherence theory of autism. In particular, the structured design processes described by Grandin and Roland can lead to success, but their over-structured approaches are not likely to be successful under situations requiring them to start from first principles. The key to expertise in design seems to be in the flexibility of approach (Cross 2006). The design cognition scholarship has not to date attributed the inability to work from first principles to possible cognitive and neural differences. What the central coherence theory adds to our understanding about design cognition is that the mental capability to find higher-order underlying structure and principles is essential to expertise in design and is not necessarily a cognitive skill that develops through practice. It may be cognitively impaired, although the evidence from the autism literature is that it can be taught. It may also be relevant in generating ‘new to the world’ novelty by finding a novel coherence between the facts of the design situation as a new frame for the design problem. If we accept the premise that novel framing of problematic situations is foundational to design thinking (Paton and Dorst 2011), then we could conclude that central coherence is an essential cognitive ability *for design expertise* and not just an information process bias as theorized by Frith (2003).

A major weakness in the weak central coherence theory is the lack of understanding of the cognitive and neural mechanisms that underlie weak coherence (Happé and Frith 2006). At present, it is not possible to state whether central coherence is a cognitive process, a property of a number of cognitive processes, or a consequence of neural architecture. Nonetheless, our analyses indicate that finding coherence is likely to be a foundational cognitive ability associated with expertise in design.

Conclusion

In an attempt to contribute to our understanding of the fundamental cognitive processes essential to design, we explored in this paper the experiences of people who have different information processing behaviors than most people. Our study focused on people with ASC because their information processing behaviors are known to be both maladaptive and exceptional.

The profiles of expertise in design (Lawson and Dorst 2009) and studies comparing the differences between expert and novice designers do not neatly categorize Grandin’s and Roland’s capabilities though. Grandin is clearly an Expert (Lawson and Dorst 2009) given her achievements, but her design process is not necessarily situation based or strategy based, precursors to being an Expert. Roland is competent, but his reliance on explicit cases as prior examples suggests a novice design approach. These profiles of individuals with ASC who demonstrate appreciable design capabilities suggest that the framework for expertise in design and the methodology of studying expertise may require updating. Specifically, the framework may need to consider the level of outcome in addition to ways of operating in design practice.

What can we learn from individuals with ASC about cognitive processes essential to design? Analyzing narratives on their experiences with design in light of the scholarship on design cognition and cognitive behaviors associated with ASC, suggests that the weak central coherence theory of autism provides a useful prediction of the cognitive processes necessary for *expertise* in design. While this implies that the jury is still out as to which cognitive processes are essential to design, it does contribute to our understanding of design cognition. As such our study confirms the value of including in design research the experiences of designers who are different from the 'typical' designer, as suggested by earlier studies (Heylighen and Nijs 2014).

Our study analyzed the experiences of two designers only. Further research should include the design experiences of more designers with ASC. Moreover, our study analyzed their experiences with design based on the designers' accounts only. Relying on self-reporting risks a diverted presentation of design processes (Lawson 1994), and provides little insight into how these processes are situated in and distributed across a socio-material environment. Further research should include a more diverse set of data collection methods that allow to include multiple sources of evidence, e.g., design documents used and produced by autistic designers, and accounts of colleague designers or other actors who work(ed) with them. Finally, not all cognitive processes associated with design have been researched by design journals. Some cognitive processes, such as decision-making, may have been researched as an activity instead. A broadening of the scope of the review can identify other candidate cognitive processes, but should maintain focus on those processes relevant to the conceptual part of design cognition (Dong et al. 2017).

Acknowledgements The authors would like to thank Marijke Kinnaer for her help in the literature review.

References

Ahrentzen S, Steele K (2009) Advancing full spectrum housing: designing for adults with autism spectrum disorders. Arizone State University, Tempe

Alexander C, Ishikawa S, Silverstein M (1977) A pattern language. Oxford University Press, New York

American Psychiatric Association (2013) Neurodevelopmental disorders, in diagnostic and statistical manual of mental disorders. American Psychiatric Association, Arlington

Asperger H (1944) Die "Autistischen Psychopathen" im Kindesalter. Arch Psychiatr Nervenkrankh 117(1):76–136

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

Baron-Cohen S (1990) Autism: A specific cognitive disorder of and ‘mind-blindness’. Int Rev Psychiatry 2(1):81–90
Baron-Cohen S (2001) Theory of mind and autism: a review. Int Rev Res Ment Retard 23:169–184
Baron-Cohen S, Leslie AM, Frith U (1985) Does the autistic child have a theory of mind. Cognition 21(1):37–46
Baumers S, Heylighen A (2015) Capturing experience: an autistic approach to designing space. Des J 18(3):327–343
Bilda Z, Gero JS (2007) The impact of working memory limitations on the design process during conceptualization. Des Stud 28(4):343–367
Chai K-H, Xiao X (2012) Understanding design research: a bibliometric analysis of design studies (1996–2010). Des Stud 33(1):24–43
Christensen BT, Schunn CD (2009) The role and impact of mental simulation in design. Appl Cogn Psychol 23(3):327–344
Christiaans H, Almendra RA (2010) Accessing decision-making in software design. Des Stud 31 (6):641–662
Cross N (2003) The expertise of exceptional designers. In: Cross N, Edmonds E (eds) Expertise in design. Creativity and Cognition Press, Sydney
Cross N (2004) Expertise in design: an overview. Des Stud 25(5):427–441
Cross N (2006) Designerly ways of knowing. Springer, London
Cross N, Cross AC (1998) Expertise in engineering design. Res Eng Des 10(3):141–149
Davis NO, Carter AS (2014) Social Development in Autism. In: Volkmar FR et al (eds) Handbook of autism and pervasive developmental disorders. Wiley, Hoboken, pp 212–229
Delacato CH (1974) The ultimate stranger: the autistic child. Doubleday, Garden City
Delfos MF (2005) A strange world: about autism, Asperger’s syndrome and PDD–NOS. Jessica Kingsley, London
Dong A (2005) The latent semantic approach to studying design team communication. Des Stud 26(5):445–461
Dong A, Collier-Baker E, Suddendorf T (2017) Building blocks of human design thinking in animals. Int J Des Creat Innov 5(1)
Eysenck MW (1993) Principles of cognitive psychology. Lawrence Erlbaum, Hove
Frith U (2003) Autism: explaining the enigma. Blackwell, Oxford
Frith U (2008) Autism: a very short introduction. Oxford University Press, Oxford
Frith U, Happé F (1994) Autism: beyond “theory of mind”. Cognition 50(1):115–132
Gemser G, de Bont C, Hekkert P, Friedman K (2012) Quality perceptions of design journals: the design scholars’ perspective. Des Stud 33(1):4–23
Goel V, Pirolli P (1992) The structure of design problem spaces. Cogn Sci 16(3):395–429
Grandin T (1995a) How people with autism think. In: Schopler E, Mesibov GB (eds) Learning and cognition in autism. Plenum Press, New York, pp 137–156
Grandin T (1995b) Thinking in pictures and other reports from my life with autism. Doubleday, New York
Grandin T (2009) How does visual thinking work in the mind of a person with autism? a personal account. Philos Trans R Soc Lond B: Biol Sci 364(1522):1437–1442
Happé F (1999) Autism: cognitive deficit or cognitive style? Trends Cogn Sci 3(6):216–222
Happé F, Frith U (2006) The weak coherence account: detail-focused cognitive style in autism spectrum disorders. J Autism Dev Disord 36(1):5–25
Hernandez NV, Shah JJ, Smith SM (2010) Understanding design ideation mechanisms through multilevel aligned empirical studies. Des Stud 31(4):382–410
Heylighen A, Nijs G (2014) Designing in the absence of sight: design cognition re-articulated. Des Stud 35(2):113–132
Iarocci G, McDonald J (2006) Sensory integration and the perceptual experience of persons with autism. J Autism Dev Disord 36(1):77–90
Kanner L (1943) Autistic disturbances of affective contact. Nerv Child 2:217–250

Khaidzir K, Lawson B (2013) The cognitive construct of design conversation. Res Eng Des 24 (4):331–347
Kim MH, Kim YS, Lee HS, Park JA (2007) An underlying cognitive aspect of design creativity: limited commitment mode control strategy. Des Stud 28(6):585–604
Kim SH, Paul R, Tager-Flusberg H, Lord C (2014) Language and communication in autism. In: Volkmar FR et al (eds) Handbook of autism and pervasive developmental disorders. Wiley, Hoboken, pp 230–262
Kleinsmann M, Deken F, Dong A, Lauche K (2012) Development of design collaboration skills. J Eng Des 23(7):485–506
Lawson B (1994) Design in mind. Reed Educational and Profesional, Oxford University Press
Lawson B, Dorst K (2009) Design expertise. Architectural Press, Oxford University Press
Lombardo MV, Baron-Cohen S (2011) The role of the self in mindblindness in autism. Conscious Cogn 20(1):130–140
Mostafa M (2007) An architecture for autism: concepts of design intervention for the autistic user. Archnet-IJAR, Int J Archit Res 2(1):189–211
Nagai Y, Taura T, Mukai F (2009) Concept blending and dissimilarity: factors for creative concept generation process. Des Stud 30(6):648–675
Noens I, van Beckelaer-Onnes I (2004) Making sense in a fragmentary world: communication in people with autism and learning disability. Autism 8(2):197–218
Ozonoff S, Pennington BF, Rogers SJ (1991) Executive function deficits in high-functioning autistic individuals: relationship to theory of mind. J Child Psychol Psychiatry 32(7):1081–1105
Park C (1992) Autism into art: a handicap transfigured. In: Schopler E, Mesibov GB (eds) High-functioning individuals with autism. Plenum Press, New York, pp 250–259
Paton B, Dorst K (2011) Briefing and reframing: a situated practice. Des Stud 32(6):573–587
Petre M (2004) How expert engineering teams use disciplines of innovation. Des Stud 25(5):477–493
Rajendran G, Mitchell P (2007) Cognitive theories of autism. Dev Rev 27(2):224–260
Sánchez PA, Vázquez FS, Serrano LA (2011) Autism and the built environment. In: Williams T (ed) Autism spectrum disorders: from genes to environment. InTech, Rijeka
Schön DA (1988) Designing: rules, types and words. Des Stud 9(3):181–190
Schön DA, Wiggins G (1992) Kinds of seeing and their functions in designing. Des Stud 13 (2):135–156
Simon HA (1969) The sciences of the artificial. MIT Press, Cambridge
Simon HA (1995) Problem forming, problem finding, and problem solving in design. In: Collen A, Gasparski WW (eds) Design and systems: general applications of methodology. Transaction Publishers, New Brunswick, pp 245–257
Suddendorf T, Dong A (2013) On the evolution of imagination and design. In: Taylor M (ed) The Oxford handbook of the development of imagination. Oxford University Press, Oxford, pp 453–467
van der Lugt R (2005) How sketching can affect the idea generation process in design group meetings. Des Stud 26(2):101–122
Vermeulen P (2010) Autisme als contextblindheid. Uitgeverij Acco, Berchem
Welsh MC, Pennington BF (1988) Assessing frontal lobe functioning in children: views from developmental psychology. Dev Neuropsychol 4(3):199–230
Wiltschnig S, Christensen BT, Ball LJ (2013) Collaborative problem–solution co-evolution in creative design. Des Stud 34(5):515–542
Wing L (1997) The history of ideas on autism: legends. Myths Real Autism 1(1):13–23
Youmans RJ (2011) The effects of physical prototyping and group work on the reduction of design fixation. Des Stud 32(2):115–138

Solution-Oriented Versus Novelty-Oriented Leadership Instructions: Cognitive Effect on Creative Ideation

Hicham Ezzat, Marine Agogué, Pascal Le Masson and Benoit Weil

Abstract The generation of novel ideas is critical to any innovative endeavor. However, one of the key obstacles to creativity is known as the fixation effect, which is the cognitive effect that constrains the generation of novel ideas due to the spontaneous activation of existing knowledge and solutions in individuals' mind. Expert leaders have been considered to play an important role in overcoming these biases using diverse tools. One of these principal instruments is task instruction. Our hypothesis is that leaders' instructions can have significant effects on followers' ideation capacity. We investigated the effect of an instruction given by a leader to his team to generate as many original ideas to a particular creative task, either using solution or novelty-oriented approaches. Results confirmed that solution-oriented instructions activated knowledge bases in fixation, while solution-oriented instructions inhibited these knowledge bases. These results give us new sights into novel models of "less-expert" creative leadership.

Introduction

Becoming faster, smarter, and increasingly complex, today's world emphasizes the need for creativity and innovation. Recently, survival of organizations became principally linked to the creative generation capacity of their employees. In the past decade, numerous businesses have failed to maintain their position in the innovation flow, and have disappeared from the business scene. These phenomena could be an indication of a lack of creativity in these firms, or perhaps a sign that leaders are unable to benefit from employees' creativeness resources in these organizations.

One possible origin of these blocs to creativity lies principally in a pure cognitive context. Numerous studies in cognitive sciences have highlighted the

H. Ezzat (✉) · P. Le Masson · B. Weil
Mines ParisTech, Paris, France
e-mail: hicham.ezzat@mines-paristech.fr

M. Agogué
HEC Montréal, Montréal, Canada

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_6

obstructive role of cognitive biases to creativity occurring during ideation processes. One of the most common cognitive bias to creativity is the functional fixedness, also called fixation effect (Jansson and Smith 1991), which is the difficulty that individuals are facing during creative contexts, in which they have to solve problems unconventionally.

Many methods like brainstorming, brain writing, and mind mapping (among others) have helped to overcome these cognitive biases to creativity, and increased the creative generation capacity of individuals. Today, these creativity generation methods are endless. Some researchers have even presented a long list of these creativity techniques (Kowaltowski et al. 2010).

Our work highlights the key role that could be played by leaders to stimulate followers' creativity. We examined the cognitive effect of leadership instructions on the creative generation capacity of subordinates during a particular creative task. The instruction given by leaders was to generate the maximum number of original solutions to a certain problem, either using a novelty-oriented search approach or using a solution-oriented search approach. We performed this experiment via a typical creative task where the aim was to propose the maximum number of original solutions to ensure that a hen's egg dropped from a distance of ten meters does not break.

Leadership Definitions

Leadership has been recognized as one of the most observed and least understood phenomena on earth (Burns 1978). It is today extremely difficult to accurately define this term. Indeed, leadership does not have a one-size-fits-all definition. Different explanations describing what leadership is could simply vary from one sector to another (Military, Politics, Education, Sports, etc.…). Stodgill concluded that leadership has as many definitions as those who have attempted to define it (Stogdill 1974). Over the past 50 years, scholars and theorists have commonly delineated it into several agreed-upon definition fragments, and have decided to define it widely as a "process of social influence in which a person (the leader) is able to enlist the support of others (his/her subordinates or followers) in the accomplishment of a common task" (Chemers 2014).

Studies on leadership have produced numerous theories involving personality and traits, power and influence, behavior, situations, transactional, transformational, and integrative (among the most cited) (Van Seters and Field 1990). Despite the complexity of precisely delineating leadership study eras throughout the history, several scholars have made the effort to classify the major research eras according to the major theories that were developed in each of these periods (Van Seters and Field 1990; Yukl 1994; Brungardt 1997; Bolden et al. 2003; Daft 2014).

Notwithstanding the huge historical contributions, progress and development made to the literature of leadership, little focus has been placed around creativity; not because creativity was not a priority, but perhaps because it was directly incorporated in the general notion of efficiency. Until very recently, creativity began to be a subject of serious studies and findings among leadership theorists.

Creativity Versus Leadership

A recent survey made by IBM to more than 1500 large companies' CEOs revealed that the most important leadership quality is creativity (Carr and Tomasco 2010). In almost every job, or occupation, there is a place for a certain level of creativity.

Creativity has repetitively been defined as the ability to generate ideas that are both novel and useful, while innovation enlarges the definition of creativity, and involve taking those creative ideas and carrying them through to implementation (Amabile 1983). Cognitive scientists described it more deeply as a product of many types of intellectual processes that helps setting the stage for creative insights and discoveries (Mueller et al. 2011).

Creative people usually demonstrate a high ideational fluency (which is the aptitude to come up with many new ideas), high degree of novelty, and as well high level of flexibility (known as the ability to stimulate variety among new ideas) (Guilford 1959).

One of today's leadership key roles is to stimulate creativity among subordinates, not only at the individual level, but also at the group and organizational level, by directly or indirectly adapting and manipulating appropriate variables like group climate, group composition, resources, and knowledge management (Hemlin et al. 2008).

However, leadership and creativity could be perceived as antagonist concepts, compromised between control and freedom, where too much leadership control kills creativity, and too much freedom leads to chaos. Studies found that a creative leader could be perceived as deviating from the status quo, neglecting interpersonal activities, and not promoting order (Mueller et al. 2011). Although it can be argued that creativity is unpredictable and cannot be managed in a strict sense, researches proved that creative leaders could control the necessary variables for creativity (Amabile and Gryskiewicz 1987).

State of the Art on Creative Leadership

Literature review of creative leadership has underlined the role played by leaders for creativity (Mumford et al. 2002). Very early studies on creative leadership underlined the importance of the creativity of leaders themselves, assuming that by being creative, leaders will be having the necessary vision for followers creativity (Mumford et al. 2003).

Prior works have majorly reduced and concentrated leaders' roles as facilitators, mentors, or mediators to organizational creativity (Woodman et al. 1993). These studies have examined the varied factors that can either foster or hinder employees' creativity at individual, group, and organizational levels, and have subsequently introduced the role of creative leaders.

Other studies on creative leadership were centered especially on the role of leaders' behavior to enhance employees' creativity, i.e. by exploring the direct link between leaders' behavior and creativity. Moreover, studies have also highlighted the role that can be played by leaders to boost employees' creativity by managing external factors like work climate or human resources issues (Shalley and Gilson 2004). Deeper studies have analyzed leadership from a creative problem-solving perspective emphasizing the role of leaders to facilitate cognitive processes for more creativity (Reiter-Palmon and Illies 2004), while others even proposed some models of creative leadership (Sternberg et al. 2003; Rickards and Moger 2000).

Transformational and transactional leadership are two well-studied styles in creative leadership (Jung 2001). Studies made at the transactional era found positive correlations between extrinsic task-motivation (like rewards for example) and employees' creativity. Positive correlations with creativity were as well found during the transformational era, with the introduction among others of the role of emotional intelligence (Zhou and George 2003), and intrinsic task motivation for creativity (Amabile 1997).

However, these two styles are not mutually exclusive, and many leadership scholars modeled creative leadership as a combination of both transformational and transactional styles to enhance creativity, depending on leaders' cognitive goals structure (Wofford and Goodwin 1994), and their ability to manage their team to reach the goal.

Managing Goals: The Role of Leadership Instructions

One of the main functions of a creative leader is to stimulate creativity among subordinates, and guide them by appropriately specifying the target goal. Considered a key element in almost all leadership theories, the element "goal" appears very clearly as a basic component of the path-goal theory (House 1971), in which leaders inspire followers to achieve a goal, by guiding them throughout the process to undertake the appropriate paths leading to the goal.

Traditional leadership styles were pinpointed on the goal component. For instance, transactional leadership is principally based on completing clear and specific goals. When the responsibilities or requirements are successfully completed, transactional leaders give their followers reward in return, yet punish when the followers deviate from the standard.

Moreover, conventional leadership competencies, like planning, organizing, analyzing, goal setting are as well focused around the fundamental role of leadership, which is reaching a specific desired target that a leader envisions and plans to achieve. In this regard, goal could be considered as an organizational desired end-point in some kind of expected organization development.

Among the long list of leadership tools to appropriately set and manipulate goals, there is no doubt that instructions maintain a central and key position. In different studies, studying and experimenting the effects of particular goals have been implicitly modeled, represented and tested through instructions. Even though most studies in creativity consider leaderless groups, there is always an instructor (or experimenter) that clarifies task objectives (and sometimes expectations), by instructing participants what to do at the beginning of the task, or even within the task.

Problematic: A Highly Expert and Knowledgeable Leadership

Setting and managing task goals is one of the critical roles of leaders. Not only leaders should be competent enough to support their followers in achieving the goal, but also capable of clarifying the paths towards reaching it. Moreover, according to path-goal theory of leadership, an efficient leader is even able to ensure that his followers are attaining the goal, by removing all obstacles and difficulties that are preventing them from reaching it (House 1971).

There is no doubt that all these facts assume a standard of leadership with high levels of task-domain knowledge and expertise, which is not the case in most circumstances. In certain cases, if leaders are not enough expert, they could mislead their team in wrong directions.

In this research, we were interested to study a case where leaders are less expert and less knowledgeable, but even though capable of giving their followers appropriate task instructions, in order to reach creative solutions to problems.

Our research problem includes the following questions: could leaders increase subordinates' ideation capacity without necessary having strong task-domain expertise and knowledge? In other words, what type of instructions could leaders give to their team to guide their ideation processes towards creativity, with the minimum level of task-domain expertise and knowledge?

To answer these questions, (1) we analyzed the literature review of the impact of instructions on creative ideation, and derived the associated implicit model of leadership; (2) we proposed a model of leadership instructions, and its hypotheses; (3) we presented the theory-driven experiment to test our theoretical predictions; (4) we analyzed the obtained results; (5) we ended with the conclusion and limitations of our study.

Literature Review: Impact of Instructions on Creative Ideation

Ideation has been considered a fundamental process for creative idea generation. Many scholars considered that facilitating the generation of a large number of ideas should lead to creativity. For these reasons, much attention has been given to the different factors contributing to facilitate it.

Many studies in cognitive science have highlighted the close relationship existing between instructions and creative idea generation. Several researchers found that the highest creativity scores in a group occurred when individuals simply had a "creativity goal" instructed and worked alone under expected evaluation (Shalley 1995; Carson and Carson 1993). More studies have affirmed that when individuals are instructed that creativity is important as a goal, they are more likely to be creative. However, instructing that creativity is important as a goal is essential but not enough to overcome the cognitive biases occurring in creative contexts.

Studies found that instructional factors can highly influence the outcome of the creative problem-solving effort, in a way that is consistent and coherent with the instructions (Runco et al. 2005). In these studies, the type of instructions had a differential effect on the different evaluation norms (fluency, flexibility, originality) of creative solutions (Shalley 1995; Runco and Okuda 1991). When participants were instructed with flexibility-oriented instructions, they generated more different ideas than when instructed with originality-oriented instructions. However, originality rates decreased when given flexibility instructions.

More deeply, Runco et al. tested several types of instructions on creative ideation, and found that procedural instructions are better in terms of fluency, flexibility and originality than conceptual instructions (Runco et al. 2005). The level of explicitness of instructions provided for problem solvers influenced the creativity of solutions generated. The more ambiguous instructions were, the more novel and unique generated ideas and solutions were Ward et al. 2004.

Furthermore, latest studies conducted in interdisciplinary frameworks mixing cognitive sciences with management sciences have helped to clarify the nature of these cognitive biases to creativity, and have been able to identify some type of examples that could be instructed to individuals to overcome fixation effects. One of the principal result in this field is the positive effect of expansive examples on the creative ideation capacity of individuals, while the negative effect of restrictive examples on their creativity (Agogué et al. 2014).

Model of Leadership Derived from Literature Review

Despite the contributions made to the literature on the effects of instructions on creativity, and the indisputable importance of studying instructions' effects on ideation in terms of fluency, flexibility and originality; little is known about the profile of the instructor, his expertise and his knowledge.

The above-mentioned literature review describes an implicit model of instructors that are having high levels of task-domain knowledge and expertise. If we analyze it from a leadership perspective, in order to increase the creative generative capacity in ideation tasks, literature review assumes that the leader should have the following task-domain expertise and knowledge:

- Having high level of task-domain expertise to give procedural instructions to his team to complete tasks in a stepwise manner.
- Having high level of task-domain expertise to transform an explicit instruction into a more ambiguous and implicit one.
- Having high level of task-domain knowledge to stimulate his team with appropriate expansive examples, i.e. an idea that is in expansion (outside the fixation zone).

In consistence with the dominant design of creative leadership presented earlier, being able to give stimulating instructions for creativity generation requires leaders having high levels of expertise and knowledge about the task.

In this study, our motivation was to model instructions that do not necessary require very high levels of expertise and knowledge from the instructor. We assumed in this study, that in order to be able to guide their team in potential directions towards creativity, leaders should have at least the ability to identify the dominant design of the task, i.e. the principal categories of ideas and solutions supposed inside the fixation effect zone.

Based on C-K theory, we were interested to model leadership instructions from a cognitive perspective within an ideation context, in which initial instructions could guide ideation paths to certain types of ideas and solutions, whether they are restrictive, i.e. do not change object's definition or attributes, or whether they are expansive, i.e. transform object's definition and identity (Agogué et al. 2014). This cognitive perspective could be more beneficial, as it would provide us with more details on cognitive biases such as fixation effects.

Modeling Leadership Instructions Using Design Theories

In the field of design science, several theories like TRIZ (Al′tšuller 1999), ASIT, SCAMPER, have helped to stimulate creativity in industrial contexts. More recently, C-K theory (Hatchuel and Weil 2003), and KCP method (Hatchuel et al. 2009)

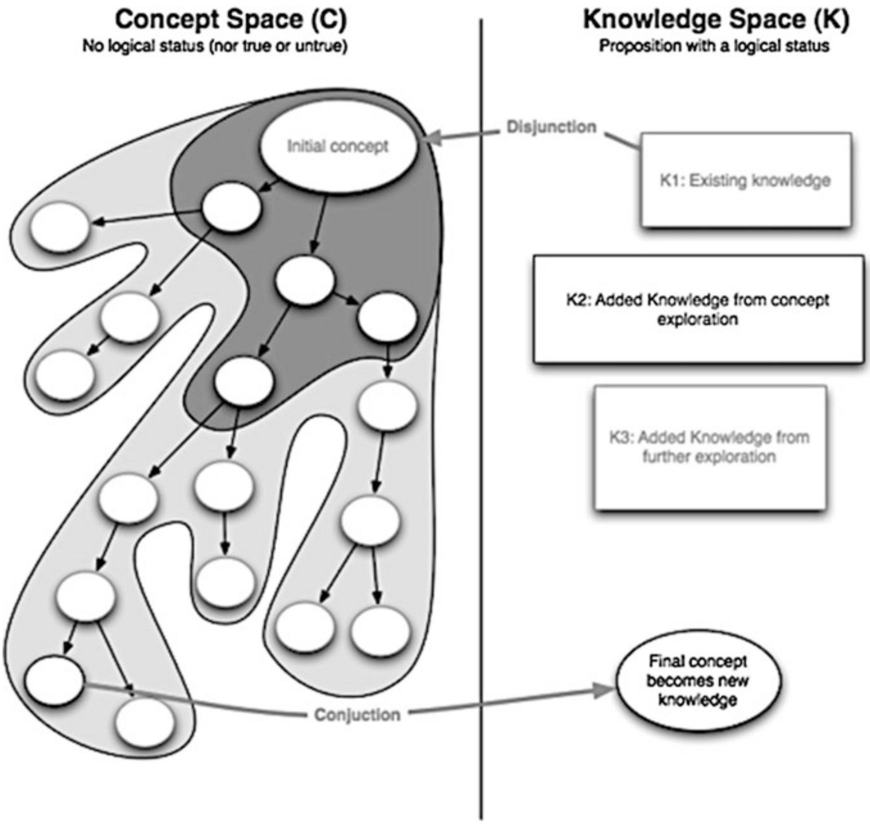


Fig. 1 C-K diagram

emerged not only as a design theory, but moreover as a theory of cognitive reasoning to bypass cognitive biases' effects occurring in design contexts.

Concept-Knowledge theory's helps overcome cognitive biases' effects occurring in creative design contexts. C-K theory defines two spaces: a space of concepts (labeled "C") and a space of knowledge (labeled "K"). The process of design is defined as a double expansion of both C and K spaces, via four operators (Hatchuel and Weil 2002) (as shown in Fig. 1):

- C → K: this operator is called "conjunction". It seeks for added (or subtracted) properties in K space to reach propositions having a logical status (true or false).
- K → C: this operator is called "disjunction". It adds (or subtracts) some properties coming from K space to form new concepts having no logical status.
- C → C: this operator expands the C space by adding a new partition to it. This new partition can be "restrictive" if it does not change object's definition or attributes, or "expansive" if it transforms object's definition and identity by adding (or removing) unexpected attributes.

- K → K: this operator expands the K space by adding new knowledge bases to it, and indicates the knowledge structure created within the design concept.

Using this theory, we could model two types of instructions, based on their cognitive effect on the knowledge space:

- Instructions (type A) that would force individuals' cognitive reasoning in K space to stimulate/activate the known and existing knowledge bases (related to fixation), and to inhibit/deactivate novel knowledge bases (unrelated to fixation).
- Instructions (type B) that would force individual's cognitive reasoning in K space to inhibit/deactivate known knowledge bases (related to fixation), and stimulate/activate novel knowledge bases (unrelated to fixation).

As illustrated in Fig. 2, these two types of instructions could have different impact on the generation of new concepts in C space, by disjunction of activated knowledge bases coming from K space.

On the one hand, we hypothesized that if leaders' instruction indicates that an expected evaluation of testability would be made, this should force subordinates to search for ideas that solve problem-solving tasks in obvious and conventional ways. In other words, knowing that generated ideas must successfully work, individuals will tend to activate already existing knowledge bases, and inhibit novel ones. We then called instructions type A: "solution-oriented".

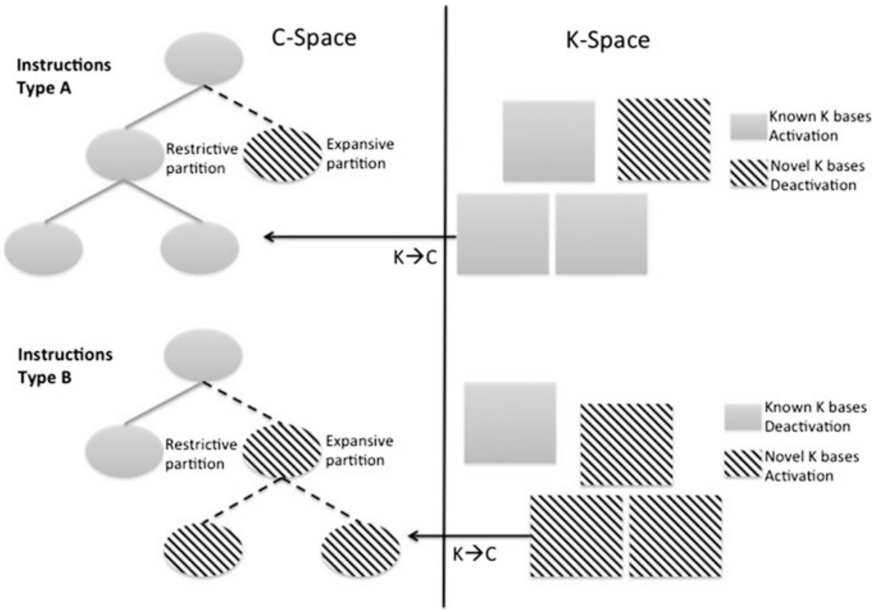


Fig. 2 Modeling instructions using C-K theory

Table 1 Solution versus novelty-oriented leadership instructions

Instruction type	Leadership expected evaluation	Subordinates expected search approach	Expected cognitive effect on knowledge space
Solution-oriented	Ideas will be tested experimentally	Obvious conventional	Stimulation of existing knowledge bases Inhibition of novel knowledge bases
Novelty-oriented	Ideas must be different from an existing set of categories of ideas	Non-obvious unconventional	Stimulation of novel knowledge bases inhibition of existing knowledge bases

On the other hand, we hypothesized that if leaders' instruction indicates that an expected evaluation of novelty would be made (by restricting the dominant design, i.e. categories of ideas related to fixation), this should force subordinates to generate ideas that respond to problem-solving tasks in non-obvious and unconventional ways. In this case, individuals will tend to inhibit already existing knowledge bases, and activate novel ones. We then called instructions type B: "novelty-oriented".

Table 1 illustrates these solution-oriented and novelty-oriented leadership instructions in terms of expected search approaches (how these instructions would approach the problem), and the expected cognitive effect on the knowledge space of C-K theory (what knowledge bases are expected to be activated/deactivated).

Methods

Participants

Participants (N = 54) of the course "Products Design and Innovation" have participated in this study. The experiment was made during the first day of the course. Participants were engineering students and professionals working in innovative fields. Subjects were between 20 and 43 years old, with a mean age of 24.4. Only two participants informed us having already done this creativity task previously in other creative design courses.

Procedure

We chose to perform our experiment on the classical hen's egg task where the aim was to propose the maximum number of original solutions to ensure that a hen's egg dropped from a distance of ten meters does not break. We selected this special creativity task among others since we have a vast existing database of ideas and solutions of more than thousands subjects having different profiles (students, engineers, designers, etc.…) that performed this task within the past years.

As presented in the C-Space tree in Fig. 3, our database indicates that more than 80% of previous subjects generated ideas around three main categories of "restrictive" solutions (which are damping the shock, slowing the fall, and protecting the egg). This enabled us to identify the dominant design of the task as being focused on the "fragility of the egg". However, less than 20% of subjects were able to generate "expansive" solutions (for instance: before and after the fall, with a living device, using the intrinsic properties of the environment, etc.).

Participants were randomly divided into three groups. Each group had to perform individually the task. Each participant was given a written initial instruction depending on the group he/she belonged to, as illustrated in Table 2.

Participants of the first group were given the control instruction with the classical creativity objective as usually instructed in all previous experiments (in our database), which expected them to generate as many original solutions as possible to the problem. We considered group 1 as a referential group for studying groups 2 and 3.

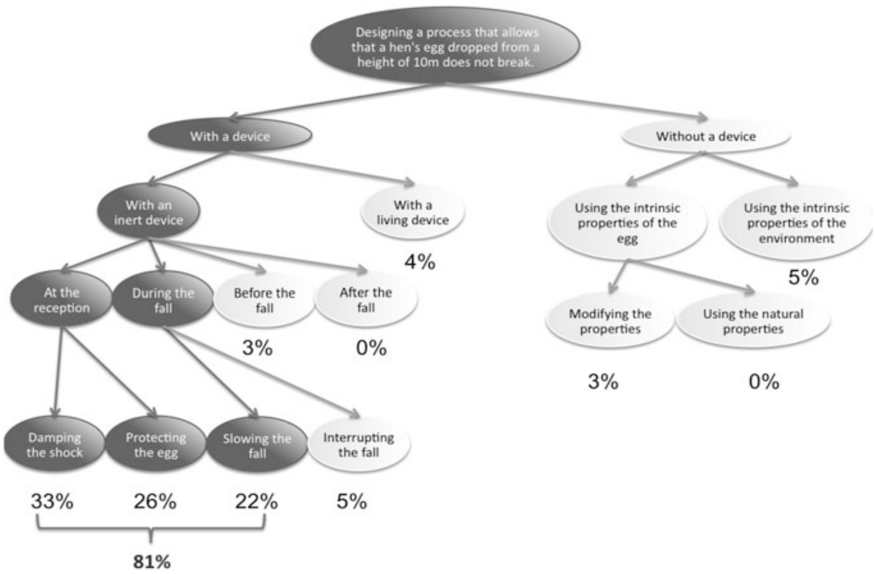


Fig. 3 Using C-Space tree to determine fixation in the egg's task (Agogué et al. 2014)

Table 2 Leadership instructions given to participants

Groups	Leadership instructions
Group 1: Control	You are a designer and your manager gives you the following problem: ensure that a hen's egg dropped from a height of 10 m does not break The evaluation of your manager will be based on the number of original ideas you will propose
Group 2: Solution-oriented	You are a designer and your manager gives you the following problem: ensure that a hen's egg dropped from a height of 10 m does not break The evaluation of your manager will be based on the number of original ideas you will propose, knowing that your solutions will be tested experimentally
Group 3: Novelty-oriented	You are a designer and your manager gives you the following problem: ensure that a hen's egg dropped from a height of 10 m does not break. The evaluation of your manager will be based on the number of original ideas you will propose, knowing that your solutions must not dampen the shock, or slow the fall, or protect the egg

Participants of the second group had to perform exactly the same task, but with a solution-oriented instruction. We expected that this instruction should force the participants to reflect on solutions focused and fixated on the principal element of the task (which is the "fragility of the egg"), knowing that their ideas will be tested and should successfully work.

The third group performed as well the same task, but was having a novelty-oriented instruction forcing them to search for ideas outside the three main categories in fixation zone (all focused on the principal element of the task). Manager expected group 3 to generate ideas and solutions, knowing that they must not belong to the already pre-existing categories of "restrictive" responses.

Results

Ideation is the process of coming up with alternative solutions to a particular problem. As we have indicated earlier, divergent thinking has been known to consider three main elements for evaluating a creative ideation process, which are the ideational fluency (which refers to the mean number of ideas generated by a population), ideational originality (which refers to the frequency of occurrence of the type of idea), and ideational flexibility (which is the number of different groups/categories of ideas generated by individuals).

In terms of fluency, we computed the mean number of ideas generated by participants of each group, as well as the SEM (Standard Error of the Mean). Results (in Fig. 4) showed that participants of the solution-oriented group (group 2) were able to generate more ideas than participants of the novelty-oriented group (group 3), as well as participants of the control group (group 1). Interestingly,

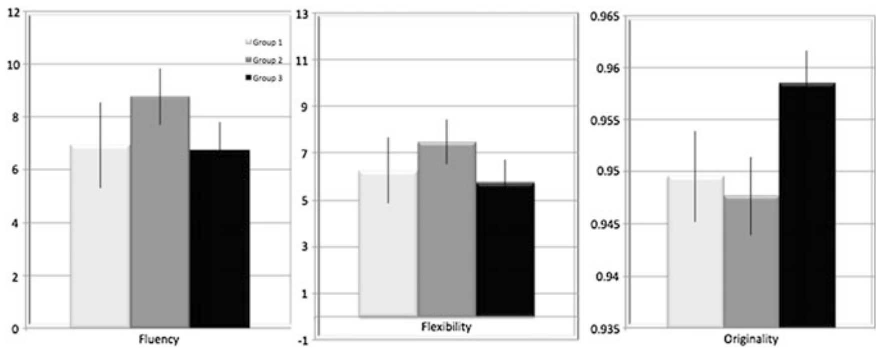


Fig. 4 Statistical analysis of groups' results

participants of group 3 were able to generate a mean number of ideas that is quite similar than group 1.

In terms of flexibility of solutions, we calculated the mean number of different categories of ideas generated by participants in each group. In this regard, group 2 had the highest score of flexibility, while group 3 had the lowest one. In this regard, SEM values shows that flexibility results are not very significant.

Finally, we analyzed originality of ideas of each group by computing the frequency of solutions given across all the subjects. In this regard, we found that participants of group 3 we able to generate more original and unique solutions than participants of group 2 and group 1.

Nevertheless, analyzing mean number of solutions of each group in terms of fluency/flexibility/originality is not enough in this study. For this reason, and in order to have a better view of the effect of instructions on creative ideation, we analyzed the mean number of solutions that groups were able to generate in the fixation zone (restrictive solutions), or in the expansion zone (expansive solutions), as illustrated in Fig. 5.

Results showed that group 2 tended to generate more restrictive ideas and solutions than group 3 and group 1. Contrary to group 2, group 3 tended to generate more expansive ideas and solutions than group 2 and group 1. As shown in Fig. 5, the mean number of expansive ideas generated by group 3 is approximately equivalent to the mean number of restrictive ideas generated by the referential group. These results show that novelty-oriented instruction forced the activation of knowledge bases outside fixation, which augmented the number of expansive ideas generated by participants. On the other hand, solution-oriented instruction forced the activation of knowledge bases inside fixation, which increased the number of restrictive ideas generated by participants.

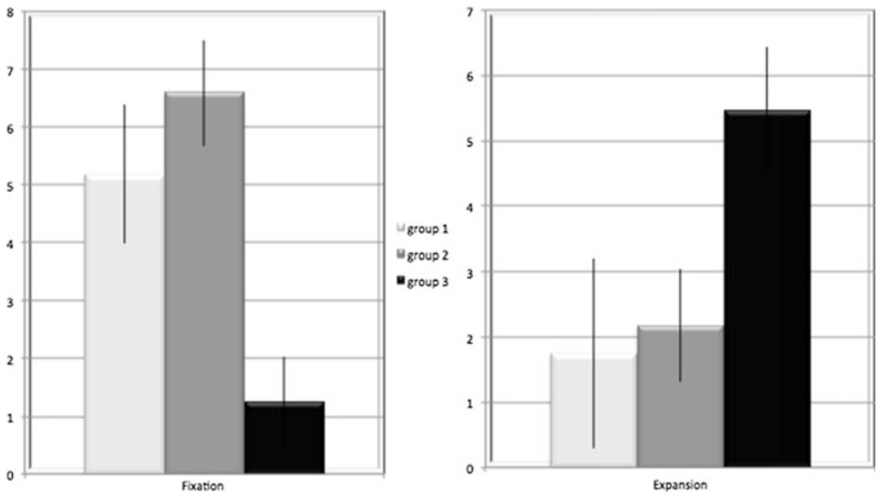


Fig. 5 Mean number of solutions inside and outside fixation in each group

Conclusions and Limitations

In this study, we explored the effect of two types of leadership instructions on subordinates' creative idea generation capacity during a creative task. The aim of the task was to propose the maximum number of original solutions to ensure that a hen's egg dropped from a distance of ten meters does not break. We compared solution-oriented leadership instructions to novelty-oriented leadership instructions.

From a cognitive sciences' perspective, two major findings emerged from this investigation: (1) the group exposed to novelty-oriented instructions was able to generate more original solutions than the group exposed to solution-oriented instructions; (2) the group exposed to solution-oriented instructions tended to generate more restrictive ideas, while the group exposed to novelty-oriented instructions tended to generate more expansive ideas.

From a managerial perspective, our results show that less expert and knowledgeable leaders (having at least the ability to recognize the task dominant design, i.e. principal categories of restrictive ideas) could play an important role in stimulating followers' creativity.

Future works will consist in testing this novelty-search approach within a more dynamic leader-member interaction, by investigating leadership instructional processes in real-time using feedbacks. This future study could enable exploring new sights in creative leadership behaviors, as well as leaders' capacity to drive the idea generation paths towards fixation or expansion in real-time.

A limitation of our study is that it is far from a real setting involving a leader-member exchange process. Our study could be considered as a building block for further more realistic interdisciplinary leadership studies mixing cognitive sciences with management. These future studies could take into account the complexity of the numerous contextual variables of leadership equation (real contexts with incentives, hierarchy issues for subordinates, specific contextual factors, etc.)

Acknowledgements This research was financed by a grant from the French National Research Agency (project ANR-13-SOIN-0004-04 IDéfixE). The authors would like to thank Sophie Hooge for giving us the opportunity to run these experiments during her course "Products Design and Innovation".

References

Agogué M et al (2014) The impact of type of examples on originality: explaining fixation and stimulation effects. J Creative Behav 48(1):1–12

Al′tšuller GS (1999) The innovation algorithm: TRIZ, systematic innovation and technical creativity. Technical Innovation Center Inc., Worcester

Amabile TM (1983) The social psychology of creativity: acomponential conceptualization. J Pers Soc Psychol 45(2):357

Amabile TM (1997) Motivating creativity in organizations: on doing what you love and loving what you do. Calif Manag Rev 40(1):39–58

Amabile T, Gryskiewicz SS (1987) Creativity in the R&D laboratory. Center for Creative Leadership

Bolden R et al (2003) A review of leadership theory and competency frameworks. University of Exeter, Centre for Leadership Studies, Exeter

Brungardt C (1997) The making of leaders: a review of the research in leadership development and education. J Leadersh Organ Stud 3(3):81–95

Burns JM (1978) Leadership. Harper & Row, New York

Carr A, Tomasco S (2010) The most important leadership quality for CEOs? Creativity. Fast Company, New York

Carson PP, Carson KD (1993) Managing creativity enhancement through goal-setting and feedback*. J Creative Behav 27(1):36–45

Chemers M (2014) An integrative theory of leadership. Psychology Press, New York

Daft R (2014) The leadership experience. Cengage Learning, Boston

Guilford JP (1959) Traits of creativity. Creativity Cultiv 10:142–161

Hatchuel A, Weil B (2002) CK theory. In: Proceedings of the Herbert Simon international conference on design sciences. Citeseer

Hatchuel A, Weil B (2003) A new approach of innovative design: an introduction to CK theory. In: DS 31 Proceedings of ICED 03, the 14th international conference on engineering design, Stockholm

Hatchuel A, Le Masson P, Weil B (2009) Design theory and collective creativity: a theoretical framework to evaluate KCP process. In: International conference on engineering design, ICED

Hemlin S, Allwood CM, Martin BR (2008) Creative knowledge environments. Creat Res J 20 (2):196–210

House R.J (1971) A path goal theory of leader effectiveness. Adm Sci Q 16(3):321–339

Jansson DG, Smith SM (1991) Design fixation. Des Stud 12(1):3–11

Jung DI (2001) Transformational and transactional leadership and their effects on creativity in groups. Creativity Res J 13(2):185–195

Kowaltowski DC, Bianchi G, De Paiva VT (2010) Methods that may stimulate creativity and their use in architectural design education. Int J Technol Des Educ 20(4):453–476

Mueller JS, Goncalo JA, Kamdar D (2011) Recognizing creative leadership: can creative idea expression negatively relate to perceptions of leadership potential? J Exp Soc Psychol 47 (2):494–498

Mumford MD et al (2002) Leading creative people: orchestrating expertise and relationships. Leadersh Q 13(6):705–750

Mumford MD, Connelly S, Gaddis B (2003) How creative leaders think: experimental findings and cases. Leadersh Q 14(4):411–432

Reiter-Palmon R, Illies JJ (2004) Leadership and creativity: understanding leadership from a creative problem-solving perspective. Leadersh Q 15(1):55–77

Rickards T, Moger S (2000) Creative leadership processes in project team development: an alternative to Tuckman's stage model. Br J Manag 11(4):273–283

Runco MA, Okuda S (1991) The instructional enhancement of the flexibility and originality scores of divergent thinking tests. Appl Cogn Psychol 5(5):435–441

Runco MA, Illies JJ, Eisenman R (2005a) Creativity, originality, and appropriateness: what do explicit instructions tell us about their relationships? J Creative Behav 39(2):137–148

Runco MA, Illies JJ, Reiter-Ralmon R (2005b) Explicit instructions to be creative and original. Int J Creativity Probl Solving 15(1):5–15

Shalley CE (1995) Effects of coaction, expected evaluation, and goal setting on creativity and productivity. Acad Manag J 38(2):483–503

Shalley CE, Gilson LL (2004) What leaders need to know: a review of social and contextual factors that can foster or hinder creativity. Leadersh Q 15(1):33–53

Sternberg RJ, Kaufman JC, Pretz JE (2003) A propulsion model of creative leadership. Leadersh Q 14(4):455–473

Stogdill RM (1974) Handbook of leadership: a survey of the literature. Free Press, New York

Van Seters DA, Field RH (1990) The evolution of leadership theory. J Organ Change Manag 3 (3):29–45

Ward TB, Patterson MJ, Sifonis CM (2004) The role of specificity and abstraction in creative idea generation. Creativity Res J 16(1):1–9

Wofford J, Goodwin VL (1994) A cognitive interpretation of transactional and transformational leadership theories. Leadersh Q 5(2):161–186

Woodman RW, Sawyer JE, Griffin RW (1993) Toward a theory of organizational creativity. Acad Manag Rev 18(2):293–321

Yukl GA (1994) Leadership in organizations. Pearson Education India, New Delhi

Zhou J, George JM (2003) Awakening employee creativity: the role of leader emotional intelligence. Leadersh Q 14(4):545–568

A Protocol Study of Cognitive Chunking in Free-Hand Sketching During Design Ideation by Novice Designers

Omar M. Galil, Kirill Martusevich and Chiradeep Sen

Abstract This paper reports a human-subject protocol study aimed to study (1) cognitive chunking during free-hand sketching of design ideas in engineering and (2) correlation between chunks and the functions of the design perceived by the designer. Voluntary participants are presented with a previously unfamiliar design problem and asked to (1) identify the intended functionality of the solution, (2) draw concept sketches of the solution structure, and (3) label the sketches. The data is captured by a smart pen: a computerized pen that records pen strokes and surrounding audio in sync. Time study of these pen strokes and naming of components reveal clear evidence of temporal clustering of pen strokes during the first two tasks, indicating that the physical structure of the design solution is perceived by the designer in small chunks, rather than in continuous streams. Further data suggest that the chunks formed during ideation are based on functional needs perceived by the designer.

Introduction: Why Study Chunking in Engineering Sketching?

Engineering design is an information-transforming process. It starts with a problem or a need (information) and ends with the description of a solution (information). In the middle, information is transformed from one state to another, in small steps connected in a complex network. Of these steps, free-hand sketching is among the most common means used by engineers to express and communicate ideas, esp. during ideation. Sketching during ideation is significant because it is a real-time translation of information from the designer's cognitive domain to external media, e.g., paper. Thus, systematic study of sketching could provide insight into the cognitive and computational aspects of engineering creativity.

O.M. Galil · K. Martusevich · C. Sen (✉)
Florida Institute of Technology, Melbourne, FL, USA
e-mail: csen@fit.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_7

Studying the information process of free-hand sketching, esp. chunking, could also serve as a basis for quantifying information content of sketches. Quantification of information in design models is an active research area. Previous research includes computing information content of function models based on static vocabularies (Sen et al. 2010) and graph topology (Sen et al. 2010). Information of plain English texts has been computed based on parts of speech tags and modeling the words as entities, relations, attributes, and values (Sen et al. 2014). However, free-hand sketches do not use a finite set of symbols as CAD sketches do, nor do they contain parts of speech. Ultimately, there is no clear answer to the question: "What is an entity or a relation in a free-hand sketch, and how to define unit information in a sketch"? The answer could lie in using the cognitive chunks as units of information in sketches. In psychology, chunking is the process of combining multiple smaller data units into larger but fewer groups, or chunks (Miller 1956), in order to help with various cognitive tasks. The experiment presented here studies chunking during free-hand sketching.

Review of Related Work

The process and tendency of combining multiple smaller units of information into larger, but fewer and meaningful, groups such as combining letters into words (Miller 1956; Gobet 2001) is called Chunking in psychology research. The concept of chunks was proposed by Miller (1956) when short-term memory was modeled as an information channel and empirical data from various tasks involving various senses could be summarized to say that this channel has a capacity of 7 ± 2 chunks. This generalization implies that the number of chunks that humans can process at once in short term memory varies within this range irrespective of the task. The reported tasks included judging and/or recalling the pitch and loudness of an audible sound, the saltiness of aqueous solutions, the position of a pointer on a linear scale, and the position of a dot on a square. In the subsequent decades, the capacity of human memory in terms of chunk count and its dependence on the task and training of the subject has been debated (Ericsson and Kintsch 1995), but the existence of the chunking phenomenon as such has been further validated. These studies include recalling technical drawings (Ullman et al. 1990), circuit diagrams (Egan and Schwartz 1979), chess positions (Gobet 2001), and in various tasks of human learning (Gobet 2001). For example, the number of chess pieces recalled correctly by experts improves if the positions are taken from a tournament game rather than being randomly positioned (Gobet and Simon 1996).

In engineering, the study of chunking is limited to recall tasks, where the objective is to assess the ability to retain and recall presented information after a pause. For example, in one experiment, novice and expert designers were shown a mechanical assembly drawing for a short period of time (Ullman et al. 1990; Waldron and Waldron 1988). The drawing was then removed and designers asked to reproduce it. The protocol analysis indicated that the novices recalled only like segments, while experts recalled bigger chunks that were functional components of the assembly. However, the chunking in ideation has not been formally tested. It should be noted that ideation is a totally different creative cognitive process than recalling, since there is no initial reference data that is presented and recalled. Instead, solution ideas for completely novel problems are generated as graphical representations (sketches) by the designer's cognitive processes from problem statements that are presented using a different representation, such as texts. While chunking in ideation is anticipated to exist, empirical evidence to that end is not available yet. The experiment presented in this paper addresses that gap.

Protocol analysis is a method of studying human activities objectively by reducing them into symbolic codes and analyzing the codes to observe patterns of interest (Gero and McNeill 1998). Originally popularized in cognitive science (Ericsson and Simon 1993) and behavioral science (Austin and Delaney 1998), this method has been used to study engineering design, esp. the use of function modeling (Sen et al. 2012), analogy (Arlitt et al. 2013), design education (Cardella et al. 2006), and ideation (Ullman et al. 1988). Detail reviews of research in design creativity and protocol studies can be found at: (Smith et al. 2005; Dinar et al. 2015; Cross 2001). In the remainder of this paper, a human-subject study and its protocol analysis to reveal the chunking in ideation is described.

Design of the Experiment

Mode of Sketching Studied

Sketching is a real-time translation of information from the designer's cognitive domain to external media such as paper. However, it is used in design to accomplish different goals, each of which could have a different cognitive process (Ullman et al. 1990). For example, the cognitive processes involved in sketching the isometric view of an object from given orthogonal projections could potentially be different from the steps involved in doing the exact opposite, or in sketching an object's shape by looking at it physically. Thus it is important to identify the sketching type studied here and to recognize that the conclusions drawn from this study may not necessarily apply to the other sketching modes. In this study, the sketches were produced when designers generated solution concepts to an unfamiliar mechanical design problem presented in plain text. This mode of sketching is referred to as the "ideation" mode in this text, as opposed to copying mode, for example.

Participants

The voluntary participants in this study are six undergraduate and three graduate students in Mechanical Engineering at an accredited program in the United States. Of the available undergrad volunteers, only three juniors and three seniors within the grade point average (GPA) range of 2.8–3.2 are selected to limit the variation in academic profiles and because this range represents a large subset of the student population. The three seniors are enrolled in the capstone design course, while the juniors are not. The grad students are research assistants in the design research group, pursuing doctoral degrees, but none have taken a design methodology course where function modeling is taught. The total participant pool comprises of seven males and two females.

In addition, two additional volunteers are used during the pilot phase to develop and refine the study and the analysis protocol. Both of them are grad students of the design research group and they are not repeated in the actual pool from which observations and conclusions are drawn.

Setup and Pre-experiment Procedure

During each run of the study, the participant is seated at a desk located in an academic office familiar to her,[1] during the middle hours of a typical work day (10:00 a.m.–2:00 p.m.). Before the study tasks, a brief slideshow with embedded voice is played to explain the participant's rights, the voluntary nature of her participation, and the approval of the study by the Institutional Review Board (IRB). Next, the participant signs a form to indicate her consent to this process, and fills out a questionnaire about her academic information and degree of familiarity with various mechanical devices. The latter of these two is used to ensure uniformity of the participants in terms of background, education, experience, and skills. The participant is then given a brief tutorial on the usage of a Livescribe™ smart pen[2]—a device that she would need to use to complete the tasks of the experiment. Once the participant's questions are clarified, she is presented with the problem scenario of designing a novel mechanical device, as explained next.

[1]'She' and 'her' are used throughout this text to refer to both male and female sexes.

[2]http://www.livescribe.com/en-us/, accessed on Dec. 7, 2015.

Sketching Problem

The problem statement presented to the participant is as follows:

> You are a product designer in an engineering design firm. Recently, a luxury hotel chain has requested your firm to develop concepts for automating the process of ironing and folding hotel laundry, which presently consumes significant time and manpower. The client has informed that the solution will be most likely installed in typical hotel basements.

This problem is so designed mainly because of the following reasons:

1. The tasks required in the problem—ironing and folding—are often performed manually in everyday life, which would permit the designers to make progress with this challenge in a reasonable time and produce enough data for analysis.
2. Automated solutions to this problem are not very common, at least in the consumer market, which assures that it is indeed a "novel" design task for most designers. Post-experiment interviews confirmed that none of the designers were previously familiar with this problem.
3. The problem is likely to produce solutions with mechanical structures and mechanisms, which ensures a reasonable amount of mechanical sketch to be available for analysis.

Certain open ended-ness is intentionally included in the problem in order to foster creativity and a variety of solution principles. For example, it is not clarified if the laundry is only hotel linens and sheets or also the personal items of guests. The type of power and resources used is also not mentioned. However, the mention of the hotel basement might provide hints about the available resources such as water, electricity, steam, compressed air, etc.

Sketching Tasks

The participant is asked to perform the following three tasks in sequence upon reviewing the design problem:

- "Describe the actions that the system will have to perform. Go as detail as you deem necessary."
- "Develop sketches for solution concept(s) to the problem. Do as many and as detail as you deem necessary for capturing your idea(s) on paper."
- "Annotate your drawing(s) with leaders and labels. Go as detailed as you feel necessary for communicating the idea to your team members."

The first task is presented along with the problem statement, in the same sheet of paper. The subsequent tasks are revealed progressively, i.e., only upon completing the previous task, using separate sheets. The tasks are not numbered and the total count or duration of tasks is not discussed with the designer a priori. However, for the sake of reporting, they are referred in this paper as Task 1, Task 2, etc. This precaution ensures that the designer is not prompted to anticipate a later task and therefore be biased in her work.

Caution is taken not to restrict or guide the designer in a particular direction of output. For example, in Task 1, ‘describe’ is used instead of ‘list’, or ‘draw’, because using those words can prompt the designer to produce only textual or graphical descriptions of functions and therefore, bias her. Also, ‘actions’ is used instead of ‘function’ because many undergrad designers are not familiar with the word ‘function’ in engineering design context. Optional plural “(s)” is used whenever possible, for the same reason.

The designer is not given any time limit for the tasks and is asked to alert the experimenter when she finishes a task. Different designers are likely to solve design problems at different speeds. To ensure uniformity of data, it is important to let the designer finish the task at her pace, terminating only when she declares to have run out of ideas or patience. Inevitably, this method produces design output at various levels of completeness and quality, but that data allows valid examination of chunking under natural and unrushed ideation. At least one experimenter is present in the room with the designer in an unobtrusive manner at all times, and provides brief clarifications of the tasks—not the design itself—only upon requests.

Post-experiment Interview

Immediately after the designer completes all three tasks, she is briefly interviewed using three semi-formal steps. First, the designer is asked to read aloud her work and narrate her rationale. Second, the interviewer identifies significant features in the sketch, which are portions of the sketched structure that has notable geometric character or working principles. Finally, the interviewer asks the designer “if there is any prior knowledge, experience, or memory that could have inspired that feature”. The interview is video recorded. The purpose of this interview is to investigate the role of analogies in cognitive chunking. Analogical reasoning, esp. the transfer of a solution stored in memory to a new problem at hand, is believed to be a core process of creativity (Goel and Bhatta 2004; Goel 1997). Recent studies in design creativity examined the attributes of artifacts that are used as the basis of detecting similarities between the problem and solutions in memory, e.g., similarity based on functions, working principles, flows properties, or motions, etc. (Arlitt et al. 2013). The interview is performed to investigate if and how the new chunks formed during ideation are related to the old chunks in memory. However, this interview data is not used to support the conclusions presented here and are reserved for future publication.

After the interview, the designer fills a post-experiment questionnaire which captures the designer’s familiarity with the activity of manual ironing and with the design problem presented. Data from a designer unfamiliar to ironing or familiar to the design problem would be rendered invalid and would not be included in analysis. However, no such case occurred in the study. This step concludes the experiment procedure.

Data Collection

The designer is instructed to complete the three tasks using a Livescribe™ smart pen—a commercial[3] product that is a mini computer housed in a pen so that it can be used normally to write on paper, but records video of each pen stroke and surrounding audio, with the ability to play them back when needed. Thus, a true-to-time re-enactment of each pen stroke committed to paper by the designer and the conversation during the interviews is possible, which permits a time study of the design. In addition, one video camera is placed unobtrusively in the room to collect data about gestures and body language, and it is used as a secondary recording device for the interviews.

Sample Raw Data

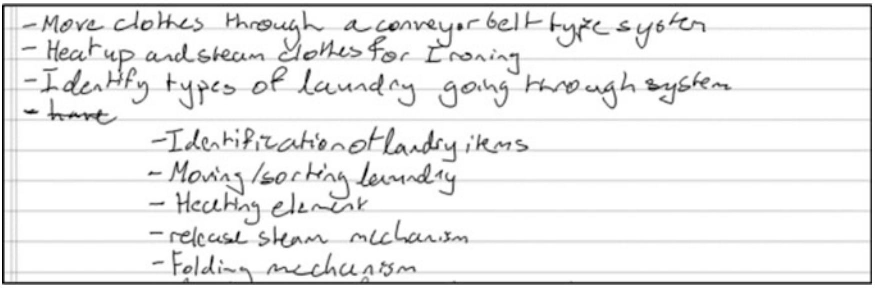
Figure 1 shows a portion of the list of requirements produced in response to Task 1 by one designer. Figure 2 shows a sample of sketches produced in Tasks 2 and 3 by one designer.

The highlighted elements, such as sketch elements and leaders are explained in the next section. The designers produced between 1 and 6 pages of handwritten material for the three tasks, and the sample shows only a portion of the work. A majority of designers (six out of nine) produced three or four pages each, without time limits, and the three other designers produced one, five, and six pages, respectively. Time of designing varied in a comparable range as the page counts: between 4:24 and 29:07 mm:ss for Task 1, and between 9:26 and 40:53 mm:ss for tasks 2 and 3 combined.

Data Analysis Protocol

A protocol is used to codify the smart pen data into a spreadsheet to facilitate objective and repeatable analysis. The design and reliability of this protocol is discussed in the next section. Below, the protocol steps are discussed.

[3]Disclaimer: The authors neither endorse nor recommend the use of this product.



- Move clothes through a conveyor belt type system
- Heat up and steam clothes for Ironing
- Identify types of laundry going through system
- ~~have~~
    - Identification of laundry items
    - Moving/sorting laundry
    - Heating element
    - release steam mechanism
    - Folding mechanism

Fig. 1 Sample of list of actions produced in Task 1

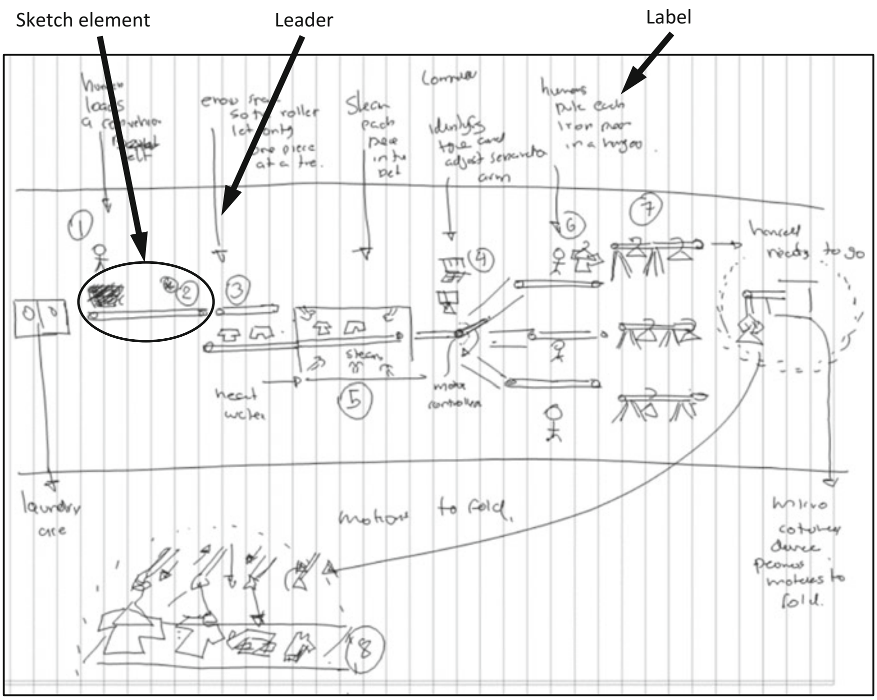


Fig. 2 Sample of design in response to Tasks 2 and 3

Vocabulary of Entity Types

The analysis is based upon three types of entities—sketch elements, leaders, and labels—found in the pilot data from two designers, as defined in Table 1. A fourth type, called 'name', is a subtype of labels. Entities comprise of <u>*pen strokes*</u>, which is defined as a mark resulting from touching the pen to the paper, scribing a mark with it, and lifting the pen.

Table 1 Vocabulary of entity types and their definitions

Entity	Definition
Sketch element	A sketch element is the whole or part of a pen stroke that comprises the sketch of a physical structure of the solution. In Fig. 2, the highlighted entity is a conveyor belt of chain running on a pair of pulleys or sprockets
Leader	A leader is a set of one or more pen strokes used to point to portions of a sketched structure in order to attach a label to it
Label	A label is a text that is used to annotate a sketch element or a portion of a sketch using a leader and is usually placed at the end of the leader
Name	A name is a special type of label used specifically to assign a name to a sketch element or a portion of the sketch

This classification of entities applies to the portions of the design that show physical structures in free-hand sketches, which occur typically in Tasks 2 and 3, and in some exceptional cases, in Task 1. The output of Task 1 is typically free-hand text and no classification is needed for it. The purpose of this classification is to separately track the entities used to draw the sketch and the names used to annotate them, so that time study of the sketching actions can be performed precisely. The following discipline is followed when counting the various entities:

1. **Sketch elements**: To count the sketch elements in a cluster, the cluster is treated as if it was created using a common CAD sketcher. Each CAD curve is then counted as one sketch element. For example, a line counts as one element, a triangle amounts to three elements, and an oblong (e.g., a round-ended slot) has four elements: two semi-circles and two lines.
2. **Leaders**: A leader can be a straight, curved, or squiggly pen stroke, or a collection of multiple strokes, some of which may comprise an arrow head added to a stem. In such cases, the whole set of pen strokes counts as one entity of type leader. In Fig. 2, the highlighted leader is made of three different pen strokes.
3. **Labels**: Designers are likely to lift the pen periodically while writing the letters and words of a label. Thus, most labels consist of multiple pen strokes, but they still count as one entity of type label, irrespective of length, until interjected by a pause of one second or more. In Fig. 2, the highlighted text "humans put(s) each iron(ed) piece in a hanger" counts as one label, because it was written with multiple pauses, none of which was a second long.
4. **Non-design strokes**: Pen strokes that are not related to the design are not classified. These include doodles at the corner of the page. The smart pen does not recognize difference in pressure at the nib and thus, strokes used to glide the pen on the paper are detected as full-pressure strokes. A review against the physical page is used to discern these strokes. While these strokes could indicate some mental activity, they were not included in the encoding scheme. In the reported study, the time spent in adding non-design strokes is an insignificant portion of the total design time of the participant (10 s or less, when present) and thus, this simplification is not likely to impact the overall outcome of the study.

Encoding Steps

1. **Playback**: The pen stroke data is played back on a computer using a proprietary software with a temporal resolution of one second.
2. **Familiarization**: The entire video is played back twice. In the first run, the coder watches the video without attempting to encode, but notes the organization of the data, e.g., the break points between the Tasks 1, 2, and 3, and patterns in the designer's activities that are visually identifiable. Sometimes designers go back to previous pages or overwrite on previous pen strokes, etc. This first step familiarizes the coder with the data, which facilitates efficient navigation through the data and error-free encoding. It takes up to 3 h to encode 30 min of design data and the number of individual pen strokes for the nine designers range from 1306 to 2848. This familiarization is therefore important.
3. **Entity encoding**: In the second run, the coder plays back the recording for Tasks 2 and 3 and looks for pauses in the pen stroke activity, where the pen does not touch the paper for one second or longer, as seen from the timer of the video player. Upon detecting such a pause, each pen stroke between two successive pauses is said to form a cluster and is assigned a unique serial number in the order in which it was added to the page. The resulting data is captured in a spreadsheet, with one cluster or one pause per row, as shown in Fig. 3.
   For example, the third row (Cluster 2) indicates that five pen strokes—serial numbers 2 through 6—were created without taking a pause of one second or longer. The start of stroke 2 occurred when the playback timer read 00:13 mm:ss (Pen-Touch column) and the end of stroke 6 occurred at 00:16 mm:ss (Pen-Lift column). The columns under ID indicate the serial numbers of these strokes. Since these five strokes are added without a one-second pause, they are recorded as a cluster, as shown in the Cluster column. The fourth row (one below Cluster 2) records the start and end timestamps of the pause that immediately followed the five strokes in Cluster 2. Consequently, this row shows no strokes under ID. The duration column records the difference between the start and end timestamps in mm:ss units. Accurate detection of a one-second

Clusters	Pen-Touch	Pen-Lift	Duration	ID																
1	00:10	00:12	00:02	1																
	00:12	00:13	00:01																	
2	00:13	00:16	00:03	2	3	4	5	6												
	00:16	00:19	00:03																	
3	00:19	00:25	00:06	7	8	9	10													
	00:25	00:28	00:03																	
4	00:28	00:46	00:18	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
	00:46	00:47	00:01																	
5	00:47	00:51	00:04	28	29	30	31	32	33											
	00:51	01:04	00:13																	
6	01:04	01:09	00:05	34	35															

Fig. 3 First eleven rows of the data encoding spreadsheet from designer G1

pause may take practice, and when in doubt, the coder reexamines the pause. Playing back the video at a reduced speed (0.5× or 0.33×) may improve this accuracy.

4. **Entity color coding**: Next, the entities in each cell is color-coded separately for sketch element, leader, or label.

Design Time Study

For each designer, data from the encoding spreadsheet in Fig. 3 is translated into a time study graph, as shown in Fig. 4. Time is plotted in the horizontal direction and number of entities is along the vertical. Each column in this chart represents a cluster, while the gaps between columns are the pauses. The height of each column is the number of entities in the cluster, which are divided into the three types: sketch element, leaders, and labels. This graph provides a visual rendering of the design activities, true to scale on both axes. The first six columns of Fig. 4 can be verified to be the first six clusters from Fig. 3. It is reiterated that this data only reflects activities in Tasks 2 and 3.

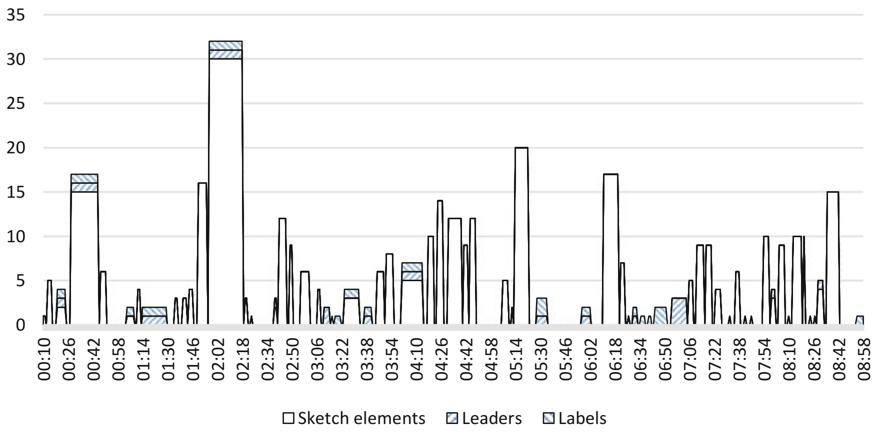


Fig. 4 Time study of design actions for designer G1 in Tasks 2 and 3

Table 2 Experimental results

	1	2	3	4	5	6	7	8	9	10	11	12
	Designer	# Pen strokes	# Pages	Total duration (mm:ss)	# Entities	# Sketch elem	# Leaders	# Labels	# Clusters	# Name-D	# Name-ID	# Names
1	G1	1485	4	13:39	650	578	36	36	104	10	1	11
2	G2	2529	6	40:53	791	584	100	107	196	35	4	39
3	G3	2513	4	24:41	274	170	34	70	125	12	2	14
4	UG2	1490	3	47:36	422	285	48	89	181	13	7	20
5	UG3	2318	3	14:24	195	120	17	58	76	10	12	22
6	UG4	1306	1	17:13	211	125	22	64	111	15	0	15
7	UG5	3732	4	20:44	428	314	45	69	107	26	1	27
8	UG6	1339	2	09:26	151	99	5	47	79	4	3	7
9	UG7	2848	5	35:14	373	226	20	127	212	17	0	17
10	G Avg.	2176	4.67	26:24	572	444	57	71	142	19	2	21
11	UG Avg.	2221	3.14	24:11	293	191	27	75	127	14	4	17
12	All Avg.	2173	3.56	24:52	388	278	36	74	132	16	3	19
13	Max	3732	6	47:36	791	584	100	127	212	35	12	39
14	Min	1306	1	09:26	151	99	5	36	76	4	0	7

Results

Table 2 shows the data collected from the nine designers, grouped as graduate and undergrad students. In the Designer column, the graduate students are named G1 through G3 (rows 1–3) and the undergrads are named UG2[4] through UG7 (rows 4–9). The group-wise average, total average, and range of the data are shown in the last five rows. Columns 2 through 4 show the basic data from the smart pen output. Columns 6 through 8 show the counts of the three entity types for each designer, which totals up to the data in column 5 (Entities). The last four columns show the number of clusters and names. Column 10 (Name-D) counts the names of sketched structures that are visible **d**irectly as labels next to the sketch produced in Tasks 2 and 3. In Fig. 2, the word "hanger" in the highlighted labels is a Name-D, as it points to the sketch elements that describe a hanger. Name-ID (Column 11) is a name that is not found directly as a label in Task 3, but is found **ind**irectly, i.e., written by the designer in the list of actions in Task 1 and later drawn in the sketch in Task 2. As seen in the table, Name-IDs are a small portion of the total count of names.

Observations, Analyses, and Conclusions

The following observations are made on the data:

- **Observation: Clustering of pen activity is clearly visible**. The time study of pen strokes (example in Fig. 4) illustrates that design output during ideation sketching is produced in intermittent clusters separated by pauses, rather than in continuous or evenly-spaced series of strokes. The cluster-pause-cluster pattern is clear from the data. Psychology research asserts that pauses of 300 ms to 3 s followed by motor activity imply mental processes leading to that activity (Goel and Bhatta 2004). The intermittent pen pattern then suggests that the entities in the cluster following a pause were conceived during the pause itself and that they were conceived at once, as a single packet of information, so as to permit transferring them to paper with needing a pause in between. This pattern agrees with the notion of chunking and thus a cluster could be indicative of a cognitive chunk. The pause length of one second is used in this study since that is the lowest interval measurable by the video player, which falls within the 300 ms–3 s range.

[4]The data for an undergrad designer named UG1 is excluded, as she had prior familiarity with the problem statement and analysis protocol. Thus, the name of the first reported is undergrad designer is UG2. .

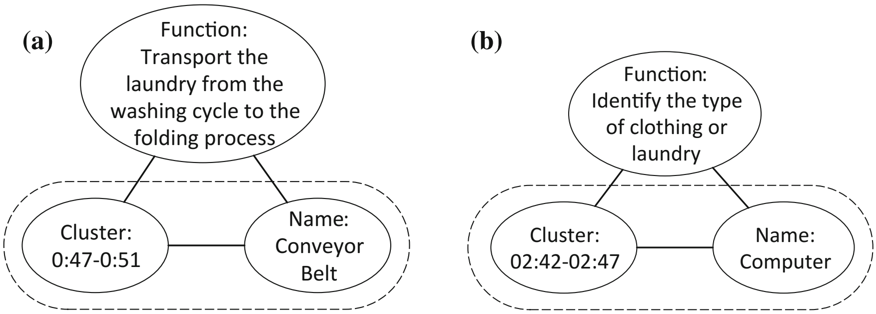


Fig. 5 Examples of diads and triads for designer G1

- **Observation: Name-Cluster diads**. When the names assigned to the various features of the sketched solution are compared to the clusters, it is clear that these two parameters are strongly correlated. A correlations is said to occur when the designer assigns a written name to a feature or a portion of the drawn structure and all the pen strokes used to draw that named feature are found to occur in a single cluster. Thus, diads of names and clusters can be identified, as shown in Fig. 5 (a and b). In each figure, the bottom left bubble represents a cluster with begin and end time stamps. The bottom right bubble is a name explicitly written by the designer as a label. The diad relation, indicated by the dotted oblong, means that the named feature, e.g., the conveyor belt in (a), was drawn entirely within the cluster 0:42–0:47 mm:ss, and nothing else was drawn within that cluster.

With the exception of UG6, the number of diads produced by designers varied between 10 and 35. Diads were formed at a rate of up to 2 per minute and the average rate was 1.23 (Table 3).

The formation of diads further illustrates the presence of chunking in ideation sketching. The features to which names are assigned are not provided to the designer for naming. Instead, the designer creates them with the necessary details and names them at a certain level of abstraction. For example, some features named by designers are: clamp, hanger, folding table, de-clutter, and conveyor belt, etc. (Fig. 6). Each of these features is created with multiple sketch elements and sometimes using sub-feature-level details. But the names are assigned not to the lower level details (e.g., the frame, legs, and flaps of the folding table), but to a suitable higher level form. This behavior implies that the designer perceives the

Table 3 Diad generation and rate for all designers

	G1	G2	G3	UG2	UG3	UG4	UG5	UG6	UG7	Avg.
# Of diads	10	35	12	13	10	15	26	4	17	16
Diads/min	1.42	1.76	1.18	0.88	1.11	1.10	2.03	0.88	0.72	1.23

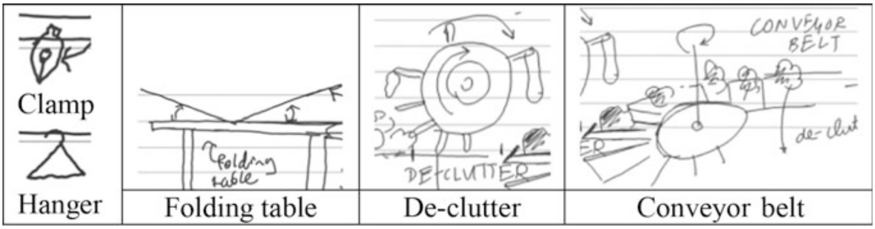


Fig. 6 Examples of naming at higher levels of abstraction

abstraction level at which names are assigned to be the unit chunks of information. The diad relation between names and clusters then reinforces that chunking occurs during ideation sketching, and that they can be visualized using the clustering and naming of sketch elements.

- **Observation: Name-Cluster-function triads.** As shown in Fig. 5, for most of the diads found, the name-cluster diad has a correlation with a function mentioned by the designer in Task 1. For example, the conveyor belt shown in the first diad of Fig. 5 is used in the design for a specific transportation function, and that function is written by the designer in Task 1. These triads are found in the works of all nine designers, for more than 50% diads for each. This observation indicates that the cognitive chunks indicated by the diads are formed on the basis of the functional needs perceived by the designer.

Secondary Observations

Overall, the data shows a higher amount of design output from the graduate students than the undergrads, as can be verified by comparing the graduate and undergrad averages in the highlighted cells of the table (columns 3, 5, 6, 7, 12). However, this trend is not sufficient to conclude that graduate students typically produce more design output, because: (1) the data pool is not statistically significant to make such a conclusion, and (2) the highest numbers in each of these five columns (# 3, 5, 6, 7, 12) come from the same graduate student, who is clearly an outlier. Out of the nine designers, only two (G2, UG2) drew a function graph with functions and flows in response to Task 1. The others made function lists, without drawing a graph. Only two designers drew a structure within Task 1, where functions were sought. However, all nine designers drew leaders and labels within Task 2, where only the structure was sought.

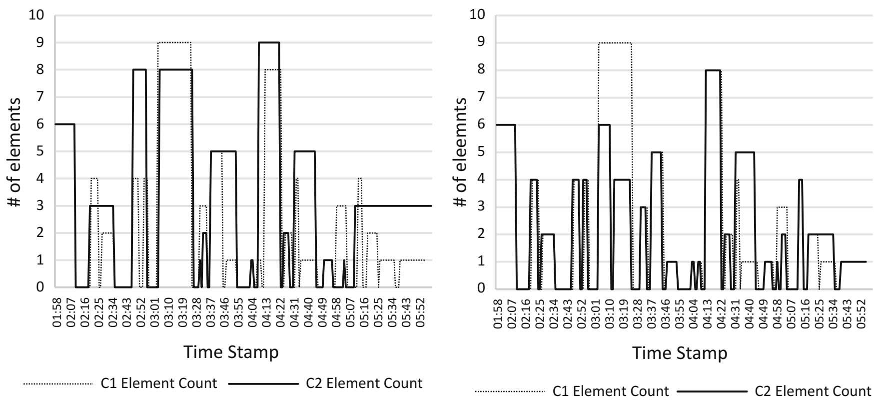
Reliability of Analysis Protocol

Experimental data that are encoded by humans, even with a strict protocol, are prone to errors. Inter-coder reliability of the protocol is evaluated to establish credibility of the conclusions and to assure that the protocol is not sensitive to the skill or training of future researchers who adopt it.

In the pilot phase, the protocol was applied by two coders (C1, C2) on the same 4-min span of data. Figure 7a shows the comparison between their time studies obtained without training or reconciliation between coders. Upon inspection, the omission of valid pauses by C2 was found to be the main cause of the disagreement. In Fig. 7a, C1 detected pauses at approximately 02:26, 02:54, 03:45, 04:11, and 04:35, which C2 omitted. In response, C2 was given more training and practice. A few refinements, such as reexamining doubtful pauses and playing back the video at reduced speed, were added to the protocol for both coders. Figure 7b shows the time study of the same 4-min span, redone by C1 and C2 after this reconciliation. The improvement is visually recognizable. The reliability of the coders continually improved through multiple iterations, as seen next.

During the actual experiment, all nine data points was encoded by C1, to ensure uniformity. To ascertain continued adherence to the protocol, C2 encoded select 5-min segments from three designers picked at random. Figure 8 shows the resulting comparison for one designer (G1). The error between the coders is visibly reduced. The impact of the occasionally persistent errors on the analysis is minimal, as explained next. The errors are classified into three types (Table 4). Their examples are shown in Fig. 8.

In Fig. 8, twelve of the total 39 columns (clusters) reported by C1 have a difference of height between coders, resulting into E1 = 12/39 = 31%. The average



(a) Before reconciliation (b) After reconciliation

Fig. 7 Coding comparison between C1 and C2 showing improvement through reconciliation and training during the pilot phase

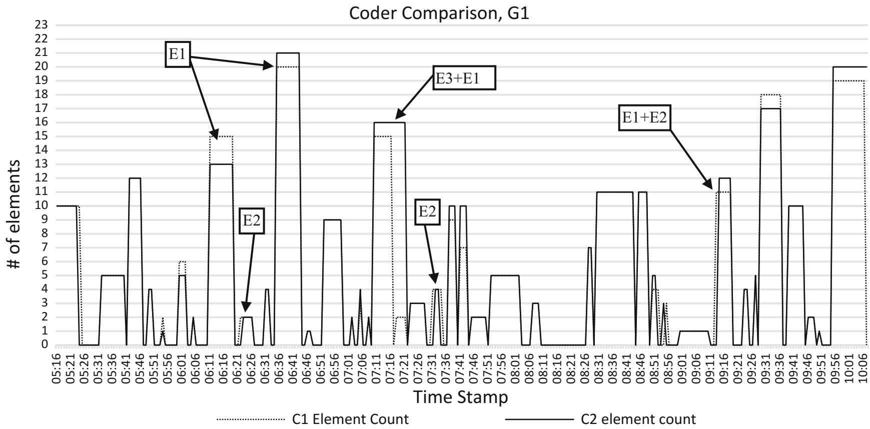


Fig. 8 Coder comparison on a sample of data from designer G1

Table 4 Types of errors between coders

Error	Visible effect in Fig. 8 and cause
E1	Difference in column height (without column shift): results from difference in counting entities in a cluster, while the start and end times are identical between coders
E2	Column shift (without difference in column height): results from difference in registering the start and/or the end time of a cluster, while the coders agree on the entity count in the cluster
E3	Column split (without difference in column height or column shift): results from coders disagreeing about a pause. In Fig. 8, between 07:08 and 07:24 mm:ss, the apparent difference between the coders result from two effects: (1) C1 registered a pause that C2 omitted, and (2) C1 counted total seventeen entities, while C2 counted sixteen

value of those twelve errors is $E1_{avg} = 1.25$ and the maximum value is $E1_{max} = 3$, occurring between 08:04 and 08:12. Eight of those 39 columns show a difference in either the start or the end time, resulting into E2 = 8/39 = 21%. All eight occurrences of E2 are limited to a difference of 1 s between coders, thus: $E2_{max} = E2_{avg} = 1$ s. As the resolution of video player is 1 s, E2 errors, if present, could not be smaller than this value. The total number of columns (clusters) counted by the two coders is 39 and 38 respectively and thus, only one pause is omitted by one coder, resulting into E3 = (39–38)/39 = 2.6% only.

The errors are similarly counted for the other two data points (UG2, UG3) on which C2 performed a validation. The average errors over the three data points are shown in Table 5. The impact of these errors is discussed next.

Table 5 Average percent errors of encoding for G1, UG2, and UG3

Error type	Error frequency (%)
E1	28
E2	30
E3	7.6

Impact of Errors

Although the frequency of error types E1 (28%) and E2 (30%) are higher than that of E3 (7.6%), E1 or E2 errors can only affect the timing or size of a cluster between coders, while still maintaining the cluster itself and its relation with the diads and triads. Hence these errors have minimal impact on the findings of this paper. E3 has a higher potential impact on the study. Ignoring a valid pause (false negative) would promote a weaker correlation (type II) to a stronger one (type I). Detecting an invalid pause (false positive) would do the opposite and therefore together, they could impact the analysis. A low value of E3 (7.6%) indicates that coders strongly agree on the most important parameter of this study and the analysis can be trusted within this margin of error.

Closure and Future Work

This paper reports an experiment to study chunking in ideation tasks of engineering design. Based on an engineering sketching task performed by nine subjects, clear evidence of chunking is found. Further, the chunks are shown to be based on the functional needs of the design perceived by the designer prior to sketching. In the future, two other variants of this study will be performed to triangulate the current observations. **In the first variant**, the same experiment will be performed without first asking the designer to identify the functions, and thus eliminating the possibility of biasing her to think in terms of functions first. **In the second variant**, a copy-mode study will be performed, where designers will be asked to sketch an article shown in physical form or as a drawing, and the correlation between chunks and other aspects of the design (e.g., function, form, etc.) will be examined.

One limitation of the presented work is the argument that the designer is possibly predisposed to function-based thinking by performing Task1 first, in which case, it is trivial that the chunks during sketching are correlated to functions. However, since correlation is sought between chunks and pre-conceived functions, it is important to collect the functions before sketching. If the designers were asked to identify the functions after the sketching task, the study could still be limited since they would be more likely to identify actual functions of the structure just drawn rather than those needed by the design. Between these two difficulties, the reported design of the experiment is chosen because even in real-life design cases, predisposition to function is likely to occur. A method to observe the preconceived functions without asking the designer to write them should be developed.

Acknowledgements This research is supported by National Science Foundation Grant # CMMI-1532894. The authors are thankful to the participants of the study.

References

Arlitt R, Nix A, Sen C, Stone RB (2013) Discovery of mental metadata used for analogy formation in function-based design. In: ASME 2015 international design engineering technical conferences & computers and information in engineering conference, 2–5 Aug 2015, Buffalo, NY

Austin J, Delaney PF (1998) Protocol analysis as a tool for behavior aanalysis. Anal Verbal Behav 15:41–56

Cardella ME, Atman CJ, Adams RS (2006) Mapping between design activities and external representations for engineering student designers. Des Stud 27:5–24

Cross N (2001) Design cognition: results from protocol and other empirical studies of design activity. Design knowing and learning: cognition in design education. Elsevier, Oxford

Dinar M, Shah JJ, Cagan J, Leifer L, Linsey J, Smith SM, Hernandez NV (2015) Empirical studies of designer thinking: past, present, and future. J Mech Des 137(2):21101

Egan DE, Schwartz BJ (1979) Chunking in recall of symbolic drawings. Memory Cogn 7(2):149–158

Ericsson KA, Kintsch W (1995) Long-term working memory. Psychol Rev 102(2):211–245

Ericsson KA, Simon HA (1993) Protocol analysis: verbal reports as data. MIT Press, Cambridge, MA

Gero JS, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19(1):21–61

Gobet F (2001) Chunking mechanisms in human learning. Trends Cogn Sci 5(6):236–243

Gobet F, Simon HA (1996) Recall of rapidly presented random chess positions is a function of skill. Psychonomic. 3:159–163

Goel AK (1997) Design, analogy, and creativity. IEEE Exp 12(3):62–70. ISSN 0885-9000

Goel AK, Bhatta SR (2004) Use of Design patterns in analogy-based design. Adv Eng Inform 18:85–94

Miller GA (1956) The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychol Rev 63:81–97

Sen C, Caldwell BW, Summers JD, Mocko GM (2010a) Evaluation of the functional basis using an information theoretic approach. Artif Intell Eng Des Anal Manuf 24(1):87–105

Sen C, Summers JD, Mocko GM (2010b) Topological information content and expressiveness of function models in mechanical design. J Comput Inf Sci Eng 10(3):031003-031001

Sen C, Summers JD, Mocko GM (2012) A pilot protocol analysis studying how designers construct function structures. In: 5th International conference on design computing and cognition, DCC 2012, 7–9 June 2012, College Station, TX

Sen C, Mukhopadhyay A, Fields J, Ameri F (2014) An approach for measuring information content of textual engineering requirements using a form-neutral representation, asme 2014 international design engineering technical conferences & computers and information in engineering conference, 17–20 Aug 2014, Buffalo, NY

Smith SM, Gerkens DR, Shah J, Vargas-Hernandez N (2005) Empirical studies of creative cognition in idea generation, creativity and innovation in organizational teams. Lawrence Erlbaum, Routledge

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(1):33–52

Ullman DG, Wood S, Craig D (1990) The importance of drawing in the mechanical design process. Comput Graph 14(2):163–274

Waldron MB, Waldron KJ (1988) Conceptual CAD tools for mechanical designers. In: Proceedings of computers in engineering conference

A Systematic Review of Protocol Studies on Conceptual Design Cognition

Laura Hay, Chris McTeague, Alex H.B. Duffy, Laura M. Pidgeon, Tijana Vuletic and Madeleine Grealy

Abstract This paper reports the first systematic review and synthesis of protocol studies on conceptual design cognition. 47 studies from the domains of architectural design, engineering design, and product design engineering were reviewed towards answering the following question: *What is our current understanding of the cognitive processes involved in conceptual design tasks carried out by individual designers?* Studies were found to reflect three viewpoints on the cognitive nature of designing: design as search; design as exploration; and design activities. Synthesising the findings of individual studies revealed ten categories of executive and non-executive function studied across the viewpoints: visual perception; mental imagery; semantic association; long term memory; working memory; selective attention; creative thinking; evaluation and decision making; externalisation; and reasoning and problem solving. The review highlights several avenues for future research, centering on the need for general formalisms, more objective methods to supplement protocol analysis, and a shared ontology of cognitive processes.

Introduction

Conceptual design refers to the early stages of the design process, where design requirements and solutions may be fuzzy, unstructured, and/or ill-defined (McNeill et al. 1998; Goel 1995). Generating a high number of ideas during conceptual design is believed to increase the likelihood of achieving a desirable product in terms of cost and quality (Jin and Benami 2010). Thus, conceptual design may have a significant impact upon design performance later in the design process. However, there is a lack of clarity regarding the nature of the cognitive processes involved in this influential phase of designing. Jin and Benami (2010) note that whilst cognitive processes "are at the center in developing new ideas, they are rarely taken into account in research and development of design support methods and systems"

L. Hay (✉) · C. McTeague · A.H.B. Duffy · L.M. Pidgeon · T. Vuletic · M. Grealy
University of Strathclyde, Glasgow, UK
e-mail: laura.hay@strath.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_8

(p. 191). In a more recent study, Kim and Ryu (2014) claim that conceptual design involves "perception, problem solving, reasoning, and thinking about the design." Nonetheless, they argue that there is a need for thorough research "to better understand designers' internal cognitive processes" (p. 519). More generally, Dorst and Cross (2001) note that the internal mechanisms involved in creative idea generation are "mysterious (and often mystified)" (p. 425).

The Systematic Review Method

A plethora of empirical studies have been conducted on conceptual design cognition since the late 1970s, contributing a wealth of observations on different aspects of cognition and the factors affecting it in the early stages of design (Cross 2001; Dinar et al. 2015). However, there has thus far been only a single attempt to systematically review and synthesise the findings from these studies (Jiang and Yen 2009). This review focused largely on methodological aspects of protocol analysis; a similar exercise could provide greater clarity on the cognitive processes involved in conceptual design. Whilst Cross (2001) and Dinar et al. (2015) provide detailed and instructive treatments of empirical studies on design cognition, their reviews are broader in scope and cannot be considered to be systematic in nature. The systematic literature review is a research method whereby every quality publication on a particular topic is gathered and synthesised through a rigorous and transparent process. The intention is that all valid and reliable evidence relating to a particular phenomenon is considered by the researcher, minimising the potential for bias. Adherence to guidelines for conducting and reporting systematic reviews (e.g. the PRISMA guidelines (Moher et al. 2009) ensures that a systematic literature review is fully reproducible, and therefore meets the same standards typically applied to empirical research. Consequently, systematic reviews can provide a foundation for the development of formal theories and models of conceptual design cognition.

To address the need for greater clarity on the nature of the cognitive processes involved in conceptual design, this paper presents the findings of the first systematic review of protocol studies specifically focusing on conceptual design cognition. Studies from the domains of architectural design, engineering design, and product design engineering were considered. The review aimed to answer the following research question by synthesising the findings of individual studies: *What is our current understanding of the cognitive processes involved in conceptual design tasks carried out by individual designers?* We focused on protocol studies because unlike other empirical methods, protocol analysis is generally considered to provide direct access to a designer's cognitive processes (Dinar et al. 2015; Lloyd et al. 1995). Consequently, Cross (2001) notes that the method "has become regarded as the most likely method (perhaps the only method) to bring out into the open the somewhat mysterious cognitive abilities of designers" (p. 80).

Methods

Our approach was informed by the PRISMA guidelines for systematic reviews (Moher et al. 2009). The review was conducted by two researchers with a background in product design engineering, with input from a cognitive neuroscience researcher where required.

Review Scope

As stated previously, the review focused on the *conceptual design* phase of the design process. Only studies focusing on design tasks carried out by *individual* designers were included; studies examining group-based tasks were not considered. Studies from three design domains were included in the sample:

- *Engineering design*, i.e. the design of technical products, with a primary focus on relatively complex functional requirements (Hubka 1982).
- *Product design engineering*, i.e. the design of products involving a combination of functional requirements and requirements for aspects such as form, aesthetics, usability, ergonomics, and marketing/branding issues (Roozenburg and Eekels 1994; Kim and Lee 2010).
- *Architectural design*, i.e. the design of buildings, their interiors, and their surroundings. Like product design engineering, architectural design may involve a combination of functional and non-functional requirements (Akin 1986).

Findings from studies originating in these domains are argued to be comparable, and therefore conducive to synthesis, on the basis that they adopt similar: (i) views on the nature of designing, i.e. designing is generally considered to involve identifying a function to meet a need, developing behaviours to fulfil the function, and synthesising structures to exhibit the behaviours (Gero 1990); and (ii) paradigms for describing design cognition, i.e. the problem solving and reflective paradigms as discussed by Dorst and Dijkhuis (1995).

Search Strategy and Article Selection Process

A flowchart outlining the article selection process and the number of articles included/rejected at each stage is presented in Fig. 1. Literature was initially gathered through searches of major engineering and psychology databases (e.g. Compendex, Technology Research Database, and PsycINFO) conducted between 27th March 2015 and 3rd April 2015. Search terms are presented in Fig. 2. Following removal of duplicate articles, a range of study types were identified within the corpus e.g. controlled experiments, protocol studies, case studies, and

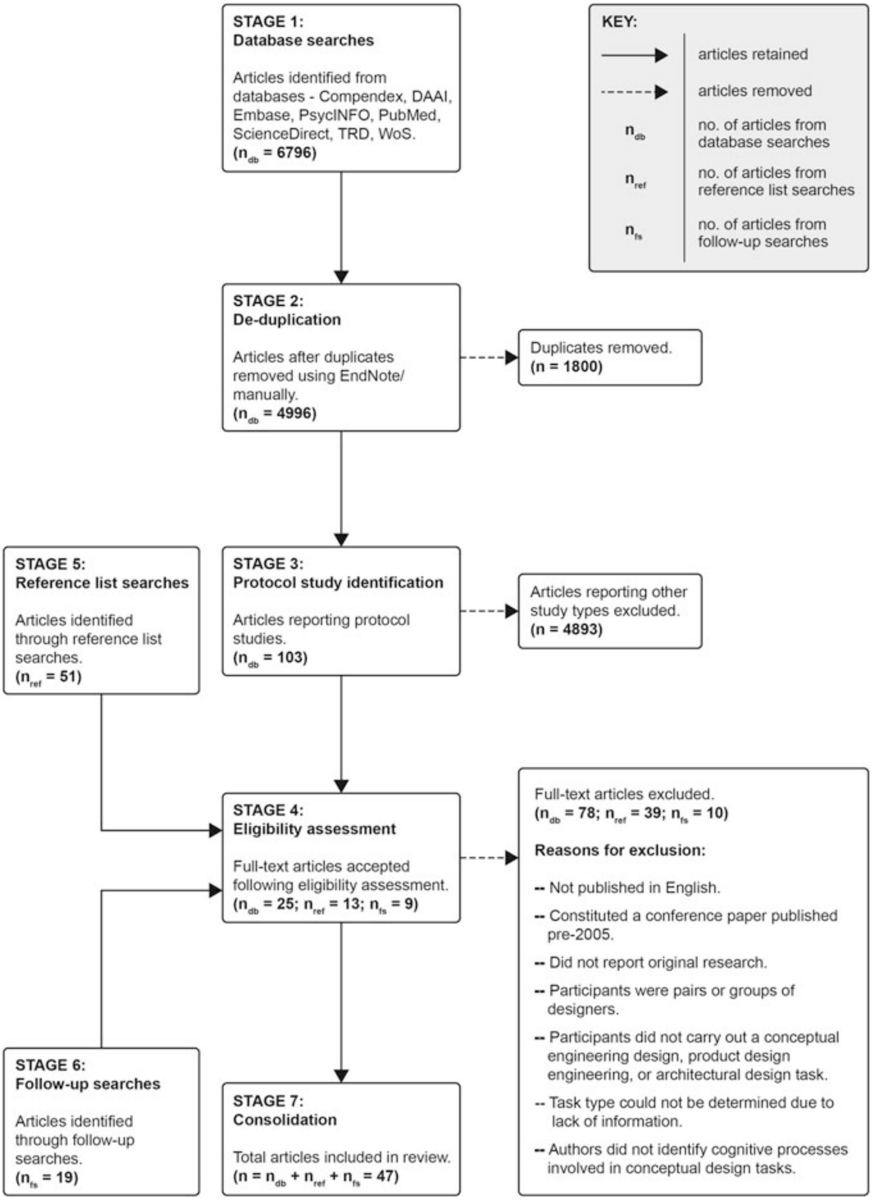


Fig. 1 Article selection process (adapted from Moher et al. 2009)

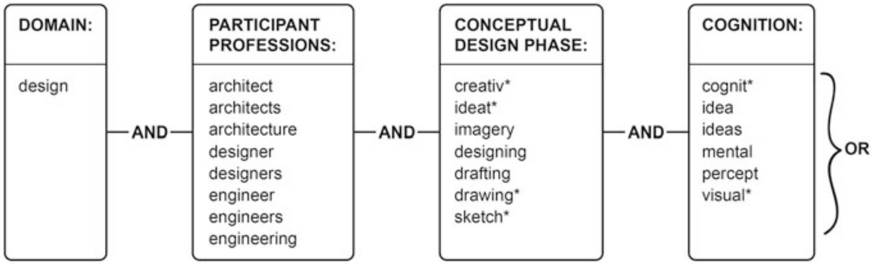


Fig. 2 Search terms and search structure

Table 1 Inclusion criteria

No.	Criterion
1	Article must be published in English
2	If constituting a conference paper, article must be published during or after 2005
3	Article must report original research
4	Study participants must be individual designers, i.e. not pairs or groups
5	Study participants must carry out a conceptual design task within the domains of engineering design, product design engineering, or architectural design
6	Authors must identify cognitive processes involved in a conceptual design task

surveys. It was decided to focus the review on protocol studies for reasons stated in the introduction. These underwent eligibility assessment against six inclusion criteria (Table 1); the reference lists of included articles were then manually searched to identify further candidates for inclusion. Additional candidates were also identified from follow-up database searches using terms reflecting the protocol analysis method (run on 9th October 2015). All candidates for inclusion underwent eligibility assessment against the criteria in Table 1. Note that conference papers published prior to 2005 were excluded on the basis that they largely constituted early versions of research that was later re-published and updated in a journal article, e.g. Suwa and Tversky (1996, Suwa et al. (1998a, 1994). Eligible articles from initial database searches, reference list searches, and follow-up database searches were then consolidated to produce a final set of 47 articles for inclusion in the review.

Characteristics of Reviewed Studies

Studies included in the review are denoted with * in the reference list at the end of this paper. Owing to space limitations, it was not possible to include the full sample in this paper. Key characteristics of the studies may be summarised as follows:

- The oldest study was published in 1984 (Akin 1984), and the newest in 2015 e.g. (Yu and Gero 2015). 53.2% of studies were published in the last 10 years.

- 76.6% of the articles reported full protocol studies, and 23.4% reported analyses of existing protocol data.
- Studies involved a total of approximately 350 participants, ranging from a minimum of 1 to a maximum of 36 participants per study and an average of 7 participants. Participants included practicing designers and architects, along with students at undergraduate, Master's, and Ph.D. level.
- The experience levels of participants ranged from 0 to 38 years, although authors were observed to apply inconsistent definitions of "experience."
- 45 distinct design tasks were studied—44.4% architectural design, 42.2% product design engineering, and 13.3% engineering design.
- The following types of data were gathered by authors: concurrent verbalisations during tasks (68.1%); retrospective verbalisations after tasks (23.4%); combined current and retrospective verbalisations (8.5%); video of designer behaviour (84.4%); and physical sketches (51.1%).
- The length of verbal protocols ranged from 15 to 600 min.

Focus of Reviewed Studies

Collectively, studies were found to reflect three viewpoints on the cognitive nature of designing: (V1) design as search; (V2) design as exploration; and (V3) design activities. A range of focus areas were then identified from studies associated with each viewpoint, as summarised in the sub-sections below. Note that the research question is answered later in the paper.

Viewpoint 1: Design as Search

The first viewpoint reflected in the studies considers designing to constitute a goal-directed search process. Central to this viewpoint is the perspective that a designer may be modelled as an information processing system (IPS) (Chan 1990; Stauffer and Ullman 1991). As an IPS, a designer transforms information from input to output states via the execution of elementary information processes known as *operators* (Stauffer and Ullman 1991). These transformations may be termed *state transformations* (Goel 1995; Akin 1984).

Stauffer and Ullman (1991) suggest that during designing, the processor component of the IPS "accesses information from the LTM [long term memory] into the STM [short term memory, i.e. working memory] as it is needed by the operators" (p. 117). Operators then act on the retrieved information in working memory to effect a transformation of the design state, which is proposed to be manifested initially as changes in the information content of working memory. Drawing from the work of Newell and Simon (1972), Chan (1990) highlights that the sequence of operators and state transformations involved in conceptual design may be formalised as "a search through […] knowledge states" (p. 64) i.e. a search process. Studies on design as search were found to reflect three focus areas, which are briefly summarised below.

Search Processes and Their Context in Problem Spaces

Search processes are considered to be bounded by a problem space. The problem space encompasses knowledge of the initial problem state, a goal state, and all possible design states in between these two states. Design as search may then be viewed as a sequence of state transformations, beginning with the problem state and proceeding through intermediate design states until the goal state is reached (Goel 1995; Chan 1990; Stauffer and Ullman 1991; Newell and Simon 1972). Transformations may be (i) lateral, i.e. generation of a new solution, or (ii) vertical, i.e. elaboration of a current solution (Goel 1995). Design problem spaces are typically large owing to the ill-defined nature of design problems; however, a designer may reduce the size of the space to be searched by implementing constraints (Goel 1995; Chan 1990). Search processes are further managed with design goals specifying desired design states (Chan 1990; Stauffer and Ullman 1991).

Search Methods, Control Strategies, and Knowledge Schema

Operators may be applied in different combinations and sequences to reach a solution and manage the search process. These patterns of operator execution are described as search methods (Stauffer and Ullman 1991) and control strategies (Chan 1990), respectively. Owing to the necessary space limitations of a conference paper, the full range of search methods, control strategies, and operators identified from the reviewed studies cannot be presented here. Chan (1990) proposes that operators are retrieved from knowledge schema stored in long term memory. That is, networks of knowledge units encapsulating both declarative and procedural knowledge (Chan 1990; Ball et al. 2004).

Problem Solving Phases

Several authors were observed to delimit designing into problem structuring and problem solving (or search) phases. Akin (1984) suggests that designers structure the problem during a pre-sketching phase, and solve it during a "search for design" phase (p. 204). Similarly, Goel (1995) argues that problem structuring may occur at the start of design tasks; however, it may also "recur periodically as needed" (p. 114). This is supported by Chan (1990), who suggests that problems may be restructured during tasks in response to a "critical problem situation" (p. 69), e.g. a decision to abandon a solution.

Viewpoint 2: Design as Exploration

A second viewpoint reflected in the studies considers designing to constitute an exploratory process operating between problem and solution spaces. Central to this viewpoint is the perspective that design problems are evolutionary in nature—i.e. they may be reinterpreted and reformulated as designing progresses and a solution is developed (Maher and Tang 2003; Jin and Chusilp 2006).

When design problems are viewed as evolutionary, the designer's task environment may be subdivided into (i) a problem space, encompassing knowledge of design requirements; and (ii) a solution space, encompassing knowledge of design solutions (Maher and Tang 2003). Rather than a search process, design may then be viewed as an exploratory process operating within and between these two spaces, where actions taken in the solution space (e.g. idea generation or concept development) may influence actions taken in the problem space (e.g. problem structuring or reformulation) and vice versa (Dorst and Cross 2001). Interactions between the two spaces may also be understood in terms of the concept of situatedness, e.g. the notion that a designer's understanding of a problem is affected by what they draw and perceive in their sketches, and vice versa (Suwa et al. 2000).

Studies on design as exploration were found to reflect two broad focus areas, namely co-evolutionary design and sketch-based design exploration. Sketch-based exploration may be further subdivided into the following areas: visual reasoning; cognitive actions; and unexpected discoveries and situated invention. Each area is briefly summarised in the following paragraphs.

It should be noted that a number of authors have also examined the role and significance of sketching in conceptual design, e.g. (Athavankar 1997; Bilda et al. 2006; Bilda and Gero 2007; Athavankar et al. 2008). Whilst several of these studies conclude that sketches act as a form of 'external memory' that serves to offload a designer's visuo-spatial working memory (Athavankar 1997; Bilda and Gero 2007), the majority also demonstrate that designing using mental imagery alone may still result in satisfactory design outcomes.

Co-evolutionary Design

The co-evolution model proposed by Maher et al. 1996 (cited in Dorst and Cross 2001) formalises the problem and solution spaces outlined above. Designing is described as a co-evolutionary process—that is, according to Maher and Tang (2003), a process that "explores the spaces of problem requirements and design solutions iteratively" (p. 48), resulting in the evolution of design problems alongside solutions. The solution space provides the basis to evaluate/re-evaluate requirements in the problem space, and the problem space facilitates evalution of solutions proposed in the solution space. Interactions between the two spaces "may add new variables into both" (p. 48), e.g. new design requirements in the problem space or potential solutions in the solution space. That is, the interactions may change the focus of designing.

Visual Reasoning

Goldschmidt (1991) proposes that during sketch-based design tasks, designers follow a pattern of "dialectical reasoning" about the visual features of the sketch in relation to the design problem (p. 139). That is, a pattern of visual reasoning that continually shifts between two modes: (i) *seeing as*, i.e. proposing properties/attributes that a design could possess based on e.g. metaphors and analogies; and (ii) *seeing that*, i.e. developing a rationale for design decisions relating to these proposals. Park and Kim (2007) model visual reasoning in terms of three interrelated cognitive activities: (i) *seeing*, i.e. the perception, analysis, and interpretation of visual information in external representations; (ii) *imagining*, i.e. the synthesis of perceptual and conceptual information produced by seeing in order to generate and transform new internal representations; and (iii) *drawing*, i.e. the evaluation, confirmation, and externalisation of internal representations.

Cognitive Actions

Suwa et al. (1998b) propose a set of "cognitive actions" intended to comprehensively capture a designer's cognition during sketching tasks. These are organised into physical, perceptual, functional, and conceptual categories. The categories are partly based on the work of Suwa and Tversky (1997), and are argued to "correspond to the levels at which incoming information is thought to be processed in human cognition" (p. 459). That is, sensorily (physical actions), then perceptually (perceptual actions), and finally semantically (functional and conceptual actions). Suwa et al. (1998b) claim that the proposed actions are supported by "an enormous amount of concrete examples" (p. 458) identified from the protocol of an architect studied by Suwa and Tversky (1997).

Unexpected Discoveries and Situated Requirements Invention

Suwa et al. (2000) propose that during sketching tasks, designers may unintentionally create spatial relations between elements. Visuo-spatial features created by these relations may then be "discovered in an unexpected way" later in the design task (p. 540). Suwa et al. (2000) found that unexpected discoveries are often followed by cognitive actions to set up goals focusing on new issues, which in certain cases "become general enough to be carried through the entire design process as one of the primary design requirements" (p. 547). This is termed "situated-invention" by the authors (p. 540). Unexpected discoveries and situated invention were observed to be correlated bi-directionally, i.e. unexpected discoveries appear to drive situated invention and vice versa over the course of a sketch-based design task.

Viewpoint 3: Design Activities

The final viewpoint reflected in the studies considers designing to constitute a cognitive activity that may be decomposed into sub-activities that occur in particular patterns (Hubka 1982; Hubka and Eder 1996). In this respect, a number of authors in the broader design literature may be seen to outline classifications of design activities, e.g. Hubka (1982) formulates a hierarchy of design activities occurring at different points in the design process, and Sim and Duffy (2003) propose an ontology of design activities organised into different categories.

Four major design activities associated with conceptual design were found to be studied by authors in the sample, namely: problem analysis; concept generation and synthesis; and evaluation. The nature of each activity is briefly outlined below. Authors were observed to study the activities both individually (Liikkanen and Perttula 2009) and as part of studies on cognitive models (Jin and Chusilp 2006) and activity patterns during the design process (McNeill et al. 1998).

Problem Analysis

Problem analysis involves understanding the design problem, setting goals, and defining constraints and requirements (Jin and Chusilp 2006). Authors were found to highlight a number of sub-activities involved in problem analysis, namely: information gathering (Kim et al. 2005); inference (Eckersley 1998; Lloyd and Scott 1994); problem decomposition (Goel 1995; Liikkanen and Perttula 2009; Lloyd and Scott 1994; Lee et al. 2014); identifying, exploring, clarifying, and prioritising constraints and requirements (Kim et al. 2005; Lane and Seery 2011; Daly et al. 2012); and problem reframing (Akin and Akin 1996).

Concept Generation and Synthesis

Concept generation may be positioned as involving the generation of ideas or partial solutions, and the synthesis of these into more mature or complete concepts (Jin and Chusilp 2006). However, in certain studies, idea generation and synthesis are treated as separate activities (Jin and Chusilp 2006; Kruger and Cross 2006). In addition to synthesis, authors were found to highlight further sub-activities involved in concept generation, namely: memory retrieval (Jin and Benami 2010; Lane and Seery 2011); association/analogical reasoning/case-based reasoning (Jin and Benami 2010; Yu and Gero 2015; Ball et al. 2004; Daly et al. 2012; Chiu 2003); and the generation/transformation/ maintenance of internal representations (Jin and Benami 2010; Park and Kim 2007; Lane and Seery 2011).

Evaluation

Evaluation entails the assessment of concepts against design requirements, constraints, and criteria (Jin and Chusilp 2006) typically defined during problem analysis (Kruger and Cross 2006). Authors were found to highlight the following as sub-activities involved in evaluation: comparing (Kim and Ryu 2014; Eckersley 1998); judging, on the basis of value (Eckersley 1998; Kruger and Cross 2006), aesthetics (Chandrasekera et al. 2013), affect (Kim and Ryu 2014), or objective criteria (Lee et al. 2014; Kruger and Cross 2006; Chiu 2003); and decision making with respect to which concept should be taken forward from a range of alternatives (Lee et al. 2014; Kruger and Cross 2006; Chiu 2003).

Cognitive Processes in Conceptual Design

As discussed in the introduction, the systematic review reported herein aimed to address the following question by synthesising the findings of individual protocol studies conducted to date: *What is our current understanding of the cognitive processes involved in conceptual design tasks carried out by individual designers?* Having summarised the characteristics and focus of the reviewed studies, the following sub-sections provide answers to the research question and briefly discuss future work and challenges for the field.

Processes Identified from Protocol Studies

To answer the research question, we firstly identified specific cognitive processes observed and discussed by authors in the focus areas reviewed previously. This revealed considerable differences in the concepts and terminology used to describe cognition. For instance, studies on design as search tend to use the language of problem solving research, describing cognition in terms of operators and state transformations (Akin 1984; Chan 1990; Park and Kim 2007). In constrast, studies on design as exploration frequently describe cognition in terms of perception and the notion of situatedness (Suwa et al. 2000; Park and Kim 2007). To some extent, these differences may be considered to derive from differences in the paradigms underlying each viewpoint. For example, it may be seen from the previous section that studies on design as search largely align with the perspectives of the problem solving paradigm (Table 2), founded in the work of Newell and Simon (1972). In contrast, studies on design as exploration typically align with the perspectives of the reflective paradigm (Table 2), drawing from the work of Schön (1983). Studies on design activities variously reflect perspectives from both paradigms, or neither.

Table 2 Key perspectives associated with the problem solving and reflective paradigms in design cognition research

Paradigms	Perspectives on the nature of: (based on Dorst and Dijkhuis 1995)			
	Designer	Design problem	Designing	Design knowledge
Problem solving	Information processor in an objective reality	Ill-defined, unstructured, stable	A rational search process	Design procedures and scientific laws
Reflective practice	Person constructing their reality	Essentially unique, evolutionary	Reflective interaction with broader design situation	When to apply what procedure or component of knowledge

A major benefit of a systematic review is that it can reveal common findings across a large set of studies. However, owing to the differences in terminology and concepts discussed above, it is difficult to determine what cognitive processes are common across different viewpoints and therefore likely to be fundamentally involved in conceptual design tasks. To gain a clearer view in this respect, we grouped identified processes based on similarities conveyed in definitions and explanations, revealing 10 categories of cognitive function that appear to be studied across the different viewpoints and domains covered by the review: (1) visual perception; (2) mental imagery; (3) semantic association; (4) long term memory; (5) working memory; (6) selective attention; (7) creative thinking; (8) evaluation and decision making; (9) externalisation; and (10) reasoning and problem solving. A useful distinction that is often made in the study and classification of cognitive processes in psychology is that between executive and non-executive functions. Executive functions refer to cognitive processes involved in the selection and monitoring of behaviours to achieve goals, and are accessible to consciousness. In contrast, non-executive functions are typically subconscious, largely automatic processes such as perception and memory retrieval (Rabbitt 2004; Chan et al. 2008). Processes 1–4 above may be classed as non-executive functions, whilst 5–10 constitute executive functions.

Each of the above functions is presented and defined in Table 3, alongside the particular cognitive processes identified through the review. Processes presented in column 2 are the outcome of generalising specific examples identified from the reviewed studies. Illustrative examples in this respect are presented in column 3. Owing to the space limitations of a conference paper, it is not possible to present every specific example identified. Note that the function categories were based on the set of identified processes, but informed by the cognitive psychology literature (as indicated by the citations in column 1).

Table 3 Identified processes and function categories

Function category	Cog. processes	Specific examples
Visual perception, i.e. the process of constructing and consciously sensing internal (visual) representations of the external world (Bruce et al. 2003; Milner and Goodale 2008; Gobet et al. 2011). Perception is driven by afferent sensory information (Eysenck and Keane 2005)	• Perceiving external representations • Analysing and interpreting afferent visual information	Data input operator (Chan 1990) (V1)
		Unexpected discovery of visuo-spatial features and relations (Suwa et al. 2000) (V2)
		Information gathering (Kim et al. 2005, 2006) (V2)
Mental imagery, i.e. the generation, maintenance, and manipulation of internal images, driven by internal information from memory but may be influenced by incoming sensory information (Kosslyn 1995)	• Generation of mental images • Maintenance of mental images • Transformation of mental images	Generation of mental images (Jin and Benami 2010; Park and Kim 2007) (V2)
		Maintenance of images (Jin and Benami 2010; Park and Kim 2007) (V2&3)
		Transformation of images (Jin and Benami 2010; Park and Kim 2007) (V2&3)
Semantic association, i.e. the formation of mental relationships between meaningful representations (Federmeier et al. 2002). Semantic association is intricately related to semantic memory (above), where associations exist (Martin and Chao 2001)	• Association of concepts • Transformation of concepts	Generalisation operator—associate attribute to supra-symbol (Akin 1984) (V1)
		Explore interactions between artefacts and people/nature (Suwa et al. 1998b) (V2)
		Transformation of concepts (Leblebici-Basar and Altarriba 2013) (V3)
Long term memory, supporting long term retention and retrieval of contextualised events (episodic memory) and conceptual knowledge (semantic memory) (Tulving 1983; Squire and Zola 1998)	• Retrieval of information from long term memory	Retrieve schema from memory (Chan 1990; Stauffer and Ullman 1991; Ball et al. 2004) (V1)
		Retrieve knowledge (Suwa et al. 1998b) (V2)
		Memory retrieval (Jin and Benami 2010) (V3)
Working memory, supporting simultaneous storage and manipulation of visuo-spatial and phonological information (Baddeley 1983, 2003)	• Activation, manipulation, and maintenance of information in working memory	Patch operator—add/combine information with making it less abstract (Stauffer and Ullman 1991) (V1)

(continued)

Table 3 (continued)

Function category	Cog. processes	Specific examples
Selective attention, i.e. the process of selecting and focusing on a stimulus while disregarding other stimuli (Gobet et al. 2011; Eysenck and Keane 2005)	• Focusing attention on different parts of external representations • Focusing attention on different properties of external representations	Select information operator (Stauffer and Ullman 1991) (V1) Attend to visual features and spatial relations (Suwa et al. 1998b) (V2)
Creative thinking, i.e. the generative and exploratory processes involved in developing ideas that are both novel and useful and/or valuable (Eysenck and Keane 2005; Finke 1996)	• Idea generation • Concept composition/synthesis • Concept development	Create operator—generation of information that appears spontaneously (Akin 1984) (V1)
		Imagining (Park and Kim 2007) (V2)
		Concept generation and composition (Jin and Chusilp 2006) (V3)
Evaluation and decision making, where decision making is the deliberate selection of one option over another (Gobet et al. 2011), and evaluation is the related process of determining the worth or value of a particular outcome/entity (Gawronski and Bodenhausen 2006)	• Comparing and judging concepts • Decision making about concepts • Evaluation of mental images, design requirements, and design solutions	Reject operator—determine unsatisfactory proposal (Stauffer and Ullman 1991) (V1)
		Evaluation and confirmation of internal representation (Park and Kim 2007), and evaluation of requirements and solutions (Maher and Tang 2003) (V2)
		Comparing and judging (Kim and Ryu 2014) (V3)
Externalisation, referring to the process of externally representing an internal idea or image, e.g. through sketching (Newell and Simon 1972; Fish and Scrivener 1990)	• Drawing/sketching • Depicting	Representation operator—create an external representation (Akin 1984) (V1)
		Drawing (Park and Kim 2007) (V2)
Reasoning and problem solving, where reasoning is the process of thinking in accordance with logic (Gobet et al. 2011), and problem solving is the process of finding solutions to problems (Newell and Simon 1972). A problem is a situation where the end goal is known, but the means of achieving it are not (Newell and Simon 1972; Gobet et al. 2011)	• Inference • Problem structuring, analysis, and redefinition • Solution search • Process control (e.g. monitoring and managing goals, constraints, and requirements)	Search process (Goel 1995; Akin 1984; Chan 1990) and deductive reasoning (Lloyd and Scott 1994) (V1)
		Invention of new design requirements (Suwa et al. 2000) (V2)
		Problem analysis (Jin and Chusilp 2006) (V3)

Key Observations and Future Work

Whilst the variation in concepts and terminology may be attributed to differences in underlying paradigms, it also highlights a lack of general models and theories of conceptual design cognition. This conclusion is supported to some extent by Dinar et al. (2015), who argue that the field suffers from a lack of "cognitive models and theories of designer thinking." Models and theories may be applied to generate predictions about different aspects of cognition, which may then be tested experimentally to further our knowledge of designers' internal processing. Thus, they are crucial for advancing the field (p. 9).

Developing general formalisms to describe the cognitive processing involved in conceptual design requires an understanding of: (i) the cognitive processes fundamentally involved; and (ii) the interactions between the processes. The review reported herein has consolidated knowledge relating to (i), outlining a range of executive and non-executive functions studied across different viewpoints and domains. With respect to (ii), a number of authors in the sample were found to propose tentative relationships and cognitive models on the basis of protocol analysis findings e.g. (Jin and Benami 2010; Suwa et al. 2000; Park and Kim 2007; Kavakli and Gero 2002; Kim et al. 2010). However, a weakness associated with knowledge derived from protocol studies is that it is necessarily based on small samples of designers, and subjective inference from verbal and behavioural data. Subjectivity is particularly problematic in the study of non-executive functions, which are typically not accessible to consciousness for verbal reporting (Rabbitt 2004) and therefore tend to be inferred from observations of physical behaviour alone (e.g. mental imagery processing in Park and Kim 2007). Although the use of standard coding schemes and multiple coders can reduce subjectivity to some extent, these do not address issues relating to sample size, e.g. the statistical significance and generalisability of results. Thus, a significant avenue for future research is testing the findings and hypotheses generated through protocol studies using more objective methods conducive to the study of larger samples. Dinar et al. (2015) highlight the use of controlled lab experiments, typical of cognitive psychology research, as a potential approach in this respect. Systematically reviewing the relatively small number of these studies conducted to date presents another task for future research. A more fundamental challenge for the field as it advances may be how to integrate rich, qualitative approaches such as protocol analysis with more objective and extensive quantitative approaches.

Finally, the variation in concepts and terminology exposed by the review also points to a fundamental question for research on conceptual design cognition: what processes and relationships are actually of interest, and how should they be defined for study? Whilst the function categories applied in this review are reasonable from a psychology perspective, they may not constitute the most appropriate means of defining and organising the processes involved in design. This may be seen to mirror current ontological debates in psychology and neuroscience research, where efforts are under way to develop a shared ontology of processes and relationships

(Poldrack et al. 2011). Several design researchers have proposed ontologies, e.g. Gero (1990), Sim and Duffy (2003), and Gero and Kannengiesser (2004). However, these tend to neglect non-executive functions and are not necessarily intended to describe design at the cognitive level. The development of a general ontology of cognitive processes in conceptual design would not only provide a consistent and comprehensive basis for developing theories and models, but it would also increase the comparability of findings from different studies and promote a more integrated body of knowledge on design cognition.

Concluding Remarks

This paper has reported the findings of the first systematic review of protocol studies specifically focusing on conceptual design cognition. Current knowledge regarding the nature of the cognitive processes involved in conceptual design tasks has been consolidated, revealing ten categories of executive and non-executive function that appear to be studied across the field. The findings highlight several avenues for future research, centering on the need for general formalisms, more objective methods to supplement protocol analysis, and a shared ontology of cognitive processes in conceptual design.

In closing, the work has demonstrated that the systematic review method provides a means to synthesise the findings of a large number of studies in a rigorous and transparent manner, revealing common findings and exposing differences in perspectives and terminology. Thus, further use of the method may help to build a more integrated body of knowledge on design cognition, and therefore make a significant contribution to advancing the cognitive component of design science.

References

*References marked with * are included in the systematic review sample.*

*Akin Ö (1984) An exploration of the design process. In: Cross N (ed) Developments in design methodology. Wiley, Chichester, pp 189–207

Akin Ö (1986) Psychology of architectural design. Pion, London

*Akin O, Akin C (1996) Frames of reference in architectural design: analysing the hyperacclamation (a-h-a-!). Des Stud 17(4):341–361

*Athavankar UA (1997) Mental imagery as a design tool. Cybern Syst 28(1):25–42

*Athavankar U, Bokil P, Guruprasad K (2008) Reaching out in the mind's space. Design computing and cognition 08. Springer, Dordrecht, pp 321–340

Baddeley A (1983) Working memory. Philos Trans R Soc Lond B Biol Sci 302(1110):311–324

Baddeley A (2003) Working memory: looking back and looking forward. Nat Rev Neurosci 4:829–839
*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
*Bilda Z, Gero JS (2007) The impact of working memory limitations on the design process during conceptualization. Des Stud 28(4):343–367
*Bilda Z, Gero JS, Purcell T (2006) To sketch or not to sketch? That is the question. Des Stud 27 (5):587–613
Bruce V, Green PR, Georgeson MA (2003) Visual perception: physiology, psychology and ecology. Psychology Press, New York
*Chan C-S (1990) Cognitive processes in architectural problem solving. Des Stud 11(2):60–80
Chan RCK, Shum D, Toulopoulou T, Chen EYH (2008) Assessment of executive functions: review of instruments and identification of critical issues. Arch Clin Neuropsychol 23(2): 201–216
*Chandrasekera T, Vo N, D'Souza N (2013) The effect of subliminal suggestions on Sudden Moments of Inspiration (SMI) in the design process. Des Stud 34(2):193–215
*Chiu ML (2003) Design moves in situated design with case-based reasoning. Des Stud 24(1): 1–25
Cross N (2001) Design cognition: results from protocol and other empirical studies of design activity. In: Eastman CM, McCracken WM, Newstetter WC (eds) Design knowing and learning: cognition in design education. Elsevier Science Ltd, Oxford
*Daly SR, Yilmaz S, Christian JL, Seifert CM, Gonzalez R (2012) Design heuristics in engineering. J Eng Educ 101(4):601–629
Dinar M, Shah JJ, Cagan J, Leifer L, Linsey J, Smith SM et al (2015) Empirical studies of designer thinking: past, present, and future. J Mech Des 137(2):1–13
Dorst K, Dijkhuis J (1995) Comparing paradigms for describing design activity. Des Stud 16 (2):261–274
*Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem–solution. Des Stud 22(5):425–437
*Eckersley M (1998) The form of design processes: a protocol analysis study. Des Stud 9(2): 86–94
Eysenck MW, Keane MT (2005) Cognitive psychology: a student's handbook, 5th edn. Pyschology Press Ltd., Hove
Federmeier KD, McLennan DB, Ochoa E, Kutas M (2002) The impact of semantic memory organization and sentence context information on spoken language processing by younger and older adults: an ERP study. Psychophysiology 39(2):133–146
Finke RA (1996) Imagery, creativity, and emergent structure. Conscious Cogn 5(3):381–393
Fish J, Scrivener S (1990) Amplifying the mind's eye: sketching and visual cognition. Leonardo 23(1):117–126
Gawronski B, Bodenhausen G (2006) Associative and propositional processes in evaluation: an integrative review of implicit and explicit attitude change. Psychol Bull 132(5):692–731
Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11 (4):26–36
Gero JS, Kannengiesser U (2004) A function–behavior–structure ontology of processes. AI EDAM 21(04):379–391
Gobet F, Chassy P, Bilalic M (2011) Foundations of cognitive psychology. McGraw-Hill Education, Maidenhead
*Goel V (1995) Sketches of thought. MIT Press, Cambridge
*Goldschmidt G (1991) The dialectics of sketching. Creat Res J 4(2):123–143
Hubka V (1982) Principles of engineering design. Butterworth & Co Ltd, London
Hubka V, Eder WE (1996) Design science: introduction to needs, scope and organization of engineering design knowledge. Springer, Berlin
Jiang H, Yen C-C (2009) Protocol analysis in design research: a review. Rigor and relevance in design: IASDR 2009, Seoul, Korea, 18–22 October, pp 147–156

*Jin Y, Benami O (2010) Creative patterns and stimulation in conceptual design. Artif Intell Eng Des Anal Manuf 24(02):191–209
*Jin Y, Chusilp P (2006) Study of mental iteration in different design situations. Des Stud 27 (1):25–55
*Kavakli M, Gero JS (2002) The structure of concurrent cognitive actions: a case study on novice and expert designers. Des Stud 23(1):25–40
Kim KM, Lee KP (2010) Two types of design approaches regarding industrial design and engineering design in product design. In: DESIGN 2010—11th international design conference, pp 1795–1806
*Kim J, Ryu H (2014) A design thinking rationality framework: framing and solving design problems in early concept generation. Hum Comput Interact 29(5–6):516–553
*Kim Y, Jin S, Lee H (2005) Dual protocol analysis based on design information and design process: a case study. Studying designers '05, Aix-en-Provence, France. Key Centre of Design Computing and Cognition, pp 71–85
*Kim YS, Jin ST, Lee SW (2006) Design activities and personal creativity characteristics: a case study of dual protocol analysis using design information and process. In: ASME 2006 international design engineering technical conferences and computers and information in engineering conference, Philadelphia
*Kim JE, Bouchard C, Omhover JF, Aoussat A (2010) Towards a model of how designers mentally categorise design information. CIRP J Manuf Sci Technol 3(3):218–226
Kosslyn SM (1995) Mental imagery. In: Kosslyn SM, Osherson DN (eds) Visual cognition: an invitation to cognitive science, vol 2, pp 267–296
*Kruger C, Cross N (2006) Solution driven versus problem driven design: strategies and outcomes. Des Stud 27(5):527–548
*Lane D, Seery N (2011) Examining the development of sketch thinking and behaviour. Annual conference for the American Society of Engineering Education, Vancouver, Canada
*Leblebici-Basar D, Altarriba J (2013) The role of imagery and emotion in the translation of concepts into product form. Des J 16(3):295–314
*Lee J, Gu N, Williams A (2014) Parametric design strategies for the generation of creative designs. Int J Archit Comput 12(3):263–282
*Liikkanen LA, Perttula M (2009) Exploring problem decomposition in conceptual design among novice designers. Des Stud 30(1):38–59
*Lloyd P, Scott P (1994) Discovering the design problem. Des Stud 15(2):125–140
Lloyd P, Lawson B, Scott P (1995) Can concurrent verbalization reveal design cognition? Des Stud 16(2):237–259
*Maher M, Tang H-H (2003) Co-evolution as a computational and cognitive model of design. Res Eng Des 14(1):47–64
Martin A, Chao LL (2001) Semantic memory and the brain: structure and processes. Curr Opin Neurobiol 11(2):194–201
*McNeill T, Gero JS, Warren J (1998) Understanding conceptual electronic design using protocol analysis. Res Eng Des 10(3):129–140
Milner AD, Goodale MA (2008) Two visual systems re-viewed. Neuropsychologia 46(3):774–785
Moher D, Liberati A, Tetzlaff J, Altman DG (2009) Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. PLoS Med 6(7):1–6
Newell A, Simon HA (1972) Human problem solving. Prentice-Hall, Englewood Cliffs
*Park J, Kim Y (2007) Visual reasoning and design processes. In: International conference on engineering design, ICED'07, pp 1–12
Poldrack RA, Kittur A, Kalar D, Miller E, Seppa C, Gil Y et al (2011) The cognitive atlas: toward a knowledge foundation for cognitive neuroscience. Front Neuroinform 5:1–11
Rabbitt P (2004) Methodology of frontal and executive function. Psychology Press Ltd: online
Roozenburg NFM, Eekels J (1994) Product design: fundamentals and methods. Wiley, Chichester, New York
Schön DA (1983) The reflective practitioner: how professionals think in action. Basic Books, Pennsylvania

Sim SK, Duffy AHB (2003) Towards an ontology of generic engineering design activities. Res Eng Des 14(4):200–223

Squire LR, Zola SM (1998) Episodic memory, semantic memory, and amnesia. Hippocampus 8:205–211

*Stauffer LA, Ullman DG (1991) Fundamental processes of mechanical designers based on empirical data. J Eng Des 2(2):113–125

Suwa M, Tversky B (1996) What architects see in their sketches: implications for design tools. In: Conference companion on human factors in computing systems: common ground, pp 191–192

*Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Des Stud 18(4):385–403

Suwa M, Tversky B, Gero J, Purcell T (1994) Seeing into sketches: regrouping parts encourages new interpretations. Vis Spat Reason Des II:207–219

Suwa M, Gero JS, Purcell T (1998) Analysis of cognitive processes of a designer as the foundation for support tools. Artif Intell Des 98:229–248

*Suwa M, Purcell T, Gero J (1998) Macroscopic analysis of design processes based on a scheme for coding designers' cognitive actions. Des Stud 19(4):455–483

*Suwa M, Gero J, Purcell T (2000) Unexpected discoveries and S-invention of design requirements: important vehicles for a design process. Des Stud 21(2000):539–567

Tulving E (1983) Elements of episodic memory. Clarendon Press, Oxford

*Yu R, Gero JS (2015) An empirical foundation for design patterns in parametric design. In: Ikeda Y, Herr CM, Holzer D, Kaijima S, Kim MJ, Schnabel A (eds) Proceedings of the 20th international conference of the association for computer-aided architectural design research in Asia CAADRIA, pp 1–9

Part III
Design Support

Is Biologically Inspired Design Domain Independent?

Ashok K. Goel, Christian Tuchez, William Hancock and Keith Frazer

Abstract Current theories of biologically inspired design assume that the design processes are domain independent. But is this assumption true? Design Study Library (DSL) is a digital library of eighty-three cases of biologically inspired design collected from a senior-level interdisciplinary class at Georgia Tech over 2006–2013. We describe a preliminary analysis of the DSL case studies. We posit that the assumption about the domain independence is questionable. In particular, some of the parameters in the domains of physiology and sensing appear to be different from the more common domains of mechanics and materials.

Background, Motivation and Goals

The paradigm of biologically inspired design espouses the use of biological systems as analogues for inspiring the design of technological systems as well as standards for evaluating technology designs (Bar-Cohen 2011; Benyus 1997; Bhushan 2009; French 1985; von Gleich et al. 2010; Turner 2007; Vincent and Man 2002; Vogel 2000). Although nature has inspired many a designer in history, including Sushruta, Leonardo da Vinci, and the Wright brothers, over the last generation the paradigm has evolved into a design movement. This transformation is pushed by the perennial desire for design creativity and pulled by the growing need for environmentally sustainable designs. The revolution is manifested through an exponentially expanding literature including both patents (Bonser and Vincent 2007) and publications (Lepora et al. 2013).

However, our understanding of the processes of biologically inspired design remains modest. It is noteworthy that biological phenomena occur at scales ranging from nanometers to megameters, and from nanoseconds to gigaannums. Similarly, biological phenomena occur in a variety of domains ranging from bacteria to archaea to eukaryotes. However, all extant theories of biologically inspired design

A.K. Goel (✉) · C. Tuchez · W. Hancock · K. Frazer
Georgia Institute of Technology, Atlanta, USA
e-mail: goel@cc.gatech.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_9

appear to assume that the design processes are domain- as well as scale-independent (e.g., Goel et al. 2014). Pedagogical techniques for teaching biologically inspired design and computational tools for supporting its practice also make the same assumption. But is this assumption true?

This raises another issue: what is a domain? To be specific, let us consider Weiler and Goel's (2015) description of a mechanical device for harvesting water inspired in part by the design of mitochondria. The issue of scale in this example seems clear; there are two scales of interest: (i) the scale of mitochondria (micrometer) and (ii) the scale of the mechanical device (meter). Thus, apparently there are two scales of interest in biologically inspired design: the scale of the biological source case and the scale of the target design problem.

Similarly we might say that in biologically inspired design, there are two domains of interest: the domain of the source biological phenomenon (mitochondria in this example) and the domain of the target design problem (mechanical devices). To be precise, we adopt a characterization of a domain from the artificial intelligence literature on design (Chandrasekaran 1990; Chandrasekaran et al. 1999; Dym and Brown 2012; Goel 1997): a domain is characterized by the kinds of objects, relations and processes that occur in it. Further, given the context of cross-domain analogical transfer in biologically inspired design (Goel 1997, 2013a; Goel et al. 2014; Shu et al. 2011), we view the real domain of interest to be the "bridging domain" between biology and design. Thus, in the example of the mechanical device, the domain of interest is water harvesting that occurs in biological as well as technological systems (and not mitochondria or mechanical devices).

The Design Study Library (DSL for short) is a digital library of eighty-three case studies of biologically inspired design (Goel et al. 2015b). The case studies were collected over 2006–2013 from extended collaborative projects in a senior-level interdisciplinary class at Georgia Institute of Technology. These case studies provide an empirical basis for examining the domain independence of biologically inspired design. In this paper, we describe a preliminary analysis of the eighty-three case studies. We posit that the assumption about the domain-independence of biologically inspired design is questionable.

Biologically Inspired Design

The growth of biologically inspired design movement has lead to a proliferation of information-processing theories, pedagogical techniques, and computational tools supporting its practice.

Information-Processing Theories

Some information-processing theories of biologically inspired design are descriptive: Design Spiral (Baumeister et al. 2012), for example, derives from observations of biologically inspired design in practice; (Shu et al. 2011) provide an alternative descriptive account. Some theories are normative: BioTRIZ (Vincent et al. 2006), for example, applies the well-known TRIZ design methodology (Altshuller 1984) to biologically inspired design; (Nagel and Stone 2010) provide an alternative method. Some theories are explanatory: Chakrabarti's and his colleagues' GEMS model (2010) and Goel's (2013a) Task Model seek to provide explanations of observed biologically inspired design practices. *All* these descriptive, normative and explanatory theories of biologically inspired design are domain-independent as well as scale-independent.

Pedagogical Techniques

Several educational programs offer opportunities for learning about biologically inspired design. For example, Arizona State University offers a variety of courses on biomimicry for professional and student designers (http://biomimicry.asu.edu/), and Georgia Tech offers a sequence of undergraduate courses that leads to a certificate in biologically inspired design (http://www.cbid.gatech.edu/). Arizona State University's courses generally use the Design Spiral (Baumeister et al. 2012) as the design methodology. Goel's (2013a) Task Model both derives from cognitive analyses of design practices in the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course on biologically inspired design, and has influenced the teaching in the class (Yen et al. 2011). *All* these pedagogical techniques are domain-independent as well as scale- independent.

Computational Tools

Many computational tools are available for supporting biologically inspired design. The Biomimicry Institute's AskNature provides access to a functionally indexed digital library of textual and visual descriptions of biological systems (http://www.asknature.org/; Deldin and Shuknecht 2014). IDEA-INSPIRE (Chakrabarti et al. 2005) and DANE (http://dilab.cc.gatech.edu/dane/) provide access to functionally indexed digital libraries of multimodal structured representations of biological and technological systems. Vincent et al. (2006) are developing BioTRIZ, a biomimetic version of the famous TRIZ system for supporting engineering design (Altshuller 1984). Nagle (2013) has developed a thesaurus for functions in biologically inspired design. Watson+ (Goel et al. 2015b) builds on IBM's Watson cognitive system and acts as a research assistant for biologically inspired design. All these computational tools are domain-independent as well as scale-independent.

The Design Study Library

The Design Study Library (DSL) is a web-based, interactive, digital library of eighty three case studies of biologically inspired design (Goel et al. 2015b). Each case study in DSL consists of one or more documents describing a design project, and is indexed by Function, Structure, Domain Principle and Operating Environment. DSL supports multiple methods for users to access these documents.

All eighty-three case studies in DSL come from open-ended extended collaborative design projects from 2006 through 2013 in the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 class. This is a yearly, interdisciplinary, project-based class taken mostly by senior level students. During 2006–2013, the class was co-taught jointly by biology, engineering, and design faculty led by Professor Jeannette Yen. During these years, the classes were composed of students from a variety of other science and engineering disciplines. The precise composition of the class varied from year to year, but in general the class consisted of a majority of engineers.

In the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 class, students work in teams of 4–5 on extended, open-ended, self-selected design projects. Instructors ensure that each team has at least one student majoring in biology and a few from different engineering and design disciplines. Each team develops a conceptual design that can address a technical problem based on one or more biological analogues. Each team has one or more faculty mentors. Yen et al. (2011) discuss the challenges in teaching the class; Yen et al. (2014) trace the evolution of the class from 2006 through 2012.

Prior Analysis of DSL

Prior analysis of the case studies in DSL pertained to the relationship between biologically inspired design and environmental sustainability. Goel et al. (2015b) found that environmental sustainability was an explicit goal of about one fourth of the case studies. They also found that in some case studies, although sustainability was not a design goal, the designers' analyses indicated that the design would be more sustainable than conventional designs. They found this kind of serendipitous sustainability in about 8% of the case studies. Taking serendipitous sustainability into account, sustainability was a factor in about a third of the case studies.

Categorization of the Case Studies

Our analysis makes use of a dozen categories for classifying the case studies in DSL in addition to Function, Structure, Principle, and Operating Environment that apply to all eighty three case studies. First, as noted above, the DSL case studies were classified and labeled as “intentionally sustainable” or “serendipitously sustainable”. Second, cases that contained “environmental impact analysis” were tagged as such. Third, five labels were obtained from Goel’s (2013a) Task Model of biologically inspired design: problem decomposition, compound analogy, problem reformulation, problem-driven design, and solution-based design.

Finally, four labels for classifying domains were obtained from Professor Yen, a Georgia Tech Professor of Biology and the primary instructor of the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course: physiology, mechanics, materials, and sensing. While the domains of mechanics, materials, and sensing are straightforward, the real domain of interest in case of “physiology” is the abstract system that can be instantiated in both physiology and technology. Table 1 provides brief characterizations of the twelve categories including the four domains.

A preliminary analysis revealed that nine case studies in DSL were too short or vague to be tagged with consistency, and thus were deleted from further analysis. The

Table 1 Description of the semantic labels on the case studies

Semantic label	Description
Problem decomposition	The case study contained a functional decomposition of the problem
Compound analogy	The resulting design contains elements from two or more biological analogues
Problem reformulation	The case study specifically mentioned that the problem changed due to some reason
Problem-driven design	The case study started with a problem and a solution for the problem was generated
Solution-based design	The case study started with a design pattern from biology and a problem was found that could be addressed with the pattern
Environmental impact analysis	The case study contained such an analysis
Mechanics	Some form of movement was critical to the proposed design
Materials	The proposed solution emphasized a particular material that was more beneficial than another
Sensing	The design contained some form of sensing mechanism derived from biology
Physiology	The solution used as inspiration a pattern (mechanism, principle, structure, form) related to the internal functioning of an organism
Intentional Sustainability	The primary goal of the case study related to sustainability
Serendipitous sustainability	The case study did not mention sustainability, but the solution was sustainable

remaining seventy four case studies were categorized independently by two of the coauthors (Tuchez and Hancock). Both are computer scientists familiar with biologically inspired design. The two coders initially labeled the case studies independently, then negotiated about the precise characterizations of the categories, and next relabeled the case studies independently. A case study may have multiple labels.

Table 2(a) shows the legend used in Tables 2(b) and (c); the latter two tables show the association matrices for the two coders. Thus, the first row in Table 2(b) says that 34 case studies (out of the total 74) were labeled as problem decomposition, 14 had the labels problem decomposition and compound analogy, and so on.

Cohen's kappa coefficient was used to measure the degree of agreement between two coders: the kappa score was 0.88, corresponding to a 94% agreement between the coders, which is commonly considered to be very accurate.

A preliminary analysis reveals six patterns common to the association matrices in Tables 2(b) and (c):

(P1): *Compound analogy is rare with solution-based design.* A possible explanation for this pattern is that in the solution-based approach, designing typically starts with a design principle in a single biological system and then a problem that can be solved using the principle is identified. This leaves little room for compound analogy as it requires drawing inspiration from more than one biological analogue. A corollary of this hypothesis is that solution-based design may lead to fixation on a single analogue.

(P2): *Problem decomposition is likely to be found when problem-driven design too is found.* An explanation for this pattern appears to directly follow from the characterizations of problem-driven design and problem decomposition.

(P3) *Solution-based design is commonly found in physiology, and not as much in other domains.* This pattern initially was a surprise to us; insofar as we know, it has not been previously discussed in the literature. However, Coder 1 found that 8 out of 9 case studies that used solution-based design were in physiology; Coder 2 found the same for 7 out of 12 case studies. One possible explanation is that the domain of "physiology" affords system-level design principles that trigger solution-based design more commonly than the other three domains of mechanics, materials and sensing. This is consistent with our characterization of real domain of interest here, namely, the mechanism of internal functioning of a system, and thus at least partially validates our characterization of the domain.

(P4): *Sensing commonly uses problem-driven design, not solution-based design.* Again insofar as we know, this pattern has not been previously discussed in the literature. However, Coder 1 found that 8 out of 9 case studies in sensing used problem-driven design; Coder 2 found the same for 7 out of 10 case studies. It appears that biologically inspired design in sensing mostly begins with a problem and not a solution, perhaps because the domain presents relatively well-defined problems.

(P5) *Materials and sensing rarely occur together.* We do not presently have a good explanation for this hypothesis.

(P6) *Environmental impact analysis is seldom done with solution-based design.* Again, we do not presently have a good explanation for this hypothesis.

Table 2 (a) Legend for (b) and (c), (b) Association matrix for coder 1, (c) Association matrix for coder 2

(*a*)

PD	Problem decomposition	ME	Mechanics
CA	Compound analogy	MA	Materials
PR	Problem reformulation	SE	Sensing
PB	Problem-based design	PH	Physiology
SB	Solution-based design	IS	Intentional sustainability
EI	Env. impact analysis	SS	Serendipitous sustainability

(*b*)

	PD	CA	PR	PB	SB	EI	ME	MA	SE	PH	IS	SS
PD	34	14	2	32	2	20	20	19	5	16	16	3
CA	14	31	3	31	0	19	13	20	4	15	13	2
PR	2	3	3	3	0	1	1	2	0	3	1	0
PB	32	31	3	65	0	33	35	37	8	33	25	5
SB	2	0	0	0	9	1	6	3	1	8	1	0
EI	20	19	1	33	1	34	17	22	3	16	19	3
ME	20	13	1	35	6	17	41	18	3	22	12	3
MA	19	20	2	37	3	22	18	40	0	21	14	4
SE	5	4	0	8	1	3	3	0	9	2	2	0
PH	16	15	3	33	8	16	22	21	2	41	17	1
IS	16	13	1	25	1	19	12	14	2	17	26	0
SS	3	2	0	5	0	3	3	4	0	1	0	5
Total	34	31	3	65	9	34	41	40	9	41	26	5

(*c*)

	PD	CA	PR	PB	SB	EI	ME	MA	SE	PH	IS	SS
PD	34	14	2	29	5	20	16	21	6	15	16	3
CA	14	31	4	31	0	19	13	22	4	14	11	2
PR	2	4	4	4	0	2	1	3	0	3	1	0
PB	29	31	4	62	0	31	28	42	7	30	23	5
SB	5	0	0	0	12	2	10	4	3	7	1	0
EI	20	19	2	31	2	33	15	22	3	13	18	3
ME	16	13	1	28	10	15	38	20	4	18	8	2
MA	21	22	3	42	4	22	20	46	1	25	14	5
SE	6	4	0	7	3	3	4	1	10	3	2	0
PH	15	14	3	30	7	13	18	25	3	37	14	0
IS	16	11	1	23	1	18	8	14	2	14	24	0
SS	3	2	0	5	0	3	2	5	0	0	0	5
Total	34	31	4	62	12	33	38	46	10	37	24	5

Word Cloud Analysis

We generated word cloud images to visualize patterns in the documents of each of the seventy-four case studies. We then aggregated the word clouds for each of the four domains: physiology, mechanics, materials, sensing. As Fig. 1 illustrates, sensing shows a higher relative frequency of the verb "detect" as well as "need" compared to the other three domains.

In addition, we compared the frequencies of six nouns for the four domains: system, function, structure, behavior, mechanism, and environment based on the Structure-Behavior-Function modeling (Goel 2013b). Figure 2 illustrates the normalized frequency of these words for the four domains. Note that case studies in the domains of physiology and sensing have a higher occurrence of noun "system". Further, sensing has a higher frequency of "environment" and a lower frequency of "structure" compared to the other three domains.

Fig. 1 Verb clouds for the four domains

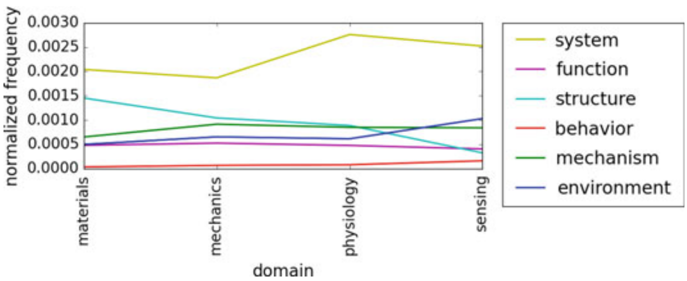


Fig. 2 Frequency of selected words in the four domains

Statistical Analysis

We measured associations between the labels of Tables 2(b) and (c) using Fisher's exact test and the Pearson correlation coefficient. Fisher's exact test is a well known statistical significance test for analyzing association tables such as Tables 2(b) and (c). Fisher's test is appropriate for this study because the categorical nature of data in the two tables. Pearson correlation coefficient is a standard measure of the linear correlation between two variables X and Y, giving a value between +1 and −1, where 1 is total positive correlation and −1 is a total negative correlation.

As Tables 3(a) and (b) indicate, the two-tailed test with $p < 0.05$ does not confirm patterns P2 and P4. P2 refers to problem decomposition in problem-driven design: Problem Decomposition is likely to be found when problem-based design too is found. In retrospect, the reason for the failure to confirm this pattern is clear: while the numbers in the relevant cells in Tables 2(b) and (c) are fairly large (32 for coder 1 and 29 for coder 2), the proportions are relatively small compared to the number of the case studies with problem-driven design (65).

Pattern P4 pertains to the domain of sensing: (P4) Sensing commonly uses problem-based design, not solution-based design. We expect this is (only) because of the small sample size (10) of the case studies pertaining to the sensing domain. We note that the word cloud analysis provides additional evidence that the domain of sensing is different from the other three domains. Thus, this hypothesis requires additional investigation.

Discussion

In this section, we critique this work from the perspectives of (i) research methodology, (ii) design theory, and (iii) limitations of the current study. We also discuss some directions for future work.

First, from the perspective of research methodology, while biologically inspired design is a well-known paradigm, systematic research of the design paradigm is relatively new. Much research on biologically inspired design appears to be based on informal retrospective analysis of a small number of skeletal and anecdotal case studies. There is a need for more rigorous analysis of larger samples of case studies of biologically inspired design.

Vattam et al. (2007) analyzed seventy seven case studies of biologically inspired design. While sixty of the case studies were reported in the design literature, seventeen were taken from the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course (and are now a part of the DSL digital library). The previous research resulted in several findings, for example, (a) biologically inspired design is characterized by two core design methods, namely, problem-driven design and solution-based design, and (b) compared to problem-driven design, solution-based design often results in multifunctional design but runs the risk of fixation on the

Table 3 (a) Coder 1's significant correlations, (b) Coder 2's significant correlations

	Tag A	Tag B	Fisher *P* value	Pearsons correlation
(*a*)				
1	PB	CA	0.008	0.316
2	SB	CA	0.008	−0.316
3	SB	PB	0.000	−1.000
4	EI	CA	0.034	0.261
5	EI	PB	0.033	0.260
6	EI	SB	0.033	−0.260
7	SE	MA	0.000	−0.404
8	PH	PB	0.037	−0.251
9	PH	SB	0.037	0.251
10	IS	EI	0.001	0.401
	Tag A	Tag B	Fisher *P* value	Pearsons correlation
(*b*)				
1	PR	CA	0.027	0.282
2	PB	CA	0.001	0.374
3	SB	CA	0.001	−0.374
4	SB	PB	0.000	−1.000
5	EI	PD	0.035	0.264
6	EI	CA	0.019	0.285
7	ME	PB	0.025	−0.282
8	ME	SB	0.025	0.282
9	MA	PB	0.047	0.262
10	MA	SB	0.047	−0.262
11	SE	MA	0.000	−0.425
12	IS	PD	0.024	0.288
13	IS	EI	0.000	0.424
14	IS	ME	0.047	−0.250

design structure. In contrast, this paper explores the role of domain in the design processes. Note that pattern P1 in the current study (Compound analogy is rare with solution-based design) appears to confirm the second half of the (b) pattern from the previous study.

A related methodological point is about the importance of digital libraries of case studies of biologically inspired design such as DSL. As Goel et al. (2015b) note, digital libraries such as DSL enable systematic documentation and analysis; this research would be much harder to conduct without DSL.

Second, from the viewpoint of design theory, there long has been a debate about the domain independence of design tasks and methods (e.g., Chandrasekaran 1990; Cross 2006; Dym and Brown 2012; Eastman et al. 2001; French 1994; Goel 1997; Simon 1996). On one hand, design disciplines such as architecture, engineering and computing have developed many domain-specific design theories. Within computing, the design domains of computational architecture, software, and interface have developed their own domain-specific design theories. Yet, there also appears to be a degree of generality to many design tasks and methods across various domains. Indeed, the search for this design generality is one of the motivations for the conference series on Design Computing and Cognition.

Kannengiesser and Gero (2015) have argued that their Function-Behavior-Structure framework for design captures the generality of design processes across the domains of engineering, software and service design. However, Vermaas (2013) has enumerated several meanings of “function” within engineering itself, and Goel (2013b) has described the evolution of the meaning of “function” within the Structure-Behavior-Function theory of system modeling: as the scope of SBF modeling evolved from problem solving to memory to learning, so did its characterization of “function”. Nevertheless, it is interesting to search for levels of abstraction for capturing the generality of a design paradigm. The organizing principle of using analogies to nature for inspiring the design of technological systems and evaluating technological designs captures the unity of biologically inspired design across domains and scales.

While it is interesting to search for a level of abstraction for capturing the unity of many design processes, it is also important to search in the opposite direction of domain-specificity of many design methods. We posit that current assumptions about the domain-independence of biologically inspired design processes may have obscured important differences between domains. Given the importance of mechanics and materials in engineering, the focus of much research on biologically inspired design has been on biomechanics and biomaterials. For example, the Georgia Tech undergraduate certificate in biologically inspired design comprises of a sequence of courses starting with ME/ISyE/MSE/PTFe/BIOL 4740 and continuing with courses on biomechanics and biomaterials. However, as we conduct systematic analysis of case studies of biologically inspired design from different domains, we are beginning to find domain-specific parameters.

Third, while this study deals with a fairly large sample size, it has a few limitations. One limitation pertains to possible sample bias: as noted earlier, all case studies in DSL come from extended collaborative design projects from 2006 through 2013 in the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 class. While it is true that the students in this class are novice designers, it is also true that engineers in general are not necessarily experts at biology and biologists in general are not necessarily sophisticated at design. Thus, it is not clear how to characterize expertise in biologically inspired design or exactly who is an expert in it, and the results of this study might be more general than appears at first glance. Nevertheless, it is very important to replicate this preliminary study with larger samples of biologically inspired design case studies acquired from different groups of subjects such as the professional and student designers participating in The Biomimicry Institute's Design Challenges (https://biomimicry.org/design-challenges/).

Another limitation is that while the two coders in this study are familiar with biologically inspired design, neither has much formal background in biology. It might be useful to replicate this study with a different set of coders with stronger backgrounds in biology.

Yet another limitation pertains to the classification of domains. As noted earlier, we obtained the four labels classifying domains from Professor Yen, a Georgia Tech Professor of Biology and the primary instructor of the Georgia Tech ME/ISyE/MSE/PTFe/BIOL 4740 course. The rationale behind this classification apparently is that leading biology journals, such as the *Journal of Experimental Biology* and the *Journal of Bioinspiration and Biomimetics,* use it. However, as we noted earlier, "physiology" here appears to be a biological instantiation of the real domain of interest, the "bridging domain" of the mechanism of internal functioning of a system. A revised or refined classification of bridging domains may reveal additional domain-specific differences.

These limitations notwithstanding, we submit that this study raises an important question about a basic assumption of all current theories of biologically inspired design, namely that the design processes are domain independent. Thus, this study represents a necessary first step: now that it has proposed a novel hypothesis, it can be replicated and tested, and revised and refined through additional studies.

Finally, while our analysis thus far has primarily addressed the question of domain independence of the processes of biologically inspired design, it also pertains to the issue of scale independence of the design processes. As we noted in the introduction, the example of a mechanical device for water harvesting inspired by the design of mitochondria has two scales of interest: the micrometer scale of mitochondria and the meter scale of the mechanical device. However, we also said that the real domain of interest here is that of water harvesting (and neither mitochondria nor mechanical devices per se). We note that the design pattern for water harvesting in the Weiler and Goel (2015) example evidently is scale-invariant (or analogical transfer from mitochondria to the mechanical device would not be feasible). Thus, we posit that biologically inspired design processes likely are scale-independent. This counter intuitive hypothesis calls for analysis and testing in future work.

Conclusions

Current theories of biologically inspired design assume that the design processes are domain-independent as well as scale-independent. Current pedagogical techniques and computational tools for supporting biologically inspired design too make the same assumption of domain- and scale-independence. In this paper, we examined the assumption of domain independence by analyzing eighty three cases of biologically inspired design collected from a senior-level interdisciplinary class at Georgia Institute of Technology over 2006–2013 and organized in a digital library called DSL. We discovered that some of the parameters in the domains of physiology and sensing appear to be different from the more common domains of mechanics and materials. In particular, we discovered that solution-based design is commonly found in the domain of "physiology" (actually, the domain of internal functioning of systems) and not as much in other domains. While our study did not directly validate the additional finding that sensing commonly uses problem-driven design and not solution-based design, there is strong evidence in favor of this pattern as well.

Of course it is important to replicate these preliminary studies with larger samples of biologically inspired design case studies acquired from different groups of subjects and using a refined classification of domains. If these hypotheses about the differences between the parameters of the various domains hold, then they likely will have important implications not only for building new, more detailed information-processing theories of biologically inspired design, but also for developing pedagogical techniques for teaching the design paradigm as well as computational tools for supporting its practice.

Acknowledgements We are grateful to the developers of the Design Study Library, including Gongbo Zhang, Bryan Wiltgen, Swaroop Vattam, and Yuqi Zhang. We are especially grateful to Professor Jeannette Yen, the primary instructor of the Georgia Tech ME/ISyE/MSE/BME/BIOL 4740 class from 2006 through 2013.

References

Altshuller G (1984) Creativity as an exact science: the theory of the solution of inventive problems. translated by Anthony Williams. Gordon & Breach, Amsterdam

Bar-Cohen Y (ed) (2011) Biomimetics: nature-based innovation. CRC Press, Boca Raton

Baumeister D, Tocke R, Dwyer J, Ritter S, Benyus J (2012) Biomimicry resource handbook. Biomimicry 3.8, Missoula

Benyus J (1997) Biomimicry: innovation inspired by nature. William Morrow, New York City

Bhushan B (2009) Biomimetics: lessons from nature—an overview. Philos Trans R Soc A Math Phys Eng Sci 367(1893):1445–1486

Bonser R, Vincent J (2007) Technology trajectories, innovation, and the growth of biomimetics. Proc Mech Eng Part C J Mech Eng Sci 221(10):1177–1180

Chakrabarti A, Sarkar P, Leelavathamma B, Nataraju B (2005) A functional representation for aiding biomimetic and artificial inspiration of new ideas. AIEDAM 19:113–132

Chandrasekaran B (1990) Design problem solving: a task analysis. AI Mag 11(4):59–71
Chandrasekaran B, Josephson J, Benjamins V (1999) What are ontologies and why do we need them? IEEE Intell Syst 14(1):20–26
Cross N (2006) Designerly ways of knowing. Springer, Berlin
Deldin JM, Shuknecht M (2014) The AskNature database: enabling solutions in biomimetic design. In: Goel A, McAdams D, Stone R (eds) Biologically inspired design: computational methods and tools. Springer, Berlin
Dym C, Brown DC (2012) Engineering design: representation and reasoning. Cambridge University Press, New York
Eastman C, Newstetter W, McCracken M (eds) (2001) Design knowing and learning: design cognition in education. Elsevier, Amsterdam
French M (1985) Conceptual design for engineers, 2nd edn. Springer, Berlin
French M (1994) Invention and evolution: design in nature and engineering, 2nd edn. Cambridge University Press, Cambridge
Gebeshuber IC, Gruber P, Drack M (2009) A gaze into the crystal ball: biomimetics in the year 2059. Proc Inst Mech Eng Part C J Mech Eng Sci 223(12):2899–2918
Goel A (1997) Design, analogy and creativity. IEEE Intell Syst 12(3):62–70
Goel A (2013a) Biologically inspired design: a new program for computational sustainability. IEEE Intell Syst 28(3):80–84
Goel A (2013b) A 30-year case study and 15 principles: implications of an artificial intelligence methodology for functional modeling. AI EDAM 27(3):203–215
Goel A, McAdams D, Stone R (eds) (2014) Biologically inspired design: computational methods and tools. Springer, London
Goel A, Creeden B, Kumble M, Salunke S, Shetty A, Wiltgen B (2015a) Using watson for enhancing human-computer co-creativity. In: Proceedings of AAAI 2015 fall symposium on cognitive assistance, Arlington, VA, November
Goel A, Zhang G, Wiltgen B, Zhang Y, Vattam S, Yen J (2015b) On the benefits of digital libraries of case studies of analogical design: documentation, access, analysis and learning. AIEDAM 29(2):215–227
Kannengiesser U, Gero J (2015) Is designing independent of domain? Comparing models of engineering, software and service design. Res Eng Des 26(3):253–275
Lepora N, Verschure P, Prescott T (2013) The state of the art in biomimetics. Bioinspir Biomim 8(1)
Nagel J (2013) A Thesarus for biologically inspired design. In: Goel, McAdams & Stone (eds) Biologically inspired design: computational methods and tools
Nagel J, Stone R (2010) A computational approach to biologically inspired design. AIEDAM 26(2):161–176
Shu L, Ueda K, Chiu I, Cheong H (2011) Biologically inspired design. Keynote Paper. CIRP Ann Manuf Technol 60(2)
Simon H (1996) Sciences of the artificial, 3rd edn. MIT Press, Cambridge
Srinivasan V, Chakrabarti A (2010). An integrated model of designing. ASME J Comput Inf JCISE 10(3)
Turner J (2007) The Tinkerer's accomplice: how design emerges from life itself. Harvard University Press, Massachusetts
Turner J, Soar R (2008) Beyond biomimicry: what termites can tell us about realizing the living building. In: Proceedings of 1st international conference on industrialized intelligent construction, pp 221–237
Vattam S, Helms M, Goel A (2007) Biologically inspired innovation in engineering design: a cognitive study. Technical Report, GIT-GVU-07-07
Vermass P (2013) The coexistence of engineering meanings of function: four responses and their methodological implications. AIEDAM 27:191–201
Vincent J, Man D (2002) Systematic technology transfer from biology to engineering. Philos Trans R Soc A Math Phys Eng Sci 360(1791):159–173
Vincent J, Bogatyreva O, Bogatyrev N, Bowyer A, Pahl A (2006) Biomimetics: its practice and theory. J R Soc Interface 3:471–482

Vogel S (2000) Cat's paws and catapults: mechanical worlds of nature and people. W.W. Norton and Company, New York City

von Gleich A, Pade C, Petschow U, Pissarskoi E (2010) Potentials and trends in biomimetics. Springer, Berlin

Weiler C, Goel A (2015) From mitochondria to water harvesting: a case study in biologically inspired design. IEEE Potentials 34(2):38–43

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 & Francis, Boca Raton

Yen J, Helms M, Goel A, Tovey C, Weissburg M (2014) Adaptive evolution of teaching practices in biologically inspired design. In: Goel A, McAdams D, Stone R (eds) Biologically inspired design: computational methods and tools. Springer, London

A Meta-Analytic Approach for Uncovering Neural Activation Patterns of Sustainable Product Preference Decisions

Kosa Goucher-Lambert, Jarrod Moss and Jonathan Cagan

Abstract This paper explores the use of neuroimaging data to inform results from a preference decision study involving product sustainability. Neurosynth, a meta-analytic database of functional magnetic resonance imaging (fMRI) studies, was used to extract regions of interest (ROIs) for cross comparison with an empirically collected fMRI dataset. The tasks for the empirically collected fMRI dataset were product preference decisions involving sustainability. In particular, participants were engaged in preference judgments separated into two conditions; one with and one without calculated environmental impact values displayed alongside each design alternative. Extracted meta-analytic ROIs were generated based upon keywords (moral, emotion, etc.) from hypotheses on the ways individuals formulate opinions regarding the environment. Furthermore, additional keywords were seeded based on the results of a whole-brain fMRI analysis. Results indicate the important role of moral reasoning and theory of mind processing in product evaluations within social choice domains, such as sustainability.

Introduction

What are people thinking while they evaluate products when sustainability is a factor? Goucher-Lambert and Cagan (2015) showed that product evaluations based in part on sustainability become social choice problems, where product attributes are evaluated differently. If this is the case, what mental processes are involved in such social choice evaluations? In this paper, neuroimaging data is analyzed using separate broad activation networks, each defined by specific theme-based keywords derived from a meta-analytic database. By identifying related brain activation networks, unique features of social choice preference judgments can be uncovered.

K. Goucher-Lambert · J. Cagan (✉)
Carnegie Mellon University, Pittsburgh, USA
e-mail: cagan@cmu.edu

J. Moss
Mississippi State University, Mississippi, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_10

Understanding user behavior and decision-making is a critical aspect of the design process. However, common qualitative and quantitative approaches are limited in that that they often require response information from human subject participants. Neuroimaging techniques offer a powerful set of methods to understand open design research questions, as they can provide insight into information that human participants are unaware, unable, or unwilling to provide. One such technique, functional magnetic resonance imagining (fMRI), is already being expanded into distant domains outside of cognitive psychology and neuroscience, in order to learn more about the underlying brain activation supporting specific functions. In this project, fMRI was used to investigate features of preference judgments involving sustainable products.

The current work is part of a multi-year project (Fig. 1), which aims to connect behavioral and neuroimaging data to inform design research on features of sustainable preference judgments. Minimizing the impact on the environment is a critical issue facing the design research community; however, little is understood regarding the way people interact with and decide between various product options when sustainability is being considered (Peattie and Charter 2003; Peattie and Belz 2010). Behavioral data such as preference modeling techniques provide insight into participants' stated decisions. However, behavioral data do not always tell the full story; at times people do not say what they feel, or their logic stifles their internal preferences. When used in compliment with behavioral data, neuroimaging techniques provide much deeper insight into participants' mental activity during preference decisions—insight into what participants do not or cannot say. Within this project, fMRI data has been explored using a variety of analysis techniques. In this paper a meta-analytic approach is presented.

The meta-analytic approach used in this paper serves as a standalone analysis of the empirically collected neuroimaging data, through the creation of specific regions of interest (ROIs). fMRI data can be examined using whole-brain analysis,

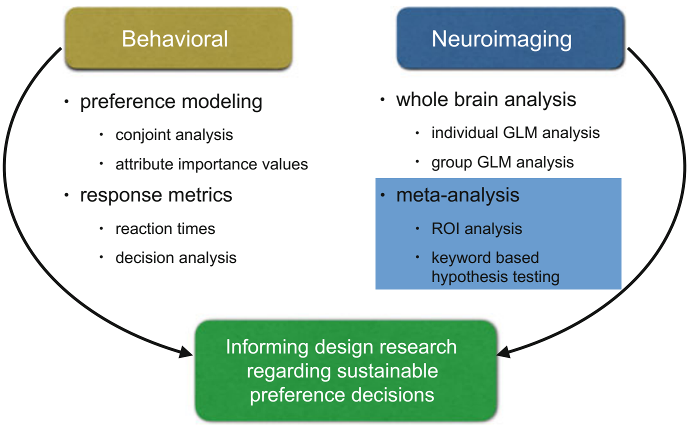


Fig. 1 The entire project contains behavioral and neuroimaging components. The work in this paper represents the meta-analysis contribution under neuroimaging

where statistical tests are performed for every voxel (volume element) in the brain (which was captured during data acquisition), or over a subset of the brain defined by ROIs. ROIs are used to analyze fMRI data for a variety of reasons. For example, ROIs can be useful for data exploration, as they limit statistical testing and analyses to specific regions of the brain. Additionally, ROIs can be created for areas that are functionally pre-defined based on results and known associations from existing literature (Poldrack 2007). In this work, ROIs were extracted for both purposes using an open-sourced meta-analysis tool called Neurosynth (Yarkoni et al. 2011). Using Neurosynth, activation maps are generated based on meta-analytic results, which represent brain activation from specific keywords. This allows for the analysis of neuroimaging data to be completed over a reduced area, and the cross validation of the empirical data using external datasets.

Five keywords were selected for ROI extraction; moral, theory of mind, empathy, emotion, and vision. These keywords were included in this work based largely on a previous analysis of behavioral data, as well as an initial analysis of fMRI data completed separately (Goucher-Lambert et al. 2016). It was hypothesized separately that the neuroimaging results would show brain activation in emotion related regions, due to the fact that the environment was thought to be evoking emotional, and even empathetic responses. Contrary to this, initial results from a group analysis of the neuroimaging data indicated that there was not a significant difference in activation for regions associated with emotion and empathetic decision-making when participants were considering a product's environmental impact (Goucher-Lambert et al. 2016). Instead, areas of activation were found related to theory of mind (i.e., what will others in society think of my actions) and decreased activation in visual processing related areas while considering the environmental impact of a product (Goucher-Lambert et al. 2016).

In this work, behavioral data was collected using a visual conjoint preference survey inside of an fMRI scanner. Study participants were asked to select their most preferred option between various reusable water bottles comprised of form, function, and price attributes. In the "Environmental" condition of the experiment, calculated environmental impact values were presented along with each design alternative; in the "Control" condition this information was absent, and replaced with an additional decision variable, "Poisson's Ratio". The Control condition added an extra attribute to keep the number of decision variables consistent between conditions, but did not contribute meaningful information to the eventual decision. The goal of this experiment was to uncover unique features of sustainable preference judgments. Brain imaging data was analyzed on an individual level using regression statistics. The results presented in this paper focus on the results from a ROI analysis based on activation keywords from a meta-analytic database. The meta-analytic approach allows for further insight regarding sustainable preference judgments that would be extremely difficult to obtain through traditional statistical analyses of the data. It is hypothesized that the meta-analytic approach will support both evidence based results (i.e., increase in theory of mind activation and decrease in visual processing for environmental impact trials), as well as the original hypothesized results (i.e., emotion, empathy, and moral activation present in environmental impact decisions), while participants consider a product's environmental impact.

Background

The work presented in this paper utilizes a multi-disciplinary approach and draws on work from design research on preference modeling and sustainability, as well as work from the cognitive psychology and neuroscience communities pertaining to the meta-analysis of fMRI data. These areas are briefly discussed below.

Sustainability in Design Research Literature

As achieving a more sustainable society rises to the forefront of concern, trying to find ways to minimize the impact of products on the environment is becoming increasingly important. Over the past decade, the design research community has put significant work into developing methods to reduce the environmental impact of products and services during the design stage (for a few examples see Bernstein et al. 2010; Devanathan et al. 2010; Fitzgerald et al. 2010; Masui et al. 2003). The rationale behind this focus is that the majority of the decisions that influence the sustainability of a product occur while a product is being designed (Ramani et al. 2010).

Another portion of the design research community has looked to address sustainability by focusing on consumer preference and the interpretation of sustainable features by consumers. It is well documented that stated demand for sustainable products is high, while the success of these same products has remained relatively low (Blamey et al. 1999). It is believed that a better understanding of how consumers evaluate sustainable products, as well as products under the context of sustainability, will help lead to the creation of valuable strategies for designers to satisfy these requirements (for an example of such strategies see Macdonald 2012). Results from human participant experiments have provided insights into preference judgments involving sustainability. However, as is common in many preference-based research studies, participants are often subject to bias their responses, or are unable to completely express the rationale behind their decisions (Spector 2004; Nederhof 1985; Fleming et al. 2010). This problem is particularly pervasive in sustainable preference judgments, where social desirability bias is thought to be in play.

She and Macdonald (2013) explored the use of sustainable trigger features on products, finding that adding such features was more likely to increase consumer preference for environmentally smart products. Reid et al. (2010) explored the perceived environmental friendliness of various vehicles based solely on the form of the cars. They discovered that simplified form factors similar to that of existing environmentally friendly vehicles (i.e. Toyota Prius) were associated with low perceived environmental impacts. In another study, Macdonald et al. (2009) examined consumer preference within a sustainability context for paper towel configurations.

Goucher-Lambert and Cagan (2015) used a visual conjoint analysis approach to study how consumer preference was affected by environmental impact information. Visual conjoint analysis allows for the decomposition of an object's form into continuous attributes, which are controlled by design of experiment techniques (Orsborn et al. 2009). Using this approach, Goucher-Lambert and Cagan were able to combine form attributes with functional and price attributes. Using the Eco-Indicator environmental impact assessment method, calculated environmental impact values were presented to participants during half of the preference trials (Consultants 2000). Results from the work by Goucher-Lambert and Cagan indicated that, in the context of sustainability, participants value the form of a product less, and its functional attributes more. Despite this result, there is no information regarding the rationale behind these preference decisions. Using fMRI at the time of the decision, there is an opportunity to gain an understanding of the mental processes behind sustainable preference decisions that participants are unable to express during traditional behavioral data collection methods.

fMRI and fMRI Meta-Analysis

There are several neuroimaging techniques available to researchers, one of which is fMRI. In fMRI, task-related activation can be determined by measuring changes in the levels of oxygenated hemoglobin within the brain (Huettel et al. 2004). Using this as a contrast, comparisons can be made between the levels of brain activation between different experimental conditions.

While fMRI studies are an effective tool to study brain activation data during different experimental tasks, it comes with a price. In addition to the inherent limitations of fMRI studies (limited resolution $\sim$1–3 mm^3, hemodynamic response time, etc.), brain-imaging studies are incredibly expensive and time consuming. One way to supplement collected brain imaging data is through the use of meta-analytic tools, which source brain activation patterns from thousands of external neuroimaging studies to identify associated reasoning within selected ROIs.

There are several different neuroimaging meta-analysis tools, and several ways to utilize the data. For example, Moss and Schunn (2015) used the Human Connectome Project meta-analysis data to understand brain networks associated with a reading comprehension task. In the current work, Neurosynth is used to create keyword-based ROIs, which are then used to explore data collected by the authors. Doing so provides external cross validation for the activation patterns found in the empirically collected dataset. Additionally, it provides a means to take a more focused approach to the analysis by searching over specific regions of interest, as opposed to looking for activation in the entire brain, which would involve over 10,000 independent statistical tests.

Neurosynth operates by using a text-mining approach to identify the neuroimaging studies that utilize specific terms (ex. "moral") with a high frequency (Yarkoni et al. 2011). After these studies have been identified, the tool extracts the brain coordinates for all of the areas of activation discussed in these papers. A meta-analysis of the coordinates is performed, creating a map of activation over the entire brain, which shows locations of similarly active brain regions. These results can then be thresholded using a variety of techniques (e.g., clustering) to localize more specific areas of activation. Statistical tests are then performed to examine the level of neural activation in those specific regions alone, independent from other areas in the brain, thus identifying associations between brain function and the way people think while making a choice decision involving environmental sustainability.

Methods

This experiment investigated multi-attribute preference judgments involving sustainability, with the goal of uncovering differences in these preference judgments compared to those outside the context of sustainability. The study required two sessions of data collection. The first session was a behavioral session (24 participants) and was also used to identify participants for inclusion in the fMRI session. As discussed in the Introduction, this project is part of a larger research project seeking to inform design research regarding sustainable preference decisions (Fig. 1). Under this approach are both neuroimaging and behavioral techniques. The analysis in this paper covers a meta-analytic approach to examining the neuroimaging data. The meta-analytic approach complements the whole brain analysis presented in Goucher-Lambert et al. (2016).

The methods described below cover a meta-analytic approach to analyzing and interpreting empirically collected neuroimaging data examining this question. The overall goal of the meta-analytic approach was to use specific keywords (emotion, moral, etc.) to create neural activity maps. Using these maps, an ROI analysis was performed over each individual participant's fMRI data. The goal was to determine whether or not the activation in the experimental condition containing the environmental impact information ("Environmental") was significantly different than the experimental condition without the environmental impact information ("Control") for each keyword. Determining the keyword(s) that show significant differences between the Environmental and Control conditions can help inform the mental processes used by individuals while they are considering a product's sustainability.

Individual fMRI Data

Empirically collected neuroimaging data was obtained from 11 healthy, right-handed, adults at Carnegie Mellon University. The experimental task was preference judgments involving sustainability using a within-subject visual conjoint analysis technique with two conditions (Environmental and Control). Each of the 108 trials (54 per condition) asked participants to select their preferred option between two reusable water bottle alternatives consisting of form, function, and price attributes.

The individual fMRI data was collected from a Siemens 3T Verio MR scanner, using a 32-channel phased array head coil. Functional images were acquired using a T2*-weighted echoplanar imaging (EPI) pulse sequence (31 oblique axial slices, in plane resolution 3 mm × 3 mm, 3 mm slice thickness, no gap, repetition time TR = 2000 ms, echo time TE = 29 ms, flip angle = 79 degrees, GRAPPA = 2, matrix size = 64 × 64, field of view FOV = 192 mm). In addition, high-resolution anatomical images were obtained for each participant using a T1-weighted MP-RAGE sequence.

Raw neuroimaging data was processed using the AFNI software package (June 2015 version) and converted into Talairach coordinates (Cox 1996). A full description of the preprocessing and analysis steps utilized are discussed in Goucher-Lambert et al. (2016). The individual participant data was analyzed using a voxel-wise general linear model (GLM) and separated into each of the two conditions (Environmental: Form-Function-Price-*Environmental Impact;* Control: Form-Function-Price-*Additional Function*). The activation maps from these two analyses were used as inputs into the meta-analysis in this work.

Region of Interest Meta-Analytic Mask Generation

Keyword Selection

Five keywords (moral, theory of mind, empathy, emotion, and vision) were selected for analysis. Three of these keywords (moral, empathy, and emotion) were selected based on hypothesized brain activation patterns. It was an original hypothesis that participants were altering their preference judgments due to the fact that the environment may be triggering emotional and moral thoughts. Previous research has shown moral decision making brain networks active in a task regarding the preservation of national parks (Sawe and Knutson 2015). Additionally, the valuation of attributes has been shown to activate emotional and moral networks in some decision making contexts (Greene and Haidt 2002; Greene et al. 2001; Zysset et al. 2006). Further, the prior behavioral study supported that environmental choice is a social choice decision (Goucher-Lambert and Cagan 2015).

The other two keywords (theory of mind, vision) were selected based on results from the analysis of individual and group fMRI data shown in Goucher-Lambert et al. (2016). In this work, an increase in activation was seen in some brain regions associated with theory of mind while participants were considering products with the environmental impact included. Theory of mind refers to being conscious of what others are thinking and perceiving, and how that relates to your own actions. Additionally, a decrease in activation was seen in visual processing regions during these trials. Due to the fact that these results were unexpected, they were included as keywords in the meta-analytic approach for further examination.

Neurosynth Activation Maps, and ROI Mask Generation

Meta-Analytic Brain activation maps were generated using Neurosynth for each of the five keywords chosen for analysis. Each meta-analytic map contained brain activation from different studies selected using Neurosynth's text mining algorithm. An average of 250 studies were included in each keyword's meta-analysis. Activation maps all used a z-score threshold of $z > 2.50$ to reduce spurious activation noise.

All raw keyword activation maps were opened in AFNI, and transformed into standard Talairach space using the 3dresample command. A cluster-based threshold was then applied to further reduce the impact of noise on the data. The resulting activation maps were exported as ROI masks, and used for analysis of the empirical brain activation data.

Region of Interest Analysis

The ROI analysis was completed using AFNI's 3dROIstats command. Inputs into this analysis were the keyword based ROI masks that were generated using the Neurosynth meta-analytic tool, as well as the empirically collected 11-person fMRI data set. The output of 3dROIstats was the mean voxel activation level, given as percent signal change, for each participant inside of each ROI. The mean values were obtained for both the Environmental and Control conditions and this process was repeated for each of the five keywords (moral, theory of mind, emotion, empathy, and vision).

Using the mean activation levels within each ROI, a number of factors could be determined. The main goal of this analysis was to determine if there was a (statistically) significantly different level of brain activation between the two conditions. This was established by first examining the number of ROIs that had a greater mean activation in one condition compared to the other. Next, multiple paired-sample *t*-tests were performed to determine the statistical significance of the hypothesis that the observed activation samples for each ROI came from different distributions. Each keyword was examined based on multiple criteria, including how many ROIs in the given set for each keyword had a greater level of activation in the Environmental condition, and how many of those ROIs achieved a high significance value ($p > 0.05$). Finally, Fisher's method (Eq. 1) for combined significance values was used to obtain a group p value for each keyword grouping (Fisher 1925). In Eq. 1, the chi-squared distribution with 2 k degrees of freedom can be approximated by summing the natural log of the individual significance values, p_i:

$$\chi^2_{2k} \sim -2 \sum_{i=1}^{k} ln(p_i) \tag{1}$$

Results

The keyword based meta-analytic ROI analyses were completed for each of the five topic areas described above. The results from each keyword analysis are presented below. For each keyword, there is brain map, which displays the physical location of the ROIs pertaining to that keyword, as well as a corresponding table. Each table includes the brain region(s) associated with that ROI, the mean activation level found in each ROI for both the Environmental and Control conditions, the difference value between these mean activation levels (positive taken as the Environmental condition is greater), and the significance value associated with this difference.

Emotion

The emotion keyword included seven ROIs identified by Neurosynth (Fig. 2). As mentioned previously, emotion related activation was one of the original hypotheses in this work. Of the seven ROIs explored, only four of the seven had greater mean activation levels in the Environmental condition. Furthermore, only the bilateral Dorsomedial Prefrontal Cortex ROI (ROI 6) reached a weak level of significance ($p \leq 0.1$), and none of the ROIs reached a strong level of significance ($p \leq 0.05$). ROIs 1 and 2 located in the right and left Parahippocampal Gyrus are traditionally associated with emotion related activation, neither of which reached statistical significance. Together, this illustrates that, although emotion related regions are activated in each condition, there is not a significantly *different* amount of activation between them.

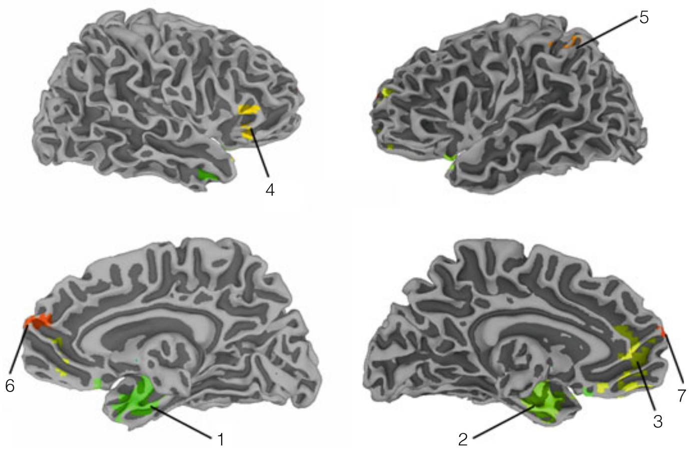


Fig. 2 Emotion keyword ROI masks. ROI numbering corresponds to Table 1

Table 1 Emotion keyword ROI analysis results

ROI	Brain regions	Environmental mean	Control mean	Diff	*p*
1	R Parahippocampal Gyrus, R Uncus	0.011	0.034	−0.023	0.149
2	L Parahippocampal Gyrus, L Uncus	0.025	0.013	0.012	0.306
3	L Orbitofrontal Cortex, L Anterior Cingulate	0.008	0.000	0.008	0.564
4	R Inferior Frontal Gyrus	−0.020	−0.054	0.034	0.282
5	L Inferior/Superior Parietal Lobule	0.167	0.177	−0.010	0.606
6	L/R Dorsomedial Prefrontal Cortex	0.023	−0.041	0.064	0.084*
7	R Dorsomedial Prefrontal Cortex	0.045	0.058	−0.013	0.592

*$p \leq 0.1$, **$p \leq 0.05$

Moral

There were five ROIs examined for the moral keyword (Fig. 3). All five of these ROIs had a greater level of activation in the Environmental condition. Additionally, two of the five ROIs were found to be strongly statistically significant. These two ROIs, a bilateral Dorsomedial Prefrontal Cortex ROI (ROI 1), and a left lateralized Superior/Middle Temporal Gyrus ROI (ROI 2) were the largest ROIs explored for this keyword and are the most often associated with moral decisions. There was perfect agreement between the positive mean difference (Environmental condition was always greater) and a high percentage of significant ROIs. Together there is strong support from the meta-analytic approach that the activation in the empirical fMRI data contains neural activity related to moral decisions while participants considered sustainability.

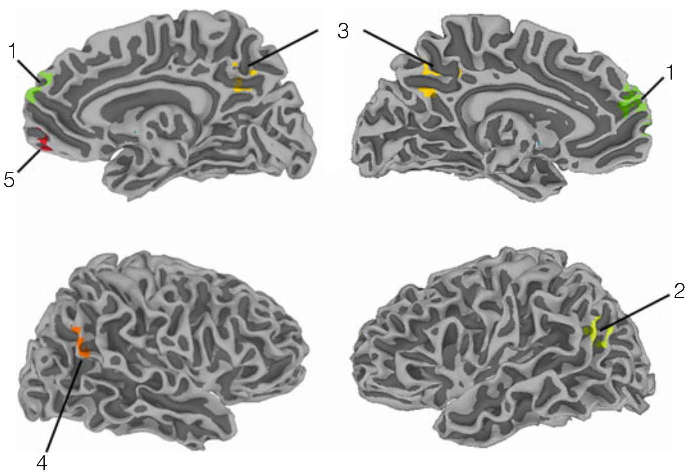


Fig. 3 Moral keyword ROI masks. ROI numbering corresponds to Table 2

Table 2 Moral keyword ROI analysis results

ROI	Brain regions	Environmental mean	Control mean	Diff	p
1	R/L Dorsomedial Prefrontal Cortex	0.046	−0.026	0.072	0.026**
2	L Superior/Middle Temporal Gyrus	0.096	0.052	0.044	0.048**
3	R/L Precuneus, R/L Posterior Cingulate Gyrus	0.101	0.081	0.020	0.402
4	R Superior Temporal Gyrus, R Supramarginal Gyrus	0.098	0.066	0.032	0.300
5	R/L Medial Frontal Gyrus	0.014	−0.021	0.035	0.169

*$p \leq 0.1$, **$p \leq 0.05$

Theory of Mind

Theory of mind activation was an unexpected result from the original group analyses of the collected fMRI data (Goucher-Lambert et al. 2016). Using the meta-analytic ROI approach presented here, more insight was gained into the presence of this activation. Theory of mind, which refers to what will others and people in society think of ones actions, recruits a very broad and extensive neural activation network. As a result there were ten ROI clusters that were used to analyze theory of mind based brain activation in preference judgments involving sustainability (Fig. 4).

Of the ten ROI clusters analyzed, all of them had higher mean activation levels in the Environmental condition, compared to the Control. Additionally, the bilateral Dorsomedial Prefrontal Cortex (ROI 1), as well as the L Middle/Inferior Temporal Gyrus (ROI 6) reached a significant threshold of $p \leq 0.05$. Several other ROIs, such as ROIs 5 and 9 fell just short of reaching weak significance threshold. However, with the two strongly significant ROIs, as well as the general trend presented with the mean activation levels, there is strong support from the

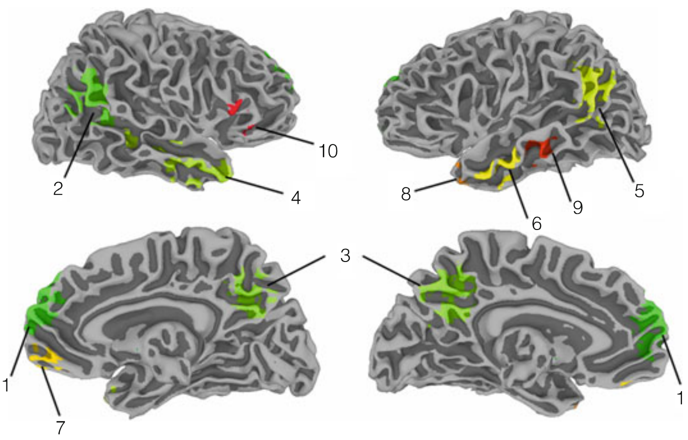


Fig. 4 Theory of mind keyword ROI masks. ROI numbering corresponds to Table 3

Table 3 Theory of mind keyword ROI analysis results

ROI	Brain regions	Environmental mean	Control mean	Diff	*p*
1	R/L Dorsomedial Prefrontal Cortex	0.057	0.007	0.050	0.042**
2	R Middle/Superior Temporal Gyrus, R Supramarginal Gyrus	0.022	−0.003	0.026	0.331
3	L/R Precuneus, L Cingulate Gyrus	0.103	0.095	0.008	0.712
4	R Inferior/Middle Temporal Gyrus	0.024	0.009	0.015	0.374
5	L Middle/Superior Temporal Gyrus, L Supramarginal Gyrus	0.054	0.019	0.036	0.122
6	L Middle/Inferior Temporal Gyrus	0.042	0.008	0.034	0.044**
7	R Medial/Superior Frontal Gyrus, R Orbital Gyrus	0.064	0.047	0.017	0.249
8	L Temporal Pole	0.013	0.001	0.012	0.485
9	L Middle Temporal Gyrus	0.043	0.012	0.031	0.106
10	R Inferior Temporal Gyrus	0.029	0.006	0.024	0.347

$^{*}p \leq 0.1$, $^{**}p \leq 0.05$

meta-analytic approach that theory of mind based activation is present when participants are considering the environmental impact of a product.

Vision

The vision ROI was selected based upon an interesting finding in previous analyses, in which the Control condition had significantly more activation in visual processing related areas (Goucher-Lambert et al. 2016). For the meta-analytic ROI analysis, there were a total of seven vision ROIs (Fig. 5). All seven of these ROIs

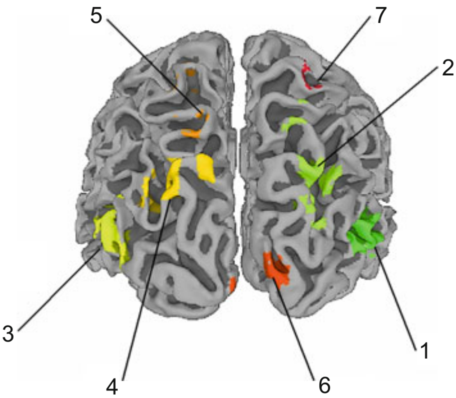


Fig. 5 Vision keyword ROI masks. ROI numbering corresponds to Table 4

Table 4 Vision keyword ROI analysis results

ROI	Brain regions	Environmental mean	Control mean	Diff	p
1	R Middle Occipital Gyrus, R Middle/Inferior Temporal Gyrus	0.176	0.211	−0.035	0.138
2	R Cuneus, R Middle Occipital Gyrus	0.517	0.544	−0.027	0.073*
3	L Middle Occipital Gyrus, L Middle Temporal Gyrus	0.136	0.158	−0.022	0.231
4	L Cuneus	0.425	0.482	−0.056	0.082*
5	L Precuneus, L Superior Parietal Lobule	0.400	0.420	−0.019	0.521
6	R Cuneus, R Lingual Gyrus	0.477	0.532	−0.055	0.265
7	R Superior Parietal Lobule	0.191	0.230	−0.039	0.290

*$p \leq 0.1$, **$p \leq 0.05$

had higher mean activation levels in the Control. Two of these ROIs show a weak level of significance ($p \leq 0.1$). It should be noted that the magnitude of the visual processing ROIs is far greater than the ROIs for other keywords. This is likely due to the heavy involvement of visual processing involved in the task, where users were looking at various designs while they were making their preference judgments. Despite this, there is strong comparative evidence of greater visual processing when the environmental impact information is not present.

Empathy

With fourteen ROI clusters, empathy represented the most widespread groupings of any of the keywords used for analysis Fig. 6. It was predicted that empathy based activation would be more present in the Environmental condition. This was due to

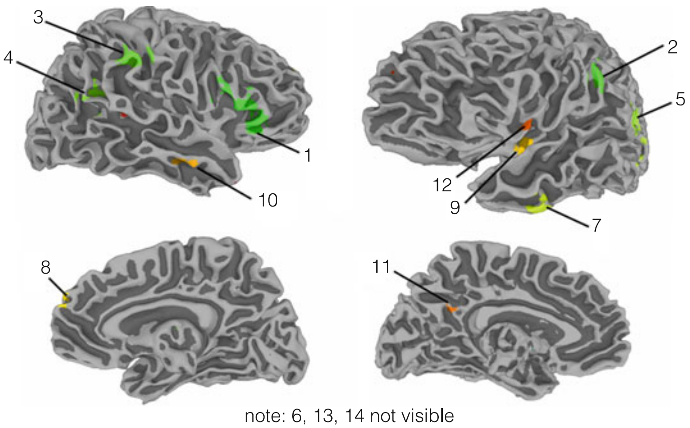


Fig. 6 Empathy keyword ROI masks. ROI numbering corresponds to Table 5

Table 5 Empathy keyword ROI analysis results

ROI	Brain regions	Environmental mean	Control mean	Diff	p
1	Inferior Frontal Gyrus	0.007	−0.022	0.029	0.290
2	L Postcentral Gyrus, L Inferior Parietal Lobule	−0.142	−0.083	−0.059	0.046**
3	R Postcentral Gyrus, R Inferior Parietal Lobule	−0.112	−0.077	−0.035	0.407
4	R Superior Temporal Gyrus	−0.043	−0.062	0.020	0.384
5	L Middle/Superior Temporal Gyrus	−0.052	−0.107	0.055	0.141
6	L Middle/Inferior Temporal Gyrus, L Middle Occipital Gyrus	0.200	0.223	−0.022	0.707
7	L Inferior Temporal Gyrus, L Fusiform Gyrus	−0.018	−0.013	−0.005	0.785
8	R Middle/Superior Frontal Gyrus	0.016	−0.119	0.135	0.023**
9	L Insula	−0.228	−0.200	−0.028	0.759
10	R Inferior/Middle Temporal Gyrus	−0.002	−0.042	0.040	0.268
11	L Cingulate Gyrus, L Precuneus	0.020	−0.013	0.032	0.347
12	L Insula	0.077	0.104	−0.027	0.932
13	L Middle/Superior Frontal Gyrus	−0.018	−0.017	0.000	0.877
14	R Superior/Middle Temporal Gyrus	−0.172	−0.158	−0.014	0.391

*$p \leq 0.1$, **$p \leq 0.05$

the hypothesis that it was necessary to care about the well being of others when caring about the environment, and this would be reflected in participants' preference decisions. However this was not the case given the results of the ROI analysis. Six of the fourteen ROI clusters had a higher mean activation level in the Control condition. This was the highest percentage of any keyword explored in this study. Furthermore, ROI 2, which straddled portions of the left Postcentral Gyrus, and left Inferior Parietal Lobule, was significantly ($p \leq 0.05$) more active in the Control condition. Conversely, ROI 8 in the Middle/Superior Frontal Gyrus was more active in the Environmental condition. Due to the fact that such a limited number of the ROIs had greater activation in the Environmental condition, and because there were significant ROIs with higher mean activations in both conditions, empathy related activation was not considered to be a unique feature of sustainable preference judgments.

Discussion

Discussion of Results

From a high level perspective, this paper examined five separate keyword based meta-analytic ROI groups related to various hypotheses regarding the way

Table 6 Summary of results from meta-analytic ROI analysis

Keyword	# of ROIs	Environmental > Control	$p < 0.1$	$p < 0.05$	Fisher's group p
Moral	5	5	2	2	0.02
Theory of mind	10	10	2	2	0.04
Empathy	14	7	2(1)	2(1)	0.25
Emotion	7	4	1	0	0.26
Vision	7	0	3(3)	0	0.05

Included are the number of ROIs explored, how many ROIs had a greater mean neural activation level in the Environmental condition compared with the Control condition, the number of ROIs that reached significance thresholds, and the group p statistic calculated using Fisher's method of combined significance. Numbers inside parentheses indicate significant clusters where Control > Environmental

preference judgments are made in the presence of environmental impact information. Table 6 summarizes the results from of these analyses, and includes the total number of ROIs that were examined for each keyword. Specifically, Table 6 includes the number of ROIs that had a higher level of activation in the Environmental condition compared to the Control, and the number of ROI groups that reached different levels of significance thresholding.

It should be noted that achieving a high level of significance for *every* ROI within a keyword's set of explored ROIs is not a necessary requirement to determine that there was support for the existence of a keyword's neural activation (Bartra et al. 2013). Each keyword examined in this work utilizes a broad brain activation network to support its function. Finding significant activation within every ROI associated with a keyword's set is highly unlikely. Furthermore, it is possible that each ROI within a keyword's set supports a vastly different feature of that keyword's function. Thus, finding multiple significant clusters, as well as the general trend from the non-significant clusters, together is sufficient to illustrate the presence of a keyword's activation. To understand the overall significance of each keyword, Fisher's method of combined significance is used.

In Table 6, the keywords have been positioned in decreasing order in terms of their "support" of a higher level of activation in the Environmental condition. For example every ROI for the "moral" keyword had a greater level of mean activation in the Environmental condition, and two of the five ROIs achieved a high level of significance. This is considered a strong level of support for moral activation. In addition to moral activation, the data indicates that there is strong support for theory of mind activation under the same conditions. For theory of mind there were ten ROIs explored, each of which had a higher mean activation level in the Environmental condition. Additionally, there were 2 ROIs that were strongly significant within this set. It should be pointed out that the moral and theory of mind keywords shared some overlapping regions (for example the Dorsomedial Prefrontal Cortex). Aspects of these two processes have been shown at times in previous research to support each other (Greene and Haidt 2002). It is possible that in the task presented here, individuals are engaging in moral reasoning by considering what others would think of their decisions. It is this consideration of what others are thinking that is a theory of mind process.

Overall, there was no conclusive support for significantly different fMRI activation in the empathy or emotion keywords. While the activation from emotion or empathy was not significantly *different* between the two conditions, this activation was still present in both. It is more likely that this activation was a general feature of product preference decisions, and not something that is unique to product preference decisions involving sustainability. For example, for empathy there were fourteen ROIs explored. Of these, seven had a higher level of activation in the Environmental condition. Furthermore, there were 2 ROIs that reached that reached $p \leq 0.05$. However, one of these ROIs was significant with a negative difference (Control had higher levels of activation than Environmental). Fisher's method of combined significance was used to obtain overall significance values for each keyword ROI cluster. The results from this analysis confirmed (and were in perfect agreement with) the conclusions from examining the proportion of ROIs with higher activation levels, as well as the number of ROIs with (individually) highly significant values.

Some Implications for Design

Using neuroimaging-based approaches, researchers are able to uncover physical evidence relating to observed behavior. This is a powerful ability, and the design research community is only just starting to scratch the surface of its potential to help answer difficult open questions. In terms of the sustainable preference judgments studied in this work, the evidence can help change the way the community creates tools to assist designers in developing environmentally smart products, and the way products and services are designed to be more sustainable. With the use of neuroimaging, this work was able to gain insights into participant decision-making in ways that would be difficult otherwise.

The reduced activation in visual processing based brain regions in the Control condition supports the behavioral results found in the study by Goucher-Lambert and Cagan (2015). Specifically, those results showed that in the context of environmental impact information, participants value the form of a product less. Results from the current study demonstrate that, on a neural level, individuals are putting less effort into examining visual based stimuli when the environmental impact information is present. With this increased evidence, designers and product developers should put additional effort into ensuring that the functional requirements of environmentally friendly products are met. According to the findings in this work form is a less important attribute to consumers when evaluating products under the consideration of sustainability. Future work should consider whether this effect lessens as products (like a car) become more associated with a person's identity. It is unclear how a product's increased connection with ones identity may influence the theory of mind based brain activation found to be active in this work during sustainable preference judgments. Additionally, the ROI analysis data show that individuals are heavily considering what others may think of their actions (theory of mind) and even what the moral and ethical implications of their decisions may be (moral). Emotional and empathetic reasoning appear to be a feature of difficult decisions in general, and not a unique characteristic of sustainable preference judgments. This would be in line with the findings from Sylcott et al. (2013) in which emotion related brain regions were active in form-function conflict decisions. Overall, it appears that there is awareness from participants regarding the meaning of the environmental impact metrics, and that they are considering how this may appear in the minds of other people. Using this information, one strategy could be for design practitioners to increase efforts to engage the emotions of consumers when designing sustainable products, rather than, for example, trying to educate consumers on environmental issues. The emotional and empathetic connection to the environment needs to be more present for individuals to become personally invested, requiring increased design focus. Previous work has shown the importance that emotion plays in captivating consumers (Boatwright and Cagan 2010). If people are already aware of the moral issues involving the environment, as well as considering what others are thinking (theory of mind), then increasing emotional and empathetic responses may be the key to the success of future sustainable products.

Conclusion

In this work neuroimaging data pertaining to sustainable preference judgments was analyzed using a meta-analytic ROI approach based upon keywords (emotion, moral, theory of mind, empathy, and vision). With this approach, significant evidence was found supporting the presence of brain activation data relating to theory of mind and moral decision making, while evaluating products with their environmental impact displayed. Emotion and empathy related neural activation was present in both preference decisions with environmental impact information and without. Additionally, there was a decrease in visual processing during preference judgments involving sustainability. By combining these meta-analytic neuroimaging results with previous behavioral results, there is a more complete picture regarding participants' preferences that would be challenging at best to obtain from purely behavioral studies. The meta-analytic approach in this paper greatly strengthens additional fMRI analyses by validating empirically collected brain activation data with externally collected brain activation data. Further research is needed to explore the consistency of the results presented in this paper across products and people.

Acknowledgements The authors would like to thank the staff at the Carnegie Mellon University Scientific Imaging and Brain Research Center for their support during fMRI data acquisition. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant DGE125252 and the National Science Foundation under Grant CMMI1233864.

References

Bartra O, McGuire JT, Kable JW (2013) The valuation system: a coordinate-based meta-analysis of BOLD fMRI experiments examining neural correlates of subjective value. NeuroImage 76:412–427. doi:10.1016/j.neuroimage.2013.02.063

Bernstein W, Ramanujan D, Devanathan S, Fu Z, Sutherland J, Ramani K (2010) Function impact matrix for sustainable concept generation: a designer's perspective (DETC2010-28340). In: Proceedings ASME 2010 international design engineering and technical conferences, pp 377–383

Blamey RK, Bennett JW, Louviere JJ, Morrison MD (1999) Validation of choice model involving green product choice. Choice modelling research reports no. 10. University of New South Wales, Canberra

Boatwright P, Cagan J (2010) Built to love, 1st edn. Berrett-Koehler Publishers Inc, San Francisco

Consultants P (2000) Eco-indicator 99 manual for designers. Ministry of housing, spatial planning and the environment (October). http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Eco-indicator+99+Manual+for+Designers#0

Cox RW (1996) AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. Comput Biomed Res Int J 29(3):162–173. doi:10.1006/cbmr.1996.0014

Devanathan S, Ramanujan D, Bernstein WZ, Zhao F, Ramani K (2010) Integration of sustainability into early design through the function impact matrix. J Mech Des 132 (8):081004. doi:10.1115/1.4001890

Fisher R (1925) Statistical methods for research workers. Biol Monogr Man. doi:10.1056/NEJMc061160

Fitzgerald DP, Herrmann JW, Schmidt LC (2010) A conceptual design tool for resolving conflicts between product functionality and environmental impact. J Mech Des 132(9):091006. doi:10.1115/1.4002144

Fleming SM, Thomas CL, Dolan RJ (2010) Overcoming status quo bias in the human brain. Proc Natl Acad Sci USA 107(13):6005–6009. doi:10.1073/pnas.0910380107

Goucher-Lambert K, Cagan J (2015) The impact of sustainability on consumer preference judgments of product attributes. J Mech Des 137(August):1–11. doi:10.1115/1.4030271

Goucher-Lambert K, Moss J, Cagan J (2016) The truth in the decision: using neuroimaging to understand multi-attribute product preference judgements involving sustainability. J Mech Des (Submitted) 1–11

Greene J, Haidt J (2002) How (and where) does moral judgment work? Trends Cognit Sci. doi:10.1016/S1364-6613(02)02011-9

Greene JD, Sommerville RB, Nystrom LE, Darley JM, Cohen JD (2001) An fMRI investigation of emotional engagement in moral judgment. Science 293(5537):2105–2108. doi:10.1126/science.1062872

Huettel SA, Song AW, McCarthy G (2004) Functional magnetic resonance imaging. ISBN: 0878932887

Macdonald E (2012) Seven cognitive concepts for successful sustainable design (DETC2012-70676). In: ASME 2012 international design engineering technical conferences 1–16

MacDonald E, Gonzalez R, Papalambros PY (2009) Preference inconsistency in multidisciplinary design decision making. J Mech Des 131(3):031009. doi:10.1115/1.3066526

Masui K, Sakao T, Kobayashi M, Inaba A (2003) Applying quality function deployment to environmentally conscious design. Int J Qual Reliab Manag 20(1):90–106

Moss J, Schunn CD (2015) Comprehension through explanation as the interaction of the brain's coherence and cognitive control networks. Front Hum Neurosci 9(October):1–17. doi:10.3389/fnhum.2015.00562

Nederhof AJ (1985) Methods of coping with social desirability bias: a review. Eur J Soc Psychol 15(3):263–280

Orsborn S, Cagan J, Boatwright P (2009) Quantifying aesthetic form preference in a utility function. J Mech Des 131(6):061001. doi:10.1115/1.3116260

Peattie K, Belz F-M (2010) Sustainability marketing—an innovative conception of marketing. Mark Rev St. Gallen 27(5):8–15. doi:10.1007/s11621-010-0085-7

Peattie K, Charter M (2003) Green marketing. The Marketing Book 726–755. ISBN: 9780874216561

Poldrack RA (2007) Region of interest analysis for fMRI. Soc Cogn Affect Neurosci 2(1):67–70. doi:10.1093/scan/nsm006

Ramani K, Ramanujan D, Bernstein WZ, Zhao F, Sutherland J, Handwerker C, Thurston D (2010) Integrated sustainable life cycle design: a REVIEW. J Mech Des 132(9):091004. doi:10.1115/1.4002308

Reid TN, Gonzalez RD, Papalambros PY (2010) Quantification of perceived environmental friendliness for vehicle silhouette design. J Mech Des. doi:10.1115/1.4002290

Sawe N, Knutson B (2015) Neural valuation of environmental resources. NeuroImage. doi:10.1016/j.neuroimage.2015.08.010

She J, MacDonald E (2013) Priming designers to communicate sustainability. J Mech Des 136(1):011001

Spector P (2004) Social desirability bias. In: The SAGE encyclopedia of social science research methods, 1045–1046. doi:10.1002/9781444316568

Sylcott B, Cagan J, Tabibnia G (2013) Understanding consumer tradeoffs between Form and function through metaconjoint and cognitive neuroscience analyses. J Mech Des 135(10):101002

Yarkoni T, Poldrack RA, Nichols TE, Van Essen DC, Wager TD (2011) Large-scale automated synthesis of human functional neuroimaging data. Nat Methods 8(8):665–670. doi:10.1038/nmeth.1635

Zysset S, Wendt CS, Volz KG, Neumann J, Huber O, von Cramon DY (2006) The neural implementation of multi-attribute decision making: a parametric fMRI study with human subjects. NeuroImage 31(3):1380–1388

Second Guessing: Designer Classification of Problem Definition Fragments

Meghna Polimera, Mahmoud Dinar and Jami Shah

Abstract Engineering designers progressively develop their own understanding of ill-defined problems through a process of abstraction, decomposition, completion, enhancement, classification and conflict resolution. We have developed a computational aid to support problem formulation in which designers input problem definition fragments into different categories as free form text. We used natural language processing to determine if designers had misplaced problem fragments in inappropriate categories. In this paper, we present our work on how this problem can be addressed by looking for keywords in design descriptions and extracting knowledge from text ontologies using these keywords. We collected data from a group of students who used the Problem Formulator testbed to express their understanding of the given design problem. For each of the six categories in the Problem Formulator ontology, we identified classes using existing ontologies and tools that closely define them. In our method, we first parsed the user inputs to extract the keywords depending on the category in which they were entered. We then associated these keywords to the previously identified classes and categorized them into the correct category.

Introduction

Engineering design problems are ill structured and working on them is a knowledge-intensive task. To verbally express, or textually represent designing, an overwhelming amount of words is needed, especially at the early stages of conceptual design, and without any restriction on using a predefined vocabulary or ontology. One way to overcome this challenge is to adopt an ontology whereby designers express their problem formulation in a structured and more limited rep-

M. Polimera · M. Dinar · J. Shah (✉)
Arizona State University, Tempe, AZ, USA
e-mail: shah.493@osu.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_11

resentation compared to completely free-form verbalization. In previous work, we have developed Problem Formulator, a web-based tool designed for designers to express their understanding of design problems. Preliminary analysis of data collected from students using the tool revealed that the major issue was that the students ascribed fragments of their problem definitions to inappropriate categories in the ontology. For example, though the students were taught to write functions in form of action verbs (as a description of some action that the design performs), problem fragments such as *"screen to display videos"* (which is an artifact in our ontology) were often added under functions. Even if a predefined ontology was used, the number of possible words that a designer might use in describing the conceptual design process of a problem can still be significantly large and diverse. Description of different problems by different designers will be even more verbose. This verbosity adds to the problem of misuse of the ontology, and thus to manually evaluate the descriptions of design problems becomes quite challenging and resource consuming. We therefore used Natural Language Processing techniques to address this problem of verification of user classification of problem entities and relations.

In studies of applying automatic text categorization in engineering design, Cheong and Shu (2012), and Glier et al. (2014) have emphasized the application of natural language processing techniques to automatically categorize text in Bioinspired design. Since, applying classification techniques in conceptual design has not been explored, we came up with a computational solution that automatically classifies words or short phrases that designers use in formulating design problems. In particular, we identified classes, which were hypernyms from WordNet (2010), relations from K-parser (Sharma et al. 2015), and modals extracted using Stanford parser (Klein and Manning 2003), that closely define the six categories of the Problem Formulator testbed. In our method, we first parsed the user inputs to extract the keywords depending on the category in which they were entered. We then associated these keywords to the previously identified classes and categorized them into the correct category.

Text Classification Using WordNet

Machine learning approaches are well known for performing automatic text classification (Sebastiani 2002). However, the majority of applications use the traditional *Bag of words* representation of text where each term is treated as an independent feature (Scott and Matwin 1999). This is proven to show unsatisfactory results due to neglecting the relations between the terms in sentences (Elberrichi et al. 2008). For this same reason, a traditional machine learning technique might not produce good classification results for the nature of the data we have. The best example to explain this can be the following two sentences, "the snake ate the man" and "the man ate the snake". In a *Bag of Words* approach, these two sentences would have no difference between them as it treats each term as an independent feature. For better classification of data, it is crucial to understand the meaning of the sentence rather than having just the frequency of words.

Several applications use WordNet for Text Classification. Elberrichi et al. (2008), unsatisfied with the results obtained by the Bag of words representation of text, proposed a method to use WordNet to categorize text documents, where WordNet was used to extract the generic concept of terms and all different combinations of these words to form a new representative vector.

de Rodríguez et al. (1997) on the other hand used WordNet knowledge to improve the performance of text categorization systems by integrating it with Rocchio and the Widrow-Hoff algorithms on the Reuters-21578 data set.

The advantage of using WordNet in our project is the hierarchal structure of the words it contains (Miller 1990). The hypernyms of the extracted keywords are matched with one of the predefined classes and are classified to a category based on that. However, using this technique alone was insufficient due to the variation in the type of data we collected for each category of the Problem Formulator testbed. We therefore, incorporated two other techniques that used Stanford parser and K-parser, each of which is explained in the implementation section.

Data Collection Testbed

The Problem Map Framework

The Problem Map ontological framework (MacLellan et al. 2013; Dinar et al. 2015) consists of six types of categories: requirements, use scenarios, functions, artifacts, behaviors, and issues. Each group has a few sub-types (with the exception of the function category); e.g., objective is a sub-type of requirements, and question is a sub-type of issues. Entities in each group can form a hierarchical structure with disjunctive branches as alternative combinations, which achieve a parent node. Entities in different groups can also be linked. Figure 1 shows the categories and the relations among them within the Problem Map ontology. The ontology was taught to the subjects before data collection with specific definitions and examples of each type of category. The subjects were then given design problems and asked to express their thinking about the problem within the provided structure, i.e., they were asked to freely express their thinking about different aspects of the problem but specify the fragment as one of the given categories. For example, when a subject was asked to work on designing a water-sampling device, the subject might specify "should collect water samples" as a requirement, "submerge" as a function, "bladder" as an artifact, and "buoyancy" as a behavior.

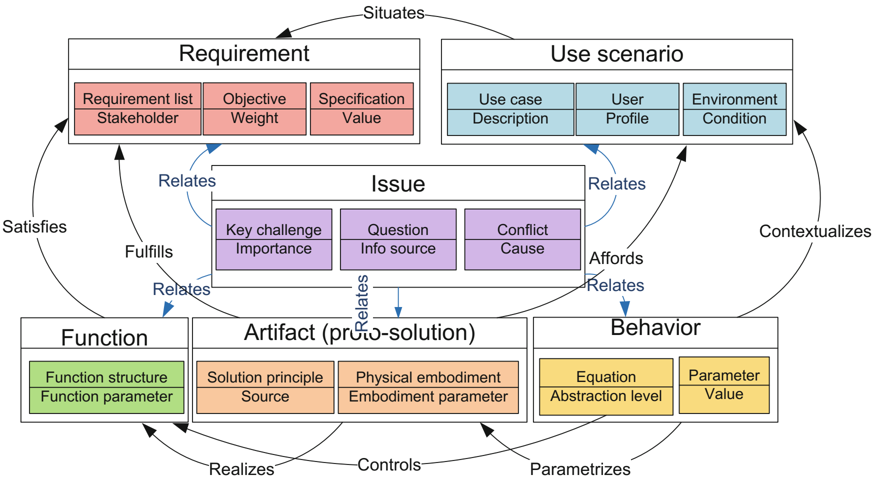


Fig. 1 The entities and their relations in the Problem Map ontology

The Problem Formulator Tool

The Problem Formulator web-based testbed was used for collecting designers' thoughts about a problem in the form of short text phrases. The testbed is based on the Problem Map ontological framework, which was also used in studying differences in designers' problem formulation. Figure 2 is a snapshot of the main user-interface of the web-based tool. The folder icon represents a parent node while integers (excluding 1) on the folder icon represents the number of disjunctive branches under the parent node.

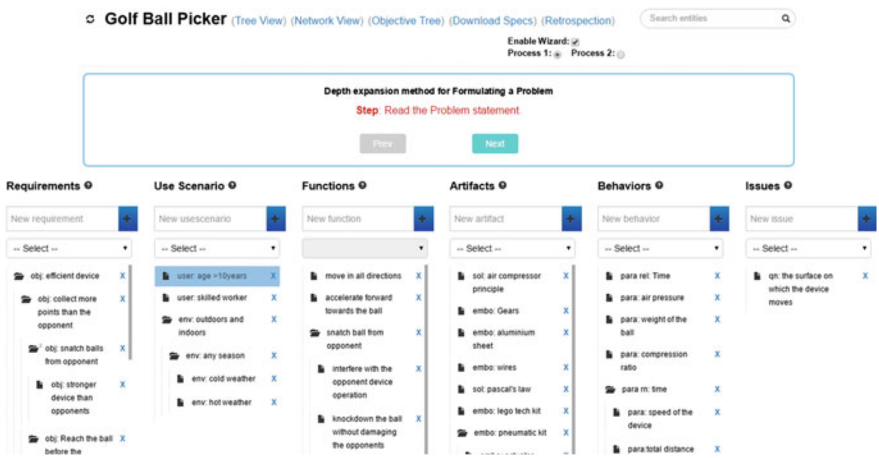


Fig. 2 A snapshot of the Problem Formulator web-based tool

Design and Implementation of Second Guessing Categorization

In this paper, we discuss techniques involved in deciding whether or not the user has placed the problem fragment into the appropriate category, *Artifacts, Behaviors, Use Scenarios, Requirements, Issues, and Functions*. The purpose of this research was to identify the wrongly placed problem fragments, determine which category they could belong to and then prompt the user with the possible correct categories. The automatic classifier we built in this paper takes two inputs, the first is the user given design description and second, the category in which the user places this description. Based on the category the input is placed in, the classifier decides whether the input belongs to that category. If it does not, then the possible correct categories are determined by checking the input with all the remaining ones. Since the tool implemented is not 100% accurate, there is a possibility for an input to be classified into multiple categories, which is dealt with in the later phases.

The following is an example of a design problem,

> "A basketball returning device needs to be designed which automatically returns the ball to the shooter. The device is to be designed for the age group of 10-18 years. This device should fit to any kind of hoop. The device should be easily transportable and storable in a small place. The device should not hinder the shooter's access to the basket and should be able to return by at least 24 feet. The device should be cheap and extremely safe to use."

Data Structure and Preprocessing

The data we used in this project are inputs from the problem formulator tool. Students from a graduate advanced product design class familiar with the working of the tool were given certain design problems and were asked to use this tool to input data into the 6 categories according to their understanding of the problem. We analyzed the data and observed that the inputs given by the users were grammatically faulted, poorly structured with lengths of inputs ranging from one word up to 15 words. The inputs varied from candidate to candidate and required significant pre-processing that involved removal of numbers, single letter words, parenthesis, and punctuations. We, however, retained the stop words as they played an important role in its syntactic structure. Table 1 shows two examples of the data before and after pre-processing,

Classification Using **Hypernyms** *from WordNet*

Artifacts, Behaviors, and Use Scenarios, three of the six categories of the problem formulator tool, have a similar structure in the data. Artifacts also known as

Table 1 Examples of data before and after pre-processing

Sample input data from the user of the Problem Formulator	
Before pre-processing	After pre-processing
1 12v DC reversible drive motor	Reversible drive motor
Parameter: Max. Translation in x-axis (feet)	Parameter Max Translation in x axis feet

physical elements, components, or devices are described as any entity that is tangible. The common examples include paper, wood, gears, motors, etc. Behaviors, on the other hand are identified as physical properties or mathematical properties associated with artifacts. Examples are length, breadth, volume, permeability, tension, pressure etc. Use Scenarios describe the usage of the product being designed. For example, who will use the design, or how it is used, and the conditions in which it is used etc.

An approach using syntax and semantics of the data is used to determine the category to which it belonged. The method implemented in this project consists of two steps. The first step focuses on the syntactic structure of the data and the second step, its semantics.

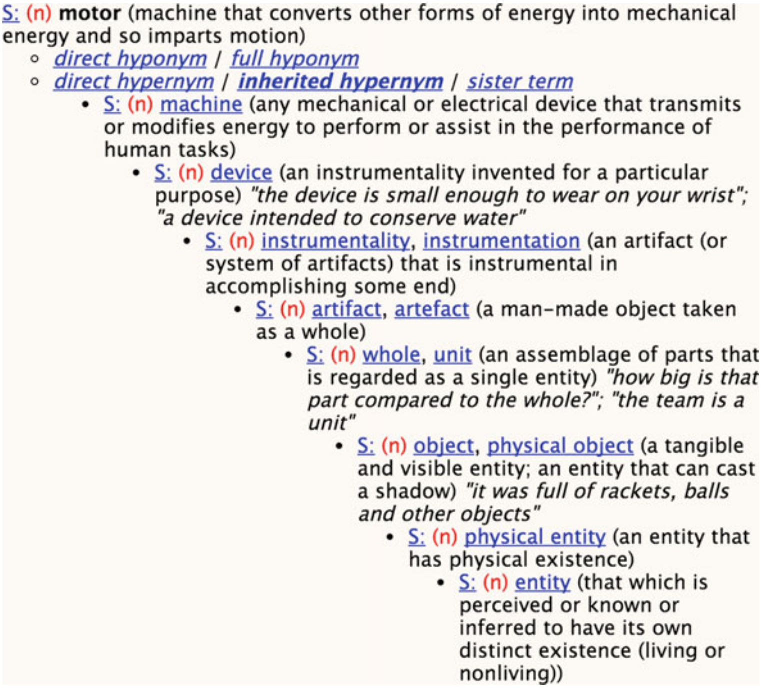
The syntactic structure of the data played an important role in determining the subject of the sentence, which is the matter of concern in the categorization here. This was obtained by identifying the parts of speech of the input text. For this, we used the Stanford parser, a natural language parser, which works out the grammatical structure of a sentence. For example, in the sentence, “pressure gauge”, the subject is *gauge* since we are talking about the *gauge*. On the other hand, in the sentence, “pressure of a gauge”, the subject is *pressure*. Once the subject is obtained, we move to the second step, i.e., the semantic analysis of the identified subject.

Semantic analysis is a technique of understanding what the meaning of the sentence is, in our case, the meaning of the subject of the sentence. For this, we use WordNet. For artifacts, from a sample data of correctly categorized user inputs by experts, we used the Stanford parser to obtain the nouns (subject) from each of these inputs and used WordNet to get a set of common hypernyms which are shown in Table 2.

As an example, consider the noun *motor*. Figure 3 shows the hierarchal structure WordNet gives for this noun. WordNet produces a similar structure for *gear*, where *device* is the first common parent in their hierarchy. Therefore, *device* can be considered as the most specialized class. The other nouns such as *wheels, springs, sheets, compressor, valves*, and *wires* also have the same structure and hence belong to the class *device*. However, nouns such as *cords,* and *tubing* have a slightly different structure and the first common hypernym in their trees is *artifact*. WordNet also shows that *device* is a specialized form of an *artifact*. Hence, just by including the class *artifact*, which is a common hypernym for them all, we covered most of the cases. Upon similar analysis on the sample data set, we narrowed down a list of four classes for artifacts, which are shown in Table 2.

Table 2 Pre-defined classes (hypernyms) for *artifacts* obtained from WordNet

Pre-defined classes for Artifacts	
Name of the pre-defined class	Definition from WordNet
Material Examples: cardboard, paper	Tangible substance that goes into the make up of a physical object
Matter Examples: steel, copper	That which has mass and occupies space
Artifact Examples: table, screws	A man-made object taken as a whole
Part Examples: handle, grip	Something less than the whole of a human artifact



S: (n) **motor** (machine that converts other forms of energy into mechanical energy and so imparts motion)
- *direct hyponym* / *full hyponym*
- *direct hypernym* / ***inherited hypernym*** / *sister term*
    - S: (n) machine (any mechanical or electrical device that transmits or modifies energy to perform or assist in the performance of human tasks)
        - S: (n) device (an instrumentality invented for a particular purpose) *"the device is small enough to wear on your wrist"; "a device intended to conserve water"*
            - S: (n) instrumentality, instrumentation (an artifact (or system of artifacts) that is instrumental in accomplishing some end)
                - S: (n) artifact, artefact (a man-made object taken as a whole)
                    - S: (n) whole, unit (an assemblage of parts that is regarded as a single entity) *"how big is that part compared to the whole?"; "the team is a unit"*
                        - S: (n) object, physical object (a tangible and visible entity; an entity that can cast a shadow) *"it was full of rackets, balls and other objects"*
                            - S: (n) physical entity (an entity that has physical existence)
                                - S: (n) entity (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Fig. 3 Hierarchy of *motor* in WordNet

One might wonder why we did not consider *physical entity* to be a class, which is a common hypernym of both *material* and *artifact*. The reason for this is that *physical entity* is almost in the topmost level of the hierarchy and many other nouns that do not belong to artifacts may fall under this. So, if we included *physical entity* as one of our classes, nouns that are not artifacts would be classified as artifacts. An example for this would be the word *phenomenon*. We know *phenomenon* is a procedure and not an artifact but if *physical entity* were one of our pre-defined

```
Algorithm 1: Artifacts Classification
begin
  subject = Stanford_parse(input);
  Retrieve all synsets of a subject using WordNet;
  for all synsets
        extract hypernyms
        if hypernym is one of the 4 pre-defined classes
           then input is an artifact;
        else
            extract hypernym of the current hypernym;
        end
  end
end
```

Fig. 4 Algorithm to classify data into Artifacts

classes, the tool would have classified *phenomenon* as an artifact. Based on this reasoning, we concluded that every noun that belonged to one of the four classes, *matter, part, artifacts,* or *materials*, is in fact an artifact. Figure 4 shows an algorithm on how this works.

The same approach is used to classify behaviors, and use scenarios. The difference lies in the classes obtained from WordNet. The algorithms for each of these two categories are same as that of Artifacts. Table 3 and Table 4 show what classes are obtained for behaviors and use scenarios respectively.

Classification Using Modals *Extracted Using Stanford Parser*

Unlike the previously discussed categories, requirements and, issues have a slightly different definition and structure in the data. Therefore, it was not feasible to use the same technique discussed previously. Since requirements define the high level specifications and more about what the design needs, it was necessary to check if the input relayed this information. In order to do this in a problem independent way, we observed in a set of sample data that all the requirements usually had words like *should, would, could, must* etc, and these could be identified by extracting the modals from the input using Stanford parser. The following are the set of modals that we used to check if an input is a requirement,

Should, Could, Would, Must, Shall, Need, Can, Will.

Issues on the other hand are defined as concerns related to the design, challenges, or questions about additional information. These usually imposed a negative sense in the inputs. Therefore, we checked if the modals in the inputs were negative. This helped categorizing them as Issues when the input type was a concern. The following are the set of modals for issues,

But, Can't, Cannot, Cant, Wont, Won't.

However, these only cover the concerns part of issues. For questions, it was observed from the sample data that the inputs were usually interrogative sentences.

Table 3 Pre-defined classes for behaviors obtained from WordNet

Pre-defined classes for behaviors	
Name of the pre-defined class	Definition from WordNet
Physical phenomenon Examples: voltage, power	a natural phenomenon involving the physical properties of matter and energy
Magnitude Examples: breadth, height	The property of relative size or extent
Magnitude relation Examples: frequency, speed	It is a quantitative relation between two magnitudes
Physical property Examples: weight, length	Any property used to characterize matter and energy and their interactions
Fundamental quantity Examples: length, time	One of the four quantities that are the basis of systems of measurement
Consistency Examples: viscosity, permeability	The property of holding together and retaining its shape
Movement Examples: amplitude, translation	A change of position that does not entail a change of location
Space Example: angle	An empty area usually bounded in some way between things
Measure Example: liter	how much there is or how many there are of something that you can quantify

Table 4 Pre-defined classes for use scenarios obtained from WordNet

Pre-defined classes for Use scenarios	
Name of the pre-defined class	Definition from WordNet
Location Examples: north, south, home	A point or extent in space
Condition Examples: physical condition	A state at a particular time
Surface Examples: vertical surface	The outer boundary of an artifact or a material layer constituting or resembling such a boundary
Time Period Examples: time, duration	An amount of time
Structure Examples: building garage	A thing constructed
Organism Examples: human, fish	A living thing
Natural phenomenon Examples: Earthquake, floods	All phenomena that are not artificial

Hence, if the input started with one of the words below, we classified them as issues.

No, Why, Will, What, Can, When, Where, How, Is, Who, Does, Do.

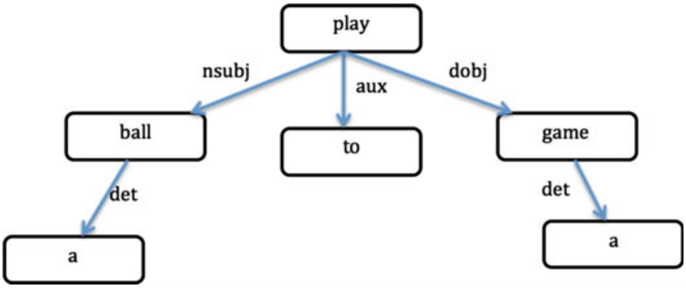


Fig. 5 Stanford dependency parse tree for “A ball to play a game”

Classification Using Relations from K-Parser

K-Parser (Sharma et al. 2015) is a graph based semantic parser, which represents event-entity and inter-event relations (KM Ontology) (Clark et al. 2004; de Marneffe et al. 2006). Given a sentence, it finds the semantic relations between entities by mapping the syntactic dependency relations to the KM relations obtained by using syntactic dependency parser such as Stanford dependency parser (De Marneffe and Manning 2008). There are total of 118 km relations. It also represents commonsense knowledge such as semantic roles of entities participating in the sentence.

For example the Stanford dependency parse tree for the sentence “A ball to play a game” can be seen in Fig. 5. The dependency relations shown in the graph are the grammatical relations as defined in De Marneffe and Manning (2008). Figure 6 displays the semantic parse graph obtained from the K-Parser for the same sentence. The Stanford dependencies are mapped to Semantic relations from KM ontology.

We used this tool to categorize inputs into the final category in the Problem Formulator tool, functions. These are expressed in the form of relations between

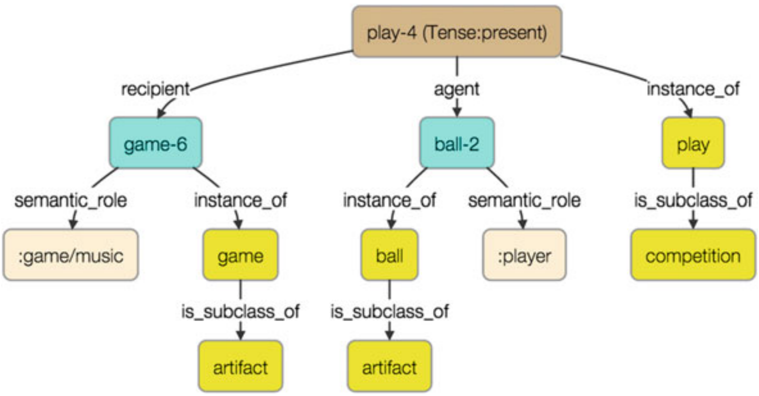


Fig. 6 A snapshot of the semantic graph in K-Parser

Table 5 Pre-defined relationships obtained by parsing sample sentences in K-Parser

Pre-defined relationships for functions			
Sample sentence	Noun/adverb	Verb	Pre-defined relationship
prevent opponent from interfering	Noun: opponent	Verb: prevent	recipient
Move forward	Adverb: forward	Verb: move	intensity
shoot the ball into the basket	Noun: basket	Verb: shoot	destination
Throw the stone at the window	Noun: Window	Verb: Throw	site
Collect the ball from the basket	Noun: Basket	Verb: collect	origin

verbs and nouns, and verbs and adverbs, which describe what the design needs to do. Unlike the other categories in the tool where only the keywords could be used for classification, it was necessary to capture the meaning of the sentence as a whole for this category. Therefore, to identify the relations between words in the inputs, we used K-Parser.

In the K-Parser, relationships between the verbs and nouns, and verbs and adverbs are given a unique name based on their usage in the sentence. We used a set of these relationships as our basis of classification. To do so, we selected a sample data and obtained the relationships from K-Parser to classify them as functions, which are, Recipient, Site, Intensity, Destination and Origin. Table 5 shows sample sentences and relationships that hold between the nouns and verbs, and nouns and adverbs. Since the size of the input could also be a single word, we only check if it is a verb and classify it as a function without using the K-Parser.

Results and Discussion

Test Data

For testing, we collected data from 25 students in mechanical engineering for a design problem, which was different from the one that was used for the sample data. On an average, a total of 300 inputs were collected from each student. All the inputs were combined and separated category-wise. For the final analysis, 50 entries were chosen at random from each category in a randomized manner and the reason for choosing only 50 in each is explained in the next section.

Table 6 shows examples of problem fragments described by students for the design problem, *"Design an autonomous surveillance vehicle to automatically and periodically tour the perimeter of two structures, stopping as close as possible to the start point."*

Table 6 Problem fragments in each category

Issue	Requirement	Use scenario	Behavior	Artifact	Function
What is the maximum number of batteries can be used?	Should have an index mark	Nature of turf	Friction	Wheels	Turn in all directions
Max turn radius allowed?	No touching the boxes	Night time	Time	Gears	Collect balls
Number of wheels allowed?	Fit in $15 \times 15 \times 15$ inches box when fully retracted	A single person should assemble and start the vehicle	Motor RPM	Storage Compartment	Store on device

Machine Versus Human: Comparison of Classification Outcome

A common method of determining unambiguousness of ontology is to find the inter-rater agreement, an examination of how different raters agree on classifying data into certain pre-defined categories.

There are two possible ways of calculating this, a three-way or multi-way inter-rater agreement among one machine and multiple raters, e.g., using Krippendorf's Kappa (1980); or pairwise comparison of machine with each rater and two raters with Cohen's alpha (1960).

To measure the inter-rater agreement in our problem, we followed Cohen's Kappa. The data set collected from the students was given to trained raters who were asked to validate the inputs by marking either a *yes* or a *no* depending on the category in which they were placed. The tool also tagged every input in each category as either a *yes* or a *no* depending on whether or not it satisfied the requirements of that category. The required sample size was found to be 278 (rounded out to 300), based on Gwet (2008) for an expected agreement of 70% among the raters, and an expected 30% error in coding for each rater. However, the probability that a rater might choose a response purely by chance should be considered in determining the sample size. Since the raters were asked to provide a yes/no response for each of the segments taken from the users (who labeled their input data as one of the categories function, artifact, or behavior), the 50% probability of choosing one of the two options by chance would lower the agreement rate considered in sample size determination to 20.

The results following a pair-wise comparison with Cohen's Kappa and a sample size of 300 with 50 randomly selected inputs from each category, show that the agreements are 0.84 between the two trained raters, 0.68 between one of the trained raters and the tool, and 0.64 between the other rater and the tool. Table 7 shows the

Table 7 Responses by the raters and the tool

	Issue	Requirement	Use scenario	Behavior	Artifact	Function
Rater 1	37	39	50	48	49	37
Tool	19	40	32	41	41	44
Rater 2	49	50	49	50	49	50

consensus between the raters, and the tool for each category. The values in the cells represent the number of *yes'* in each category given by the rater and the tool.

It is quite evident from the table that the tool's performance is fairly poor in identifying issues and use scenarios. This is because the techniques implemented to identify inputs in these categories are very minimal and do not cover all the cases. Whereas, similar techniques were applied for other categories and the results show that the tool performs comparatively well for requirements, behaviors, artifacts, and functions.

Challenges

For any classification method to be successful, in a machine learning technique, the training data plays a crucial role as it decides the basis of classification. In our approach, sample data is used to define the set of pre-defined classes that identified each category of the problem formulator tool. To come up with such sample data in order to obtain an accurate set of pre-defined classes for all the categories was a challenging task.

The first version of our tool followed a very naïve approach to categorize the data. For inputs containing multiple words, we extracted the very first noun appearing in the sentence and then used WordNet to determine its class. However, this approach caused improper classification, which led us to follow a technique where we took the syntactic structure of the sentence into consideration as well.

Future Work

The results obtained using the above technique were encouraging. However, they weren't ideal. This was not only because of the minimalistic techniques employed for classification of issues and use scenarios, but also because the inputs provided by the user could be classified into one or more categories using discussed techniques. This could lead to a confusion as to which category does an input exactly belong. To address this challenge, we are working on an ensemble where each of the techniques will have certain weight assigned to it. One of the techniques in this ensemble would be finding the semantic similarity of the user inputs with the problem statement in combination with a normative Problem Map, which is defined as a standard design description for the given design problem. Including weights for different techniques in the ensemble will better the existing results by generating more accurate categories for a given problem fragment.

Conclusion

This paper reports the study of applying text classification approaches that consider syntactic structure and semantic meaning of the text to classify it in an engineering design field. We used a natural language processing approach to take advantage of the syntax and semantics of the inputs to be classified, and used pre-defined ontologies to perform our tasks. We described a classification technique where we used hypernyms from WordNet, relations from K-Parser, and modals extracted using Stanford Parser to define our pre-defined classes that identify each category. The inputs were first parsed using Stanford parser and keywords were extracted from them, which were later associated with one of the pre-defined classes, and classified to the respective category. We used an inter-rater agreement method, Cohen's kappa, to compare the tool's evaluation of the user inputs with two raters. Results showed that the agreement between the tool and each of the raters is fairly close. But, when compared to the agreement between the two raters, it was low. This method proved to be quite satisfactory for Artifacts, Behaviors, Functions, and Requirements. However, the accuracies for Use Scenarios and Issues were unsatisfactory. Overall, the results were encouraging and motivated us to continue to expand this model by incorporating more techniques of classification to increase the accuracy.

Acknowledgements This study is supported by the National Science Foundation, Grant Number 1002910. The opinions expressed in this paper are those of the authors and are not endorsed by the National Science Foundation.

References

Cheong H, Shu LH (2012) Automatic extraction of causally relate functions from natural-language text for biomimetic design. In: ASME 2012 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers, 2012

Clark P, Porter B, Boeing Phantom Works (2004) The knowledge machine 2.0: users manual. Department of Computer Science, University of Texas at Austin, Austin

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20:37–46

De Marneffe MC, Manning CD (2008) Stanford typed dependencies manual. URL http://nlp.stanford.edu/software/dependencies_manual.pdf

de Marneffe MC, MacCartney B, Manning CD (2006) Generating typed dependency parses from phrase structure parses. In: LREC 2006 de Rodríguez MB, Gômez-Hidalgo JM, Díaz-Agudo B (1997) Using WordNet to complement training information in text categorization. In: RANLP-97, pp 150–157

Dinar M, Danielescu A, MacLellan C, Shah J, Langley P (2015) Problem map: an ontological framework for a computational study of problem formulation in engineering design. J Comput Inf Sci Eng 15(c):1–10

Elberrichi Z, Rahmoun A, Bentaallah MA (2008) Using WordNet for text categorization. Int Arab J Inf Technol 5(1):16–24

Glier MW, McAdams DA, Linsey JS (2014) Exploring automated text classification to improve keyword corpus search results for bioinspired design. J Mech Des 136(11):111103

Gwet KL (2008) Variance estimation of nominal-scale inter-rater reliability with random selection of raters. Psychometrika 73(3):407–430

Klein D, Manning CD (2003) Accurate unlexicalized parsing. In: Proceedings of the 41st meeting of the association for computational linguistics, pp 423–430

Krippendorff K (1980) Content analysis: an introduction to its methodology. Sage Publications, Beverly Hills, CA

MacLellan CJ, Langley P, Shah JJ, Dinar M (2013) A Computational aid for problem formulation in early conceptual design. J Comput Inf Sci Eng 13(3):031005

Miller GA (1990) Nouns in WordNet: a lexical inheritance system. Int J Lexicogr 3(4):245–264

Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu

Scott S, Matwin S (1999) Feature engineering for text classification. In: ICML, vol 99

Sebastiani F (2002) Machine learning in automated text categorization. ACM Comput Surv (CSUR) 34(1):1–47

Arpit Sharma, Nguyen H. Vo, Somak Aditya, Chitta Baral (2015) Towards addressing the winograd schema challenge: building and using a semantic parser and a knowledge hunting module. In: Qiang Yang, Michael Wooldridge (eds) Proceedings of the 24th international conference on artificial intelligence (IJCAI'15). AAAI Press, pp 1319–1325

The Analysis and Presentation of Patents to Support Engineering Design

Gokula Vasantha, Jonathan Corney, Ross Maclachlan and Andrew Wodehouse

Abstract This paper explores the role of patents in engineering design, and how the extraction and presentation of patent data could be improved for designers. We propose the use of crowdsourcing as a means to post tasks online for a crowd of people to participate and complete. The issues of assessment, searching, clustering and knowledge transfer are evaluated with respect to the literature. Opportunities for potential crowd intervention are then discussed, before the presentation of two initial studies. These related to the categorization and interpretation of patents respectively using an online platform. The initial results establish basic crowd capabilities in understanding patent text and interpreting patent drawings. This has shown that reasonable results can be achieved if tasks of appropriate duration and complexity are set, and if test questions are incorporated to ensure a basic level of understanding exists in the workers.

Overview

Although there are millions of online patent records instantly available, their volume and presentation combine to make interpretation and assessment of their contents a laborious process. Designers need new tools to allow them to quickly and accurately understand patents relevant in the context of a new project. We propose the utilisation of crowdsourcing to cut through the patent jungle and deliver a concise summary of the relevant Intellectual Property (IP) and its applications in an area of interest. Key components in crowdsourcing workflows are repletion (i.e. multiple, parallel tasks to generate sets of answers), peer review, iteration, and the linkage of payment to quality assessments. These characteristics can be used to locate relevant patent records, summarize their contents and collaboratively construct infographics that show the relative strength of clustering around topics. In other words, our research focuses on the use of the crowd to provide the designer

G. Vasantha · J. Corney · R. Maclachlan · A. Wodehouse (✉)
University of Strathclyde, Glasgow, UK
e-mail: andrew.wodehouse@strath.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_12

with two specific areas of functionality: patent usage assessment and patent landscape visualisation. Specifically, we aim to generate a new, visual form of patent map or gallery that incorporates measures of patent commercialization activity and technical metadata through the crowdsourcing approach that can be utilized by engineering designers during conceptualization and embodiment design. This paper reports on initial foundation steps vital to achieve this aim by developing a crowd capable of solving the upcoming high-reasoning patent analysis tasks.

Patents in Engineering Design

Despite the significant costs involved, patents have become a dominant facet of innovation with over 50 million being recorded in the European IPO database alone. International corporate strategies have been built around IP portfolios (Brown 2009), and digital infrastructures have facilitated a significant industry in patent analytics and landscaping. The patent system's efficacy, and founding principles, are challenged by contemporary issues such as patent "thicketing", patent trolling, protracted high-profile legal disputes, and controversially monopolizing patents granted in emerging technology fields. While open innovation approaches, and patent law amendments address these, the system is a deeply entrenched and internationally recognized facet of business life.

Figure 1 uses Pugh's (1991) Total Design process model to illustate common engineering design activities and how patents could support the different modes of working. While there are a number of tools such as landscaping and TRIZ, patents remain an under-utilized resource at a practical level for engineering design particularly through the conceptualisation and detailed design phases (Kanda 2008). Researchers have employed a range of approaches to attempt to remedy this. Chan et al. (2011) show that providing designers with "far-field and less common" patents can have a positive effect on creative idea generation. Chang et al. (2013) have used keywords in conjunction with the established TRIZ inventive principles to provide "design-arounds" in the latter phases of conceptual design. And the preparation of patents has also been used as an educational approach to "directed conceptual design" (Lloveras 2011). To determine how patents can best be applied in these different design contexts, it is necessary to understand various approaches to patent analysis.

The Challenge of Analyzing Patents

As an enormous publicly available source of data, patents have attracted a great deal of attention from researchers attempting to identify patterns of innovation, technological trends, and creative thinking. While much of this work originates in economics, engineering design is well-placed to utilize novel design solutions in

Fig. 1 Use of patents at different stages of the engineering design process

patents. Indeed, although there are specialist patents such as pharmaceuticals and plants that serve niche industries, the majority relate to novel technologies that can be applied in a range of settings. For example, it is impossible to patent an umbrella as a stand-alone concept, but novel materials, opening mechanisms, handle designs and so on can be patented. While these may be intended for use in an umbrella, they could be applied in other areas. The challenges in extracting and understanding the different kinds of creative thinking bound up in formal patent documents make it problematic for use in practical engineering design settings. We have therefore undertaken a review of patent use in engineering design to identify opportunities to leverage a crowdsourcing approach.

Patent Assessment

Patent assessment can be grouped into two stages: pre- and post-grant. Before the patent is granted, novelty and usefulness are the two important parameters. Once granted, patents are evaluated based on the possibility of commercialization.

Pre-Grant

Many novelty assessment methods have been proposed in engineering design literature (Chakrabarti and Khadilkar 2003; Saunders 2002; Shah and Vargas-Hernandez 2003; Lopez-Mesa and Vidal 2006; Felk et al. 2011), but fewer for usefulness. Sarkar and Chakrabarti (2011) proposed a usefulness metric based on the product of: level of importance, rate of popularity of usage, rate of use, frequency of usage, and duration of use. However, there is a wide variety of other definitions such as “functional and operable” (Kang et al. 2007), “manufacturable” (Australian Law Reform Commission 2015), “achievable” (Nicol and Nielsen 2003), “specific, substantial… and credible” (United States Patent and Trade Mark Office 2013), and “capable of industrial application” (European Patent Organisation 1977). The USPTO usefulness assessments of “specific, substantial and credible” and “capable of industrial application” (Intellectual Property Office 2015) are accepted comprehensively. A patent should include a complete specification to fully describe the use of the invention and how it can be achieved (Australian Law Reform Commission 2015). However, whether this procedure is commonly applied in patent applications is questionable, with “novelty” and the demonstration of a meaningful inventive step forming the primary factor in pre-grant assessment (Gambardella et al. 2008). Indeed, the Nuffield Council (2002) observes that the guidelines to assess utility is limited, and would significantly increase the time and cost for examination. A crowdsourcing approach could support resourcing both novelty and usefulness assessment.

Post-Grant

There are no accepted and common measures for patents’ potential for successful commercial product development. Trappey et al. (2012) proposed a methodology to shorten the time required to determine and rank the quality of patents with respect to potential value. It involves extracting patent quality indicators, identifying the key impact factors using principal component analysis, and a back-propagation neural network (BPN) trained model to predict patent quality and forecast value. Other indicators used to value patents are related to investment, maintenance, and litigation (e.g. patent trade and assignment). The following parameters are used to assess countries’ progress on technology development: Pending duration (time from application to grant of patent), Originality index (measures the extent to which the patent is based on broad technological roots), and Technology dependence (measures the proportion of self-citations) (Chen et al. 2013). For engineering design, technical value should emerge from semantic analysis of patents. In-depth technical analysis of patent claims and descriptions could be explored through crowdsourcing. Issues related to algorithmic-based patent text analysis are discussed in the next section.

Patent Searches

The literature related to patent searches is predominately focused on computer-based retrieval techniques rather than understanding real needs of designers. This section reviews a range of empirical studies, and summarizes computer-based retrieval techniques applied to engineering design.

Empirical Studies

Heisig et al. (2010) reported UK survey results of knowledge and information requirements of managers and engineers in design and service. The "patent" category was noted only once from 129 respondents. They also noted that only one other empirical study (Rodgers and Clarkson 1998) reported on patent information needs. Patent infringement checks help to avoid reinventing the wheel, prevent costly litigation, and could potentially draw inspiration from existing designs. Koh (2013) discussed challenges involved in answering when and how IP infringement checks should be conducted during the engineering design process. He reiterated that searching patents before the design stages is difficult with unrefined design problem information and a broad search scope. Koh also highlighted that research is required to identify impact of patent infringement checks on design creativity, and to develop an affordable and effective means of retrieving relevant IP information during the engineering design process. Although search relevance could be increased for IP checks during the design stages, it is costly to frequently conduct thorough IP checks (Hitchcock 2005). The possibility of timely IP checks through low cost crowdsourcing approaches should therefore be explored.

Patent Search Techniques

Patent searches are typically classified into metadata searches (i.e. prolific inventors, cited patents, UPC classes etc.) and whole patent text searches. Keyword searches are commonly used to find relevant patent documents but are not always adequate due to inaccurate usage of terminologies, synonyms, polysemy, pronouns, multiple attributions, varying detail levels of patent descriptions, and homographs present in patents.

For full document searches, patent parsing is common. Structure (i.e. syntax) and dependency (i.e. word-to-word relations) parsing are the principal approaches in many search concepts, namely: two-level parser (Wang et al. 2015), function-behavior-state (FBS) information extraction (Fantoni et al. 2013), knowledge-based natural language analysis approach (Sheremetyeva 2003), concept-based patent search (Montecchi et al. 2013), transform queries to Subject–Action–Object-structures (Jeong et al. 2012), and conceptual graph extraction (Yang and Soo 2012). Wang et al. (2015) argued that the proposed Independent Claim Segment Dependency Syntax approach had improved efficiency (i.e. less computer memory and parsing time), as well as identifying some of the challenges in patent parsing due to peculiarities of claim syntax. Bonino et al. (2010) have summarized a range of different search types (Patentability, Validity, Infringement, Technology, Portfolio) but there is no accepted evaluation method to compare these. This offers scope to develop a hybrid approach blending crowdsourcing with computer algorithms to increase search efficiency.

Patent Clustering

Patent clustering is a process of grouping and graphically representing related patents to support patent valuation, technology relatedness, competitor analysis, patent strategy development, and technology management. Currently this is primarily done through classification systems (e.g. CPC) developed by Patent Offices. These provide a broader understanding of patent groups rather than in-depth patent differences and details required for engineering design. Kitamura et al. (2004) proposed use of patent maps based on functional decomposition for design review, indicating applicable ways to achieve a function, and preparing robust patent applications. Users describe function trees on a graphical user-interface; a time consuming procedure for designers. An alternative approach is required to extract functional information from patents. Fu et al. (2013) demonstrate that computationally-generated structure, compared to human experts, (i.e. using Latent Semantic Analysis (LSA) and hierarchical Bayesian algorithm) is sensible in clustering patents and organizing clusters (i.e. functional similarity). However, there is no optimum approach to patent clustering to support engineering design, and the usefulness of commercial patent landscaping software (e.g. AcclaimIP™, Patent iNSIGHT Pro™, ThemeScape™) is not explored adequately in the literature.

Technology Forecasting

Patent clustering has mostly been purposed for technology forecasting. Trappey et al. (2011) proposed a methodology clustering patent content using life cycle stages of: saturation; maturity, and; early growth to find a niche for RFID technology development in China. Other ontology-based patent clustering approaches are proposed for the strategic prediction of development trends and knowledge flows (Trappey et al. 2014; Bermudez-Edo et al. 2013).

Kim and Jun (2015) analyzed all Apple Inc. patents using the graphical causal inference method and the semiparametric Gaussian copula regression model to show technological trends, and relations between Apple's technologies. Whether an expert could generate useful knowledge (i.e. to find vacant technology areas) from the generated technology path diagram without greater detail or semantics of relationships is questionable.

Jin et al. (2015) used quality function deployment (QFD) matrices along with semi-automatic extraction and comparison of keywords from text to generate a technology-driven roadmap (TRM). They argue that TRMs could form a basis for identify profitable markets and promising product concepts. This comparative approach cannot be applied to all new technology, experts are needed to screen keywords, and presently the TRM is created manually. These are tasks that could be reformatted to assign to a crowd and efficiently integrated with a visualization tool.

Patent Knowledge Transfer

Knowledge transfer is the utilization of knowledge embedded in patents to solve problems and create new design solutions. Design-by-analogy and TRIZ are the most commonly used modes in engineering design, and are summarized in the following two sections.

Design-by-Analogy

Busby and Lloyd (1999) found that solution search activity provided more innovative influences than conservative ones. Design-by-analogy, serendipity, forced analogy, relational words and random input are some of the methods commonly used for solution search and transfer activities (Zusman and Zlotin 1999). Kurtoglu et al. (2009) evaluated a computational method which extracts product design knowledge at two levels of abstraction; functional and component, and creates procedural rules that depict the mapping between these. Vandevenne et al. (2015) proposed a scalable search for systematic biologically-inspired design, using elements extracted from patent and biological databases. Cascini and Zini (2008) used functional hierarchical trees to search for patent similarity. Similarly, Fu et al. (2014) presented the results of testing a method for extracting functional analogies based on functional vector space representation (Murphy et al. 2014) from patent databases to assist designers in systematically seeking and identifying analogies. The work extracted vocabulary of functions from a patent database, building on the hierarchical structure of a functional basis (Otto and Wood 2001). The results demonstrated significantly improved novelty of solutions generated, but no significant change in the total quantity of solutions generated.

TRIZ

TRIZ is a Russian acronym meaning the Theory of Inventive Problem Solving (TIPS). Altshuller developed TRIZ by studying over 1.5 million patents and noticing patterns in the evolution of technical systems (Altshuller et al. 1994). A set of universal principles for problem-solving were subsequently identified by observing recurring engineering conflicts and their solutions (Altshuller 1999). The reported research mostly focusses on building knowledge-based systems to support TRIZ processes, and attempting to apply TRIZ in different application domains (Lee et al. 2015; Lin and Luh 2009). The four main process steps (problem definition, problem classification and tool selection, solution generation and evaluation) need many support tools to apply generated principles effectively. Although the merits of TRIZ are widely cited, demerits can be less novelty when compared to the intuitive brainstorming method (Chulvi et al. 2012), and the difficulty finding 'out of the box' solutions (Reich et al. 2012).

Harnessing the Crowd

We have identified several areas where tasks could be undertaken, augmented or enhanced through crowdsourcing. In this section we explore the crowdsourcing concept and its application in the engineering design context. Crowdsourcing (aka micro-outsourcing) is an approach where smaller tasks are posted online for a crowd of people to participate and complete. The term crowdsourcing was coined by Jeff Howe in 2006 as 'the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call' (Howe 2006). These activities are executed by people via virtual tools who can access tasks, execute them, upload the results and receive various forms of payment using any web browser. This is a labour market open 24/7, with a diverse workforce available to perform tasks quickly and cheaply. The distributed network of human workers provide on-line, "black-box", reasoning capabilities that can exceed the capabilities of current AI technologies (i.e. genetic algorithms, neural-nets, case-based reasoning) in terms of flexibility and scope.

Although many automatic approaches are proposed for patent searching, human intervention is required for verification, validation, and providing judgements for the outcomes generated between search procedures. These tasks could be crowd-sourced supplementing automatic approaches to improve their accuracy and reducing time spent by designers in refining the search process. Crowdsourcing could also be utilized for retrieving informative graphical content from patents, where automatic approaches are difficult to apply. In terms of idea generation, proposed computational techniques for design-by-analogy proved that the platforms help designers generate novel solutions.

Applications of crowdsourcing in design research are developing slowly. Table 1 summarizes work on crowdsourcing in engineering design. These reported works could be well summarized within the three research directions proposed by Maher (2011): technology development, creative design processes, and evaluating creativity. However, the focus is predominately on the crowd evaluation process.

Crowd Capability

The approach of using crowds in analyzing patents depends on their basic abilities in understanding and processing patent information. Patents are rich in textual and graphical content, and the crowd should possess abilities to process both. To assess the abilities of crowd, crowdsourcing tasks should be framed in ways that are easy to understand and quickly undertaken (e.g. maximum 15 min). Considering requirements of patent understanding and crowdsourcing task design, two initial experiments were conducted within the Crowdflower (2015) platform. Crowdflower

Table 1 Summary of research work reporting the crowdsourcing approach in engineering design

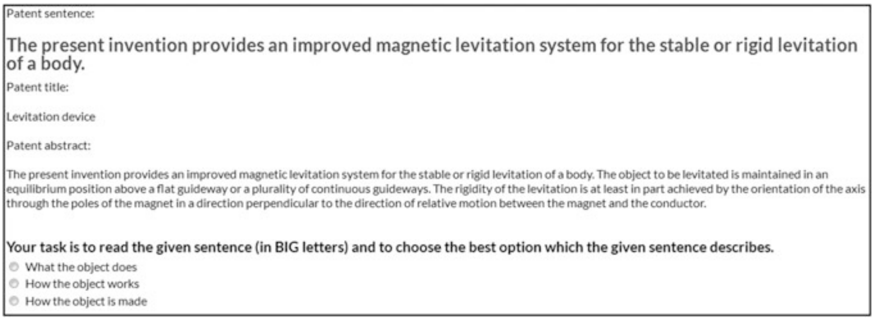
Authors	Aim/objectives	Design tasks	Important findings
Wu et al. (2015)	Propose 'Crowdsourced Design Evaluation Criteria' (cDEC) to support generation and evaluation of crowd-enabled design activities	Design living room layouts	Effective evaluation of design quality is a key component to leverage virtual workforce's creative activities, and vital to iterative optimization processes
Yu and Nickerson (2011)	Support crowd creativity through an iterative process of design, evaluation and combination	Chair design	Iterative process leads crowd to inherit and modify presented features enabling to generate creative product
Burnap et al. (2015)	Propose a simulation-based crowd consensus model to identify experts in crowd for design evaluation	Bracket strength evaluation	The model predicts experts only if the assumptions made are correct (e.g. only experts have consistent evaluations)
Maher et al. (2011)	Support collective design by understanding representation, communication and motivation	Google Image labeller, threadless	Proposed schema for evaluating crowdsourcing platform for representation, communication and motivation
Sun et al. (2014)	Integrate crowd's sketching processes via collaborative crowdsourcing design	Communal facilities for elderly people's recreation	Crowd rely heavily on the idea tree for inspiration, and best ideas appear around ends of the idea tree
Luther et al. (2015)	Help designers to receive design critiques from non-expert crowd workers	Poster designs	Aggregated crowd critique approaches expert critique, improved design process, and assists design changes
Grace et al. (2015)	A process model for crowdsourcing experience design for online communities	Citizen science project	Increases motivation of crowd and creativity of the design
Bao et al. (2011)	Compare evaluation methods to increase crowdsourcing system's effectiveness	Solutions to oil spill issue in the Mexico Gulf	Argued that evaluation methods should be assigned in relation to the distribution of quality present at each stage of crowdsourcing
Bayus (2013)	Characterizing individual's ideation efforts in crowd.	Dell's *IdeaStorm* community	Ideators struggle to repeat their success due to fixation
Poetz and Schreier (2012)	Compare novelty, customer benefit, and feasibility of ideas generated between experts and crowd users	Baby products	Crowd user ideas score higher in novelty and customer benefit, but lower in feasibility
Vattam and Goel (2011)	To catalogue and annotate research articles using the FBS-based approach	Biological sources	Created Biologue (social citation cataloguing system) to gather, organize, share, and annotate articles

has 30 million workers, with simple task design and monitoring interfaces. The first experiment reviews how well the crowd categorized written patent content, and the second interpretation of patent drawings.

Crowd Experiment 1: Patent Textual Information Categorization

Crowd workers predominantly undertake jobs such as text transcription, data cleaning, opinion surveying, and image recognition. To determine whether they can interpret more complex patent text, a data categorization task was designed. The task aimed to test classification of function, behavior and structure (FBS) information present in a patent abstract. The FBS structure was chosen for its relevance to design, comprehensiveness, and because it provided the crowd with a simple three part code. It was presented to the crowd as categories of 'What the object does', 'How the object works', and 'How the object is made'. Figure 2 shows the task posted on Crowdflower. The crowd were then asked to categorize a sentence taken from a patent abstract. To aid understanding, the title, full abstract and an example for each category was also provided. Seventeen tasks were posted, with twenty responses requested for each task. Three test questions, with rationalized solutions, were included to improve crowd responses.

76 workers from 29 countries participated. 340 judgments were received within 1 h 10 min. On average, each worker gave 4 responses (maximum: 12 responses and minimum: 2 responses). Figure 3 illustrates maximum agreement percentage for each of seventeen posted tasks ranging from 40 to 70% with an average of 55.29%. For 15 of 17 tasks, the highest agreement % related to the correct category. In 2 tasks, the sentence related to both F and B, so agreement was split between these. This initial test showed at least 55% of the participants could understand and choose the correct FBS category for the given sentence. Although 45% of the crowd lacked patent understanding, the maximum agreement % among the crowd for each question helped to identify the correct FBS categorization.


Patent sentence:

The present invention provides an improved magnetic levitation system for the stable or rigid levitation of a body.

Patent title:

Levitation device

Patent abstract:

The present invention provides an improved magnetic levitation system for the stable or rigid levitation of a body. The object to be levitated is maintained in an equilibrium position above a flat guideway or a plurality of continuous guideways. The rigidity of the levitation is at least in part achieved by the orientation of the axis through the poles of the magnet in a direction perpendicular to the direction of relative motion between the magnet and the conductor.

Your task is to read the given sentence (in BIG letters) and to choose the best option which the given sentence describes.

- What the object does
- How the object works
- How the object is made

Fig. 2 FBS patent information categorization task

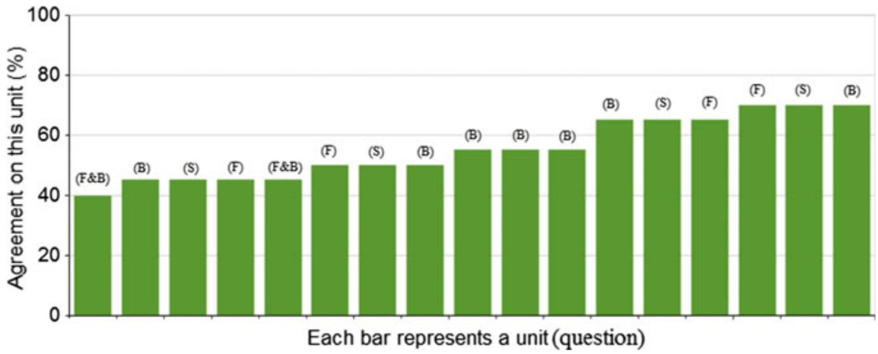


Fig. 3 Maximum agreement percentage for each of 17 posted questions (in *brackets* show the maximum chosen category in FBS)

Crowd Experiment 2: Understanding Patent Drawing Information

Patent drawings usually contain numerically annotated parts which are referred to in claims and invention descriptions. These annotations were used to test understanding of patent drawing information. Figure 4 illustrates the task posted, whereby a patent drawing was provided, and the crowd asked to describe a particular number in the drawing. The process was presented as follows: Download the patent document; Read the patent abstract; Search the number identified in the question; and Read and understand all paragraphs relating to the number, and write a description of it. The initial test concerned 10 numbers on 6 drawings from a single patent (US 9148077 B2).

Twenty responses for each number were requested and received for this task. One hundred and ninety five responses for this task were received within 2 h 40 min. The other five responses were received in 8 h 40 min. In total, 30 crowd workers participated from 17 countries. On average, each worker categorized 7 responses (maximum: 10 responses/worker and minimum: 1 response/worker). Analyzing the description for each part number revealed that only 24% of the

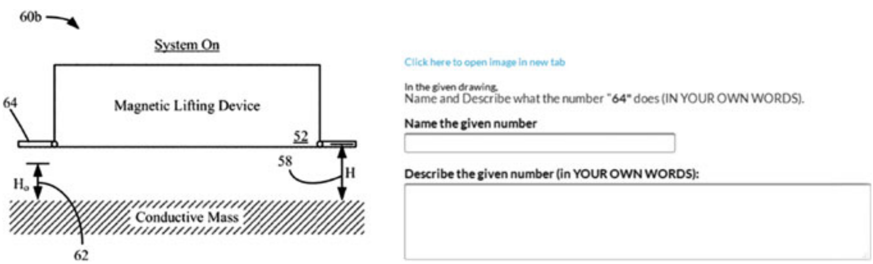


Fig. 4 Task for understanding patent drawing information

Table 2 Key variables observed between with and without test questions in understanding patent drawing information task

	Without test questions	With test questions
Total task completion time	8 h 40 min	19 h 22 min
Number of workers participated (from countries)	30 (17)	53 (27)
Number of workers failed in Quiz	N/A	25
Total number of responses	200	305
Average responses from a worker	7	6
Average percentage of acceptable responses	24%	66%
Accepted responses percentage range	15–50%	44–83%

answers received provided a correct description about the highlighted number. For example, the expected answer for number ‘64’ in Fig. 4 is “*used to give a stand-off height to the magnetic lifting device*”. It was surmised that a major reason for these relatively low percentages was due to not including test questions within this exercise. Although test questions are common practice in crowdsourcing, the challenge in this experiment was to add test questions allowing qualitative answers (i.e. textual description of a number in patent drawing). We were able to identify a mechanism by asking the crowd to first name the number and checking this before inviting them to add the description (shown in Fig. 4). By following this approach we could successfully add test questions for qualitative questions by requiring workers to provide the exact name (given in the patent document) of a given annotation number. Only workers who could answer at least 50% of test questions correctly could progress to the main task. This resulted in a much better percentage of acceptable responses, as set out in Table 2.

Figure 5 summarizes key variables observed with and without test questions in understanding patent drawing information task. The “test questions mode” significantly improved the percentage of acceptable responses. However, the total task completion time is almost double compared to “without test questions mode”. These initial tasks emphasize the importance of test questions to group a suitable crowd for patent related tasks.

The observations made from these initial two trails are: a large crowd of people from many countries is available on demand to undertake posted tasks; the completion time for the posted tasks is quicker (most responses received in less an hour), although it is dependent on the inclusion of test questions; judgement for categorizing patent textual information works best when aggregating all responses; clear task instructions, test questions, and payment for each task could play major role in getting acceptable responses; and the best workers among a crowd of people should be chosen through initial tests, then nurtured and developed further to potentially apply them to higher reasoning patent analyzing tasks.

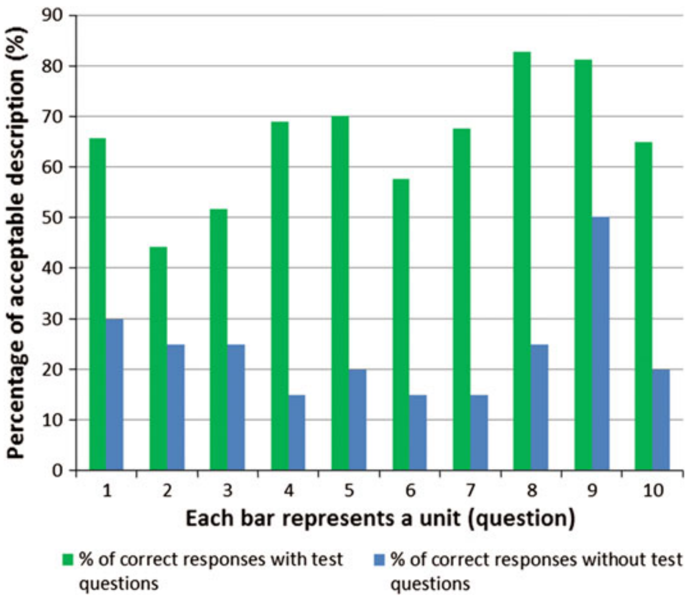


Fig. 5 Acceptable drawing number description percentage for each question

Discussion

This paper provides a broader view of patent applications in engineering design, and potential scope for using crowdsourcing. We have reported an important initial step through which further work could be developed to achieve the proposed aim: to generate a new, visual form of patent map that incorporates measures of patent commercialization activity and technical metadata through crowdsourcing. We hope this can ultimately be utilized by engineering designers during conceptualization and embodiment design.

These initial crowdsourcing experiments address two important points: (1) selecting and nurturing a crowd for patent analysis work, and (2) analyzing responses obtained from the crowd. We observed that about 55% of the participating crowd had good reasoning ability for FBS categorization. This is surprising considering that the crowd predominantly undertake simpler jobs such as text transcription and data cleaning. The successful crowd workers from this task could be selected and trained for further advanced patent analysis tasks. It is inadvisable to trust individual responses, and we found that a cumulative percentage response (in this case, 20 responses for each question) helped to identify the correct FBS categorization for each question. Therefore we intend to use cumulative aggregation of crowd responses for any patent related tasks to ascertain correct responses.

There is no acceptable/target percentage set for these crowdsourcing tasks, considering there are 30 million available crowd workers. The percentage (e.g. 66%

correct in patent drawing information task) signifies how many people from the participated crowd have shown interest and possesses minimum ability to be selected for further skills improvement. The observed correct answer percentage variation in both the initial tasks is expected, considering complexity of patent information and varying crowd skills level. For instance, some numbers in patent drawing are described in a straight-forward fashion, whereas others are hidden within the text. This makes patent text analysis complex for humans and also computer algorithms. However, more systematic study should be undertaken to understand patterns in this percentage variation. It is difficult to answer the trust level established with crowd from these initial experiments. However, with a crowd nurtured for patent analysis we are able to place increased trust in their responses.

Apart from obtaining acceptable responses quickly, we have identified a number of other limitations and possible future challenges through discussion with industrial partners. Issues include: confidentiality; trust in the crowd's responses; payment; ethics and legality. Approaches to address these issues are currently being explored in this research.

Conclusions and Future Work

This paper has illustrated the relevance of patents for engineering design. While the analysis of patent statistics such as classification, nationality and use of citations is well established, making robust interpretations based on these is problematic. Qualitative interpretations of the content and nature of patents is potentially more useful to engineering designers in terms of identifying active areas of design, examples of patents in use and understanding the characteristics of design problems. These cannot, however, be easily accomplished by computer algorithims and their magnitude is overwhelming for individuals. If crowdsourcing proves a cheap, scalable way of interpreting patents and applying appropriate taxonomic engineering information it could fundamentally alter the early phases of engineering design. The crowd can be harnessed to improve the data and information presented to designers during the key activities identified in scoping (opportunities), generation (inspiration), embodiment (context) and testing (checking).

Our initial crowdsourcing experimentation has been to establish basic crowd capabilities in understanding patent text and interpreting patent drawings. This has shown that reasonable results can be achieved if tasks are set appropriately, particularly in terms of duration and complexity, and if test questions are incorporated to ensure a basic level of understanding exists in the workers. Our planned future work will involve expanding on these initial experiments to design and test crowdsourcing workflows optimised to support working in the four design phases described. It is anticipated this will focus on analysing quality, classification and composition and content to present patent information in a way that is readily understandable and usable by engineering designers.

Acknowledgements This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant Number EP/N005880/1.

References

Altshuller G (1999) The innovation algorithm: Triz, systematic innovation and technical creativity. Technical Innovation Center, Worcester

Altshuller G, Altov H, Shulyak L (1994) Triz, the theory of inventive problem solving. Technical Innovation Center Inc, Worcester

Australian Law Reform Commission (2015) Patentability of genetic materials and technologies. http://www.alrc.gov.au/publications/6-patentability-genetic-materials-and-technologies/usefulness

Bao J, Sakamoto Y, Nickerson JV (2011) Evaluating design solutions using crowds. In: Proceedings of seventeenth Americas conference on information systems, available at SSRN: http://ssrn.com/abstract=2201651Detroit, USA: Howe School Research Paper No. 2013–2015

Bayus BL (2013) Crowdsourcing new product ideas over time: an analysis of the dell ideastorm community. Manag Sci 59(1):226–244

Bermudez-Edo M, Noguera M, Hurtado-Torres N, Hurtado MV, Garrido JL (2013) Analyzing a firm's international portfolio of technological knowledge: a declarative ontology-based owl approach for patent documents. Adv Eng Inform 27(3):358–365

Bonino D, Ciaramella A, Corno F (2010) Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. World Patent Inf 32(1):30–38

Brown T (2009) Change by design. HarperCollins, New York

Burnap A, Ren Y, Gerth R, Papazoglou G, Gonzalez R, Papalambros PY (2015) When crowdsourcing fails: a study of expertise on crowdsourced design evaluation. J Mech Des 137 (3):031101

Busby JA, Lloyd PA (1999) Influences on solution search processes in design organisations. Res Eng Des 11(3):158–171

Cascini G, Zini M (2008) Measuring patent similarity by comparing inventions functional trees. The international federation for information processing. Springer, Berlin, pp 31–42. doi:10.1007/978-0-387-09697-1_3

Chakrabarti A, Khadilkar P (2003) A measure for assessing product novelty. In: Proceedings of international conference on engineering design (ICED 03), Stockholm, Sweden

Chan J, Fu K, Schunn C, Cagan J, Wood K, Kotovsky K (2011) On the effective use of design-by-analogy: the influences of analogical distance and commonness of analogous designs on ideation performance. In: Proceedings of international conference on engineering design, ICED11, Copenhagen, Denmark

Chang HT, Chang CY, Yang YP (2013) Combining surveying patent information, reappearing problem and discovering breakthrough for design-around. In: Proceedings of 19th international conference on engineering design (ICED 13), Seoul, South Korea

Chen S-H, Huang M-H, Chen D-Z (2013) Exploring technology evolution and transition characteristics of leading countries: a case of fuel cell field. Adv Eng Inform 27(3):366–377

Chulvi V, González-Cruz M, Mulet E, Aguilar-Zambrano J (2012) Influence of the type of idea-generation method on the creativity of solutions. Res Eng Des. doi:10.1007/s00163-012-0134-0

Crowdflower (2015) https://make.crowdflower.com

European Patent Organisation (1977) European patent convention. http://www.epo.org/law-practice/legal-texts/html/epc/2013/e/index.html

Fantoni G, Apreda R, Dell'Orletta F, Monge M (2013) Automatic extraction of function–behaviour–state information from patents. Adv Eng Inform 27(3):317–334

Felk Y, Le Masson P, Weil B, Cogez P, Hatchuel A (2011) Designing patent portfolio for disruptive innovation—a new methodology based on c-k theory. In: Proceedings of 18th international conference on engineering design, Copenhagen, Denmark

Fu K, Chan J, Schunn C, Cagan J, Kotovsky K (2013) Expert representation of design repository space: a comparison to and validation of algorithmic output. Des Stud 34(6):729–762

Fu K, Murphy J, Yang M, Otto K, Jensen D, Wood K (2014) Design-by-analogy: experimental evaluation of a functional analogy search methodology for concept generation improvement. Res Eng Des 26(1):77–95

Gambardella A, Harhoff D, Verspagen B (2008) The value of european patents. Eur Manag Rev 5 (2):69–84

Grace K, Maher ML, Preece J, Yeh T, Stangle A, Boston C (2015) A process model for crowdsourcing design: a case study in citizen science. In: Gero SJ, Hanna S (eds) Design computing and cognition '1410.1007/978-3-319-14956-1_14. Springer, Cham, pp 245–262

Heisig P, Caldwell NHM, Grebici K, Clarkson PJ (2010) Exploring knowledge and information needs in engineering from the past and for the future–results from a survey. Des Stud 31 (5):499–532

Hitchcock D (2005) Patent searching made easy. Lulu Press, Raleigh

Howe J (2006) The rise of crowdsourcing. Wired magazine 14(6):1–4

Intellectual Property Office (2015) The patent guide: a handbook for analysing and interpreting patent data. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/463319/The_Patents_Guide_463312nd_edition.pdf

Jeong J-S, Park C, Yoo K-H (2012) Hand gesture user interface for transforming objects in 3d virtual space. In: Kim T-H, Adeli H, Grosky W, Pissinou N, Shih T, Rothwell E, Kang B-H, Shin S-J (eds) Multimedia, computer graphics and broadcasting, vol 262. Springer, Berlin, pp 172–178

Jin G, Jeong Y, Yoon B (2015) Technology-driven roadmaps for identifying new product/market opportunities: Use of text mining and quality function deployment. Adv Eng Inform 29 (1):126–138

Kanda AS (2008) An investigative study of patents from an engineering design perspective. Ph.D. Clemson University, Clemson, South Carolina

Kang I-S, Na S-H, Kim J, Lee J-H (2007) Cluster-based patent retrieval. Inf Process Manag 43 (5):1173–1182

Kim J-M, Jun S (2015) Graphical causal inference and copula regression model for apple keywords by text mining. Adv Eng Inform 29(4):918–929

Kitamura Y, Kashiwase M, Fuse M, Mizoguchi R (2004) Deployment of an ontological framework of functional design knowledge. Adv Eng Inform 18(2):115–127

Koh ECY (2013) Engineering design and intellectual property: where do they meet? Res Eng Des 24(4):325–329

Kurtoglu T, Campbell MI, Linsey JS (2009) An experimental study on the effects of a computational design tool on concept generation. Des Stud 30(6):676–703

Lee C-H, Wang Y-H, Trappey AJC (2015) Service design for intelligent parking based on theory of inventive problem solving and service blueprint. Adv Eng Inform 29(3):295–306

Lin C-C, Luh D-B (2009) A vision-oriented approach for innovative product design. Adv Eng Inform 23(2):191–200

Lloveras J (2011) A process of conceptual engineering design for new patentable products. In: Proceedings of 18th international conference on engineering design (ICED 11), Copenhagen, Denmark

Lopez-Mesa B, Vidal R (2006) Novelty metrics in engineering design experiments. In: Proceedings of 9th international design conference (DESIGN 2006), Dubronik, Croatia

Luther K, Tolentino J-L, Wu W, Pavel A, Bailey BP, Agrawala M, Hartmann B, Dow SP (2015) Structuring, aggregating, and evaluating crowdsourced design critique. In: Proceedings of 18th ACM conference on computer supported cooperative work and social computing. Vancouver, Canada, pp 473–485. doi:10.1145/2675133.2675283

Maher LM (2011) Design creativity research: from the individual to the crowd. In: Taura T, Nagai Y (eds) Design creativity 2010. Springer, London, pp 41–47. doi:10.1007/978-0-85729-224-7_7

Montecchi T, Russo D, Liu Y (2013) Searching in cooperative patent classification: comparison between keyword and concept-based search. Adv Eng Inform 27(3):335–345

Murphy J, Fu K, Otto K, Yang M, Jensen D, Wood K (2014) Function based design-by-analogy: a functional vector approach to analogical search. J Mech Des 136(10):101102

Nicol D, Nielsen JL (2003) Patents and medical biotechnology: an empirical analysis of issues facing the australian industry. SSRN Electron J. doi:10.2139/ssrn.2583508

Nuffield Council on Bioethics (2002) The ethics of patenting DNA: A discussion paper. http://nuffieldbioethics.org/wp-content/uploads/2014/2007/The-ethics-of-patenting-DNA-a-discussion-paper.pdf

Otto K, Wood K (2001) Product design techniques in reverse engineering and new product development. Prentice Hall, Upper Saddle River

Poetz MK, Schreier M (2012) The value of crowdsourcing: can users really compete with professionals in generating new product ideas? J Prod Innov Manag 29(2):245–256

Pugh S (1991) Total design. Addison-Wesley, Reading

Reich Y, Hatchuel A, Shai O, Subrahmanian E (2012) A theoretical analysis of creativity methods in engineering design: casting and improving asit within c–k theory. J Eng Des 23(2):137–158

Rodgers PA, Clarkson PJ (1998) An investigation and review of the knowledge needs of designers in smes. Des J 1(3):16–29

Sarkar P, Chakrabarti A (2011) Assessing design creativity. Des Stud 32(4):348–383

Saunders R (2002) Curious design agents and artificial creativity a synthetic approach to the study of creative behavior. Department of architectural and design science, Faculty of Architecture, Ph.D. University of Sydney, Sydney, Australia

Shah JJ, Vargas-Hernandez N (2003) Metrics for measuring ideation effectiveness. Des Stud 24 (2):111–134

Sheremetyeva S (2003) Natural language analysis of patent claims. In: Proceedings of ACL-2003 workshop on patent corpus processing Sapporo, Japan

Sun L, Xiang W, Chen S, Yang Z (2014) Collaborative sketching in crowdsourcing design: a new method for idea generation. Int J Technol Des Educ 25(3):409–427

Trappey CV, Wu H-Y, Taghaboni-Dutta F, Trappey AJC (2011) Using patent data for technology forecasting: China RFID patent analysis. Adv Eng Inform 25(1):53–64

Trappey AJC, Trappey CV, Wu C-Y, Lin C-W (2012) A patent quality analysis for innovative technology and product development. Adv Eng Inform 26(1):26–34

Trappey CV, Trappey AJC, Peng H-Y, Lin L-D, Wang T-M (2014) A knowledge centric methodology for dental implant technology assessment using ontology based patent analysis and clinical meta-analysis. Adv Eng Inform 28(2):153–165

United States Patent and Trade Mark Office (2013) Guidelines for examination of applications for compliance with the utility requirement. http://www.uspto.gov/web/offices/pac/mpep/s2107.html

Vandevenne D, Verhaegen P-A, Dewulf S, Duflou JR (2015) Seabird: Scalable search for systematic biologically inspired design. Artif Intell Eng Des Anal Manuf. doi:10.1017/s0890060415000177

Vattam SS, Goel AK (2011) Semantically annotating research articles for interdisciplinary design. Proceedings of the sixth international conference on Knowledge capture—K-CAP '11. doi:10.1145/1999676.1999707

Wang J, Lu WF, Loh HT (2015) A two-level parser for patent claim parsing. Adv Eng Inform 29 (3):431–439

Wu H, Corney J, Grant M (2015) An evaluation methodology for crowdsourced design. Adv Eng Inform 29(4):775–786

Yang S-Y, Soo V-W (2012) Extract conceptual graphs from plain texts in patent claims. Eng Appl Artif Intell 25(4):874–887

Yu L, Nickerson JV (2011) Cooks or cobblers? Crowd creativity through combination. In: Proceedings of the SIGCHI conference on human factors in computing systems, Vancouver, BC, Canada, pp 1393–1402. Doi:10.1145/1978942.1979147

Zusman A, Zlotin B (1999) Overview of creative methods. TRIZ J. http://www.triz-journal.com/overview-creative-methods/

Part IV
Design Grammars

Classifications of Shape Grammars

Sara Garcia

Abstract Since shape grammars were first described about forty five years ago, several types of grammars have emerged. The goal of this paper is to provide a framework that gathers together the main existing types of grammars. The categorization is preceded by a glossary of 19 terms related to shape grammars. Then, 44 types are placed into 13 chronologically-ordered categories. Each type is characterized with its name, a short description, the reference to the original paper, three examples of existing grammars of the type, and simple illustrative grammars. The types are organized using a classification guide in the form of a checklist, which is filled according to an existing shape grammar as an example.

Introduction

During the lifetime of shape grammars of about forty five years, several distinct types have been released. New features and extensions increased the original formalism launched in 1972, in order to sustain new needs of practical applications. The goal of this paper is to provide a framework that gathers and structures the main existing types, and to launch a standardized guide to classifications of shape grammars.

The benefits of the framework are threefold. Firstly, the comparison of different classifications can reveal some overlap of concepts, when different names are assigned to the same classifications. Secondly, it can orient future developments in the field; by systematizing what is explored and what is to be explored, and to reveal what areas are less developed than others. Furthermore, one can direct the development of a new grammar by considering the advantages and disadvantages of the different types of grammars, in order to fit the goals. Thirdly, it can provide a guide to shape grammar developers that want to classify and frame their specific shape grammar among the general theory, and to clarify the novelty of their work.

S. Garcia (✉)
Faculty of Architecture, University of Lisbon, Lisbon, Portugal
e-mail: sgarcia@fa.ulisboa.pt

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_13

Literature under the subject of shape grammars was reviewed, considering multiple sources. One considered, on one hand, individual works that launched specific new types, and, on the other hand, studies that gather several types, such as the ones of Knight (1999a, b) and of different states of the art [that are usually more focused on types around more specific contexts; e.g. comparative studies in grammars for product design (Ang et al. 2006; Barros 2015)]. The main contribution of this paper is to collect the types mentioned in these different sources. This paper gathers 44 types within 13 categories. Categories are displayed by chronological order.

The organization of the paper is as follows: firstly, the original theory of shape grammars is presented, based on a glossary of terms. Secondly, the categorization is arranged in several categories that contain different types of grammars. Each type is specified by its name, original reference (where the type is launched), a short description and notations (when applicable), one to three examples of existing grammars (that apply the type), and simple illustrative grammars. In the result section is provided a checklist with the named categories, filled accordingly to a categorization exercise of one existing shape grammar.

Shape Grammars

The shape grammar formalism was introduced by Stiny and Gips (1972), by transposing the alphabet of symbols used in the linguistic grammars (Chomsky 2002) to an alphabet and language of shapes. Shape grammars “provide a means for recursive specification of shapes” (Gips 1975). They became “the most important algorithmic approach to design” (Duarte 2001). A glossary with the main concepts that support the underlying theory is provided below.

(1) *Derivation:* the step-by-step generation of a design, from an initial shape (I) to a final design (D): $I = \rangle\ S_n = \rangle\ S_{n+1} = \rangle \ldots = \rangle$ D. Also referenced as *computation* (Stiny 1990).
(2) *Design*: a design (D) is the terminal shape (complete design) that belongs to the language defined by a grammar. Notice that a design can also be referred as an intermediate state (partial design) that do not belong to the language (Knight 1999a).
(3) *Embedding*: a line l_1 is embedded in another line l_2 if each of the endpoints of l_1 is either an endpoint of l_2 or is between the endpoints of l_2.
(4) *Emergence*: is the ability to recognize and use *emergent shapes* (E) – shapes that are not predefined in a grammar but emerge during the computation, through the detection of sub-shapes and the relation of predefined shapes (Knight 2003).
(5) *Empty shape*: is a shape that does not contain maximal elements, denoted by $S_{\varnothing}$.
(6) *Initial shape*: the starting shape (I) under which the rules are applied (can be an empty shape).

(7) *Label*: a label (L) is a symbol associated with shapes. Originally defined to control the rule application (transformations under which rules apply), may also represent other aspects of shapes. A labeled point is denoted by p: A (where p is the point and A is the symbol). A labeled shape θ contains a shape and a set of labeled points: θ = <S, p>. A non-labeled shape is denoted by <S, Ø>, a labeled empty shape is <$S_{Ø}$, P>, and a non-labeled empty shape is <$S_{Ø,}$ Ø>.

(8) *Language:* the set of all designs (D) generated by the grammar; the design space. Each grammar defines one language of designs (Stiny 1980a).

(9) *Maximal line*: one element of the smallest set of lines that specify a shape. A broader definition considers *maximal elements* (Stiny 1992).

(10) *Operations*: three Boolean operations can be made between two shapes S_1 and S_2. Sum – originally named union ($S_1 + S_2$), has all of the basic elements of S_1 and S_2 and nothing else; product – originally named intersection ($S_1 \cdot S_2$), has all of the basic elements belonging to both S_1 and S_2 and nothing else; and difference ($S_1 - S_2$) has all of the basic elements that belong to S_1, but not to S_2 and nothing else.

(11) *Rule*: a rule (R) is from the type A → B, where A and B are both shapes, being A the shape of the left-hand side of the rule (lhs) and B the shape of the right-hand side of the rule (rhs).

(12) *Rule application*: the rule application is a 2-step process (Stiny 1990). Consider the example in Fig. 1, giving one rule (R) and one shape in the n derivation step (S_n) (Duarte 2001):

a. A rule R in the form $S_{lhs} \rightarrow S_{rhs}$ may be applied to a shape S_n if there is a *sub-shape* embedded in S_n that is equal to the shape S_{lhs} under a *transformation* τ. There is a matching when $\tau\ (S_{lhs}) \leq S_n$.

b. In that case, the transformation applied in S_{lhs}, and denoted by $\tau\ (S_{lhs})$, is applied to S_{rhs} and is denoted by $\tau\ (S_{rhs})$. The computation of the rule is made by *subtracting* the sub-shape $\tau\ (S_{lhs})$ to the shape S_n, and *adding* the shape τ

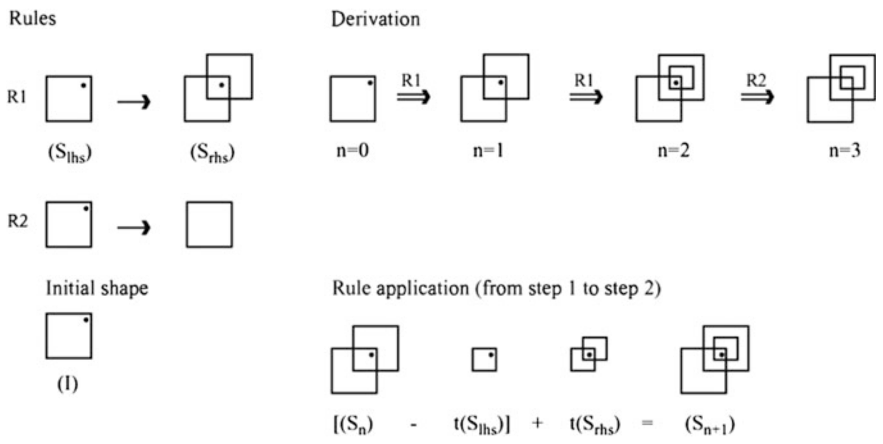


Fig. 1 Example of a shape grammar and computation of a rule

(S_{rhs}). The shape that results from that operation is S_{n+1}, where $S_{n+1} = [S_n - \tau (S_{lhs})] + \tau (S_{rhs})$.

(13) *Shape*: “finite but possibly empty set of maximal lines” (Stiny 1990). This first definition was later extended, in order to incorporate other form elements: “finite but possible empty set of basic elements that are maximal with respect to one another” (Stiny 1992). The shape may be an *empty shape*.

(14) *Shape grammars*: are originally defined (Stiny 1980b) by four components: shapes (S), labels (L), rules (R), and initial shape (I). A shape grammar is a system that generates shapes (S) through the successive application of rules (R), from the type if/then, to an initial shape (I), until a design (D) is reached.

(15) *Spatial relations:* combinations of shapes in the vocabulary (Stiny 1980a);

(16) *Sub-shape*: S_1 is a sub-shape of S_2 ($S_1 \leq S_2$) whenever every maximal line of S_1 is *embedded* in some maximal line of S_2. Every non-empty shape (and not made up of just points) has an infinite number of sub-shapes.

(17) *Termination*: The generation process is terminated when no rule in the grammar can be applied or until certain criteria is met (Knight 1999a).

(18) *Transformations*: a transformation τ on a shape S is denoted by τ(S). The similarity transformations on a shape can be one of the four Euclidean transformations – translation, rotation, reflection and scale – or the composition of at least two of them.

(19) *Vocabulary:* a limited set of different shapes, which are the basic building elements of the designs in set grammars (Stiny 1980a).

Figure 1 shows an example of a shape grammar with two rules (R1 and R2) and one initial shape. The displayed derivation has three steps (notice that the derivation could proceed indefinitely until the rule R2 is applied). One *emergent shape* appears in step 1 when a smaller square emerge from the overlapping of two squares.

Classifications of Shape Grammars

In this section various classifications of shape grammars are presented, grouped in several categories. Along with shape grammars, these classifications include other kinds of grammars that define languages of shapes. Spatial grammars (Krishnamurti and Stouffs 1993) include shape grammars, set grammars, string grammars, and graph grammars.

Shape Elements

The basic geometric elements (*i*) of the shape are points, lines, planes and solids (Stiny 1992). Originally, shape grammars were addressed to rectilinear shapes, such as straight lines and planes. Some latter extensions focus on curvilinear shapes, in order to include curves and non-planar surfaces. Curved grammars require a reframing of the rectilinear theory under particular issues, such as embedding and similarity proprieties, algebra formulation, and shape analytic description (Jowers and Earl 2015). Notice that the shape grammar formalism considers the possibility of computing with *empty shapes*.

1. *Points:* shapes are defined by points.
2. *Lines (rectilinear)*: shapes are defined by straight lines. Lines were originally defined as a set of two distinct endpoints {p1, p2} (Stiny 1980b).
3. *Lines (curvilinear):* shapes are defined by non-rectilinear lines. The first appearance of curved shapes in shape grammars is the one of the Hepplewhite grammar (Knight 1980); however curves were only used to replace straight lines in the last derivation stage. Latter, curves were applied into shape rules. Different types of planar curves, as circular arcs and Bézier curves can be found in different works. The use of non-planar curves in 3D space needs further development.

E.g.: Bottles (Chau et al. 2004) and vehicles (Orsborn et al. 2006) (circular arcs); Buick vehicles (McCormack and Cagan 2004), tableware (Castro e Costa and Duarte 2013) and Celtic knotwork patterns (Jowers and Earl 2011) (quadratic Bézier curves).

4. *Planes (rectilinear)*: shapes are defined by planes, which correspond to flat surfaces.
5. *Surfaces (curvilinear)*: shapes are defined by non-planar surfaces.

E.g.: Tableware (Castro e Costa and Duarte 2013) (revolved surface).

6. *Solids (rectilinear)*: shapes are defined by solids. Currently, shape grammars are restricted to rectilinear solids; research on curved solids needs further developments.

Dimensional Space

The shape basic elements are manipulated in a Euclidean dimensional space (*j*). Usually shapes are combined and manipulated in two or three dimensions.

1. *Zero-dimensional Grammar*: elements are manipulated in a zero-dimensional space (0D).
2. *One-dimensional Grammar:* elements are manipulated in a one-dimensional space (1D). For example, symbolic grammars transform strings in one dimension. Symbols cannot be divided, unlike shapes.

E.g.: Chomsky (2002).

3. *Two-dimensional Grammar:* elements are manipulated in a bi-dimensional space (2D).

E.g.: Ice-ray windows (Stiny 1977), Mughul gardens (Stiny and Mitchell 1980), Hepplewhite chair-back (Knight 1980).

4. *Three-dimensional Grammar*: elements are manipulated in a three-dimensional space (3D), where special relations are explored.

E.g.: Kindergarten (Stiny 1980a), Prairie houses (Koning and Eizenberg 1981), Queen Anne houses (Flemming 1987).

Different types of views can be combined in the same grammar. For example, the grammar of Malagueira houses (Duarte 2001) combines both 2D (for plans and elevations) and 3D spaces (for axonometrics).

Shape Qualities

Shape grammars originally considered the spatial aspect of form and labeled shapes. Forthcoming extensions deal weighted shapes and colored shapes. These qualities are associated to the initial shape and shape rules and, consequently, to the generated designs. Shape grammars thus allow for the combination of several descriptions in one design, as such shape may be associated with size, color, and texture.

1. *Labeled grammar* (Stiny 1980b): Labeled shapes are part of the original theory, as described above. There can be labeled points, labeled lines, labeled planes and labeled solids. Still, one can find illustrative examples of non-labeled grammars (Stiny 1980a).

E.g.: Ice-ray windows (Stiny 1977), Mughul gardens (Stiny and Mitchell 1980), Hepplewhite chair-back (Knight 1980).

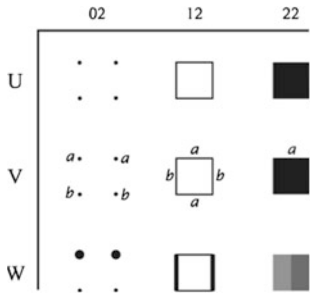
2. *Color grammar* (Knight 1989a): color grammars extend shape grammars by adding color as a qualitative aspect of the shape. Along with shapes, these grammars generate color fields (a region in 2D or 3D space filled with one or more colors) and define what happens when different color fields overlap. Colors may also be used as labels, or represent other qualities such as texture or materials.

E.g.: De Stijl paintings (Knight 1989b).

3. *Weighted grammar* (Stiny 1992): Weight is a quality of shape that can be represented in weighted points (by different radius), weighted lines (by different thickness), and weighted planes and solids (by different toned or shaded areas). Weights can be used merely to distinguish formal attributes (graphic expression), to represent physical proprieties (as mass, materials) or intentional proprieties (as function). Notice that a color grammar is a special case of weighted grammars (Duarte 2001).

E.g.: Malagueira houses (Duarte 2001), housing rehabilitation (Eloy and Duarte 2012), Thonet chairs (Barros 2015).

Fig. 2 Examples of shapes in various algebras of design in 2D space



The categories described above (shape elements, dimensional space, and shape qualities) are summarized in different algebras, described by Stiny (1992) as algebras of shapes, algebras of labels and algebras of weights (Fig. 2). Each algebra is represented in the dimension i (basic elements), and j (the space where basic elements are manipulated). Weights and labels can be manipulated under the same operations as shapes.

1. Algebras: U_{ij} (shapes); V_{ij} (labels); W_{ij} (weights).
2. Shape (basic) elements (i): i = 0 (points); i = 1 (lines); i = 2 (planes); i = 3 (solids).
3. Dimensional space (j): j = 0 (0D); j = 1 (1D); j = 2 (2D); j = 3 (3D).

Notice that $0 \leq i$; $j \leq 3$; and $i \leq j$. Algebras can be combined (in Cartesian products) in order to produce compound shapes.

Shape Properties

Extensions on the shapes of original shape grammars consider parametric shapes (*parametric grammar*) and sets of shapes (*set grammar*).

1. *Parametric Grammar* (Stiny 1980b): Unlike shape grammars that work with fixed single shapes, parametric grammars work with families of shapes, with equal topologies but different parameters. A parametric grammar uses shape schemas and rule schemas (Stiny 1990). A parameterized shape schema, denoted by S(x) is a general schema that describes a family of shapes, controlled by a set of variables x. A parameterized shape allows for the coordinates of the endpoints to be variable (Fig. 3). A particular shape is determined by assigning values to the variables through the g function, and is denoted by g(S(x)). A rule schema contains two parameterizes shapes $A(x) \rightarrow B(x)$; and when the g function is applied one obtain $g(A(x)) \rightarrow g(B(x))$. The formula for rule application is $S' = [S - \tau\ (g(S_{lhs}))] + \tau\ (g(S_{rhs}))$. Parametric grammars can be considered a generalization of the four similarity transformations originally described, plus the distortion one.

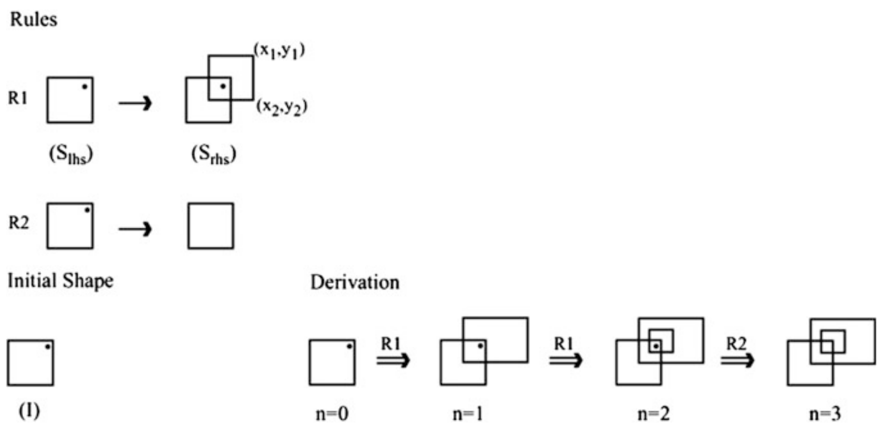


Fig. 3 Example of a parametric shape grammar

E.g.: Mughul gardens (Stiny and Mitchell 1980), Hepplewhite chair-back designs (Knight 1980), coffeemakers (Agarwal and Cagan 1998).

2. *Set Grammar* (Stiny 1982): Unlike shape grammars that work with shapes, set grammars work with finite sets of shapes (using set operations such as union, intersection, and difference). The standard formalism of shape grammars allows *emergence* proprieties. The variant called *set grammar* does not recognize emergent shapes. Shapes are thus treated as symbolic objects, that cannot be decomposed into sub-shapes and thus no spatial ambiguity is allowed.

E.g.: Malagueira houses (Duarte 2001), Thonet chairs (Barros 2015), motorcycles (Pugliese and Cagan 2002).

Rule Ordering

This category distinguishes between deterministic and nondeterministic shape grammars. Nondeterminism occurs when "multiple choices can be made in each stage" of the grammar (Stiny 1980a); while determinism happens when only one derivation is possible. This category is further described by Knight (1999a). She discourses on questions of decidability, considering restrictions in rule ordering: "the more restricted a grammar is, the less powerful it is but the more is known about how it works and what it generates" (Knight 1999a). The restriction level is a trade-off between freedom and control. The less restricted the grammar is, greater is the freedom, however, smaller is the control. Usually, the less restricted ones are more applicable to the arts domain.

1. *Deterministic Grammar*: there is no choice when executing rules; therefore there is only one possible derivation (Fig. 4). "Only one design or no design may be

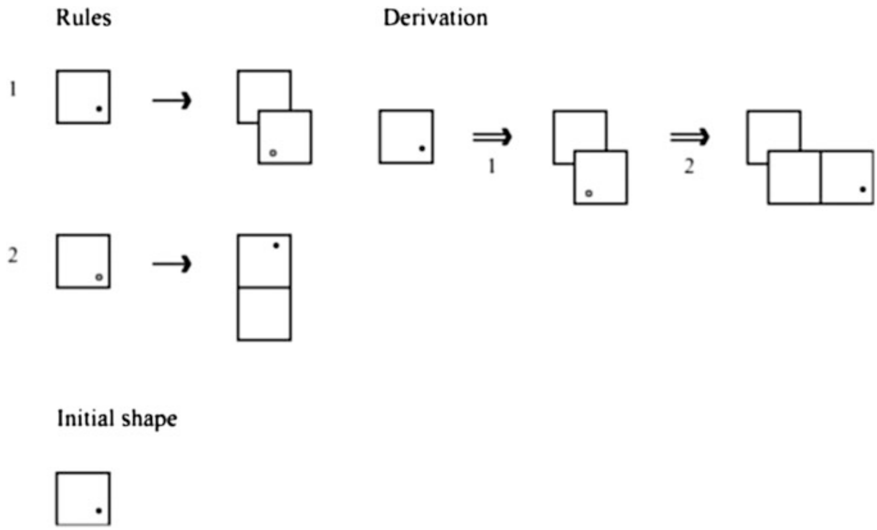


Fig. 4 Example of a deterministic grammar

generated from the previously generated design" (Knight 1999a). The restrictions can be imposed by labels, indexed rules, and so forth. Majorly, the examples of shape grammars are not from the deterministic type.

2. *Nondeterministic Grammar*: "allows multiple designs to be generated from a previous design in some step of the derivation" (Knight 1999a). There are three types of nondeterminism: the choice can be motivated on a selection of what rule to apply (type 1), where to apply the rule (type 2), and what transformations or parameters to apply (type 3). The application of each type requires certain conditions: there are at least two rules with identical lhs (type 1), there are at least two identical sub-shapes in a design (type 2), or the lhs of a rule has more than one type of symmetry (type 3). Combinations of the three types are also possible. Figure 5 illustrates an example of a grammar with several possible derivations, represented by a state-action tree of the grammar (from one starting shape to n possible ending shapes).

E.g.: Malagueira houses (Duarte 2001), housing rehabilitation (Eloy and Duarte 2012), Thonet chairs (Barros 2015).

Rule Format

Stiny (1980a) distinguishes between two main types of rules—additive and subtractive, and Knight added the combination type (Knight 1989b) (Fig. 6). Rules use the shape *operations* described above. Consider S_1 as a sub-shape of $(S_1 + S_2)$:

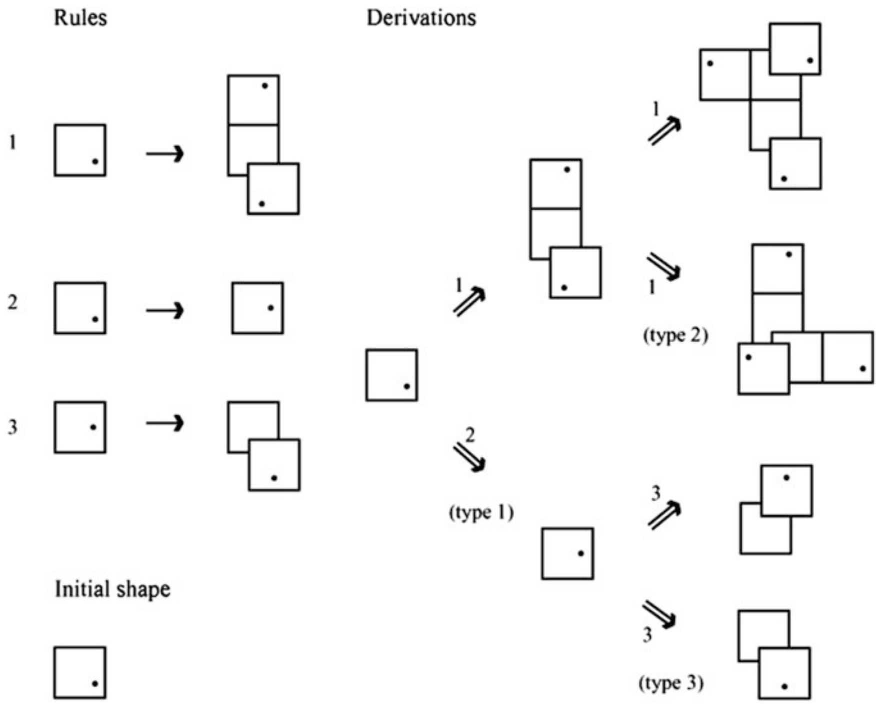
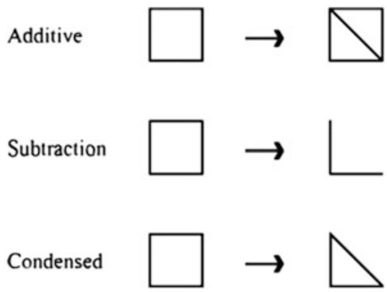


Fig. 5 Example of a nondeterministic grammar

Fig. 6 Examples of rule formats



1. *Additive Rule*: the rule adds a shape to a design: $S_1 \rightarrow (S_1 + S_2)$.
2. *Subtractive Rule*: the rule erases a shape to a design: $(S_1 + S_2) \rightarrow S_1$.
3. *Condensed Rule*: the rule combines the two previous types, by erasing and adding a shape to a design: $S_1 \rightarrow S_2$.

Design Strategy

This category is summarized by Knight (1999b) and addresses the overall design process. It provides both top-down (the case of subdivision and grid types) and bottom-up (the case of additive type) approaches. Notice that all of these types use additive rules (Fig. 7).

1. *Grid Grammar*: spaces are defined by a fixed, underlying grid (behaving as a guide), and details are added afterwards.

E.g.: Palladian villas (Stiny and Mitchell 1978), Japanese tearooms (Knight 1981), Yingzao Fashi (Li 2001).

2. *Subdivision Grammar*: used when the designs share similar boundaries. Starting by one fixed boundary, the rules of the grammar continuously subdivide the area.

E.g.: Ice-ray windows (Stiny 1977), Hepplewhite chair-back (Knight 1980), Malagueira houses (Duarte 2001).

3. *Additive Grammar*: used when designs have different boundaries. Starting by the core of the structure, the rules successively add parts. As such, the boundary varies.

E.g.: Prairie houses (Koning and Eizenberg 1981), Queen Anne houses (Flemming 1987), coffeemakers (Agarwal and Cagan 1998).

Grammar Purpose

Grammars may follow different goals and methods. The analytic grammar defines languages of designs from a given corpora of existing designs, while the synthetic

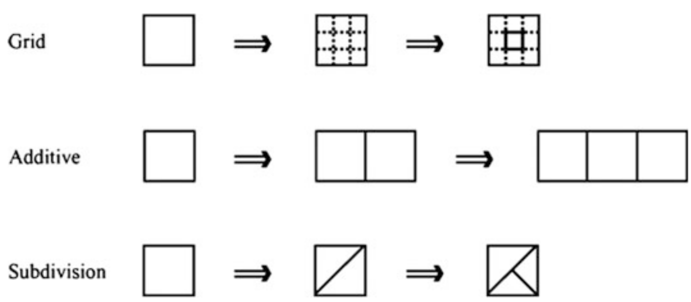


Fig. 7 Examples of derivations applying different design strategies

(or constructive) grammar defines the rules from a given vocabulary and spatial relations (Stiny 1980a). The transformation grammar defines new grammars from given grammars. These three approaches are summarized in Knight (1999b). A fourth approach is the one of generic grammars.

1. *Synthetic Grammar* (Stiny 1980a): also called *Original Grammars*: Although the original paper on shape grammars (Stiny and Gips 1972) was from this type, very few applications can be found in the following years. Synthetic grammars create languages 'from scratch', following a constructive method. They have the capacities to externalise the design process and to produce new original designs.

   E.g.: Kindergarden (Stiny 1980a), color grammars (Knight 1989a).

2. *Analytic Grammar* (Stiny and Mitchell 1978): The works on shape grammars have been focusing more on this type than on the previous one. The goal of this type of grammar is to capture historical styles and generate new designs within the style. It creates languages from a corpus of designs. A style can be characterized by a person (Knight 1980), a place (Stiny 1977), a period (Flemming 1987), a brand identity (McCormack and Cagan 2004), or a set of design standards (Li 2001). The criteria for analytic grammars (Stiny and Mitchell 1978) are: (1) to reproduce the style of the corpus, (2) to verify if a given design belongs to the style, and (3) to generate new designs within the style.

E.g.: Ice-ray windows (Stiny 1977), Palladian villas (Stiny and Mitchell 1978), Mughul Gardens (Stiny and Mitchell 1980).

3. *Transformation Grammar* (Knight 1989b): This type uses both analytic and synthetic approaches. Starting by an analytic approach, transformations are applied and synthetic results are obtained. Grammars are transformed into other grammars through rule transformation (addition, deletion, and change), or designs are transformed into other designs through transformation rules. It provides the description of different styles and their relations (by the analysis of similarities and differences between grammars) and/or the transformation of designs.

E.g.: De Stijl paintings (Knight 1989b), Yingzao Fashi (Li 2001), housing rehabilitation (Eloy and Duarte 2012).

4. *Generic Grammar* (Beirão et al. 2011): generic grammars are general-purpose grammars that can be customized into specific-purpose (synthetic) grammars, by the restriction of rules and parameters. Generic grammars are based on ontology classifications that reproduce context-independent design types, and abstract recurrent design patterns. Although not exploring the full potential of the generic types, some grammars create 'more general' designs within a product family [e.g. motorcycles (Pugliese and Cagan 2002)].

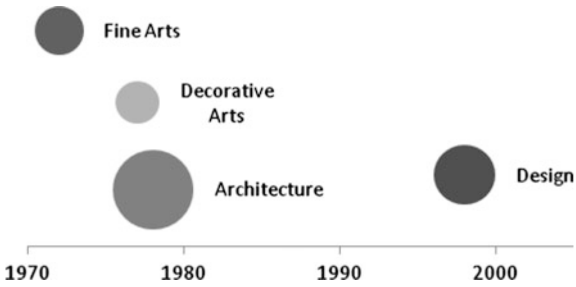


Fig. 8 Bubble chart of the application domain (based in Chau et al. 2004)

E.g.: Urban design (Beirão et al. 2011), social housing (Mendes et al. 2013), tableware (Castro e Costa and Duarte 2013).

Application Domain

Shape grammars have been applied in art and design disciplines. Chau et al. (2004) classified the grammars according to four areas of application: paintings, decorative arts, architectural plans, and engineering designs. The earlier applications considered arts disciplines, and the later ones approached design domains. The most representative area is architecture (Fig. 8). Notice that different disciplines require different abstraction levels of detail and scale. It is suggested a more general classification following five disciplines of art and design: fine arts, decorative arts, architecture, design, and urban design. The domains can consider successive levels of subdivision [e.g. furniture in product design (Barros 2015)].

1. *Fine Arts*: include the sub-domains of painting and sculpture.

E.g.: De Stijl paintings (Knight 1989b), color grammars (Knight 1989a), Celtic knotwork patterns (Jowers and Earl 2011).

2. *Decorative Arts*

E.g.: Ice-ray windows (Stiny 1977), Hepplewhite chair-back (Knight 1980), Thonet chairs (Barros 2015).

3. *Architecture*: include the sub-domain of landscape architecture.

E.g.: Palladian villas (Stiny and Mitchell 1978), Prairie houses (Koning and Eizenberg 1981), Mughul gardens (Stiny and Mitchell 1980).

4. *Design:* include engineering, product, fashion, and interior design.

E.g.: Roof trusses (Shea and Cagan 1999), coffeemakers (Agarwal and Cagan 1998), tableware (Castro e Costa and Duarte 2013).

5. *Urban Design*

E.g.: Urban design (Beirão et al. 2011), social housing (Mendes et al. 2013).

Additional Techniques

This category gathers extensions of shape grammars that combine other activities beyond analysis and generation, such as evaluation, fabrication, or integrations of techniques. For example, performance-based grammars are concerned with design evaluation, by the use of analysis and optimization techniques. There can be an exchange of information between geometry and performance.

1. *Generation Techniques*: combinations of shape grammars with other generation techniques, beyond the parametric one already mention above.

E.g.: Office chairs (Hsiao and Chen 1997) (morphological analysis).

2. *Evaluation Techniques*: provide performance data during or after the generation.

E.g.: Coffeemakers (Agarwal et al. 1999) (cost evaluation), office chairs (Hsiao and Chen 1997) (semantic evaluation), Palladian villas (Stiny and Gips 1978) (aesthetic evaluation).

3. *Optimization Techniques*: combine both generation and evaluation in an automated generative process. Optimization algorithms search for the best solution in a large solution space, considering single or multi-objective functions. Shape Annealing (Cagan and Mitchell 1993) is a combination of Shape Grammars and Simulated Annealing (a stochastic optimization technique). The algorithm searches for good solutions in a reasonable period of time (sacrificing the optimal solution), by determining "whether a randomly selected shape rule should be applied at a given design state" (Cagan and Mitchell 1993). Evolutionary algorithms (such as genetic algorithms) combine both optimization and evolution capabilities. While genetic algorithms evaluate several solutions at the same time, exploring crossover and mutation in the population, shape annealing considers just one individual in the population (one solution at a time is being evaluated) and, therefore, only mutation occurs.

E.g.: Roof trusses (Shea and Cagan 1999) (shape annealing), Digital cameras (Lee and Tang 2004) and Coca-cola bottles (Ang et al. 2006) (genetic algorithms).

4. *Fabrication Techniques*: shape grammars oriented to provide an automation between generation and digital fabrication (for example by providing constructible components for CNC machines).

E.g.: Wood frame house (Sass 2005), CNC fabrication (Ertelt and Shea 2008).

Compound Representations

Compound grammars combine multiple representations. Different types of descriptions may be associated: multiple views (plans, sections, elevations, etc.), properties (function, material, construction, aesthetics, etc.), diagrams and graphs (representing hierarchy of parts, etc.) (Stiny 1990).

1. *Abstract Grammar* (Knight 1999b): shapes do not have any associated meanings; they are only sustained on geometry/syntax. Early work on shape grammars focused on this type. The implementation becomes increasingly difficult as the detection of (emergent) sub-shapes becomes infinite (Pauwels et al. 2015).

E.g.: Ice-ray windows (Stiny 1977), Kindergarten (Stiny 1980a), De Stijl paintings (Knight 1989b).

2. *Description Grammar* (Stiny 1981): shapes have associated meanings/semantics, including functions (such as walls or windows), shape proprieties (such as area), aesthetic issues, and so forth. For each shape rule there is one or more corresponding description rules; and the designs generated by the grammar have an associated (text or numerical) description. By assigning meaning to shape, the interpretation of designs becomes more restricted (Knight 1999b).

E.g.: Queen Anne houses (Flemming 1987), Palladian villas (Stiny and Mitchell 1978), Prairie houses (Koning and Eizenberg 1981).

3. *Discursive Grammar* (Duarte 2001): is a compound parallel grammar that combines shape grammars, description grammars, and a set of heuristics. Its goal is to generate consistent adequate designs, as in design grammars (Pauwels et al. 2015).

E.g.: Malagueira houses (Duarte 2001), housing rehabilitation (Eloy and Duarte 2012), urban design (Beirão et al. 2011).

4. *Graph Grammar*: shapes are combined with graph representations (Krishnamurti and Stouffs 1993). Graphs can in turn represent shapes; notice that a line can be represented as a graph (with one edge and two vertices).

E.g.: Housing rehabilitation (Eloy and Duarte 2012).

Grammar Structure

Grammars can be composed by more than one grammar. A compound grammar can be parallel (occurring at the same time) or sequential (occurring one after another).

1. *Parallel Grammar* (Stiny 1992): two or more grammars are combined to operate simultaneously, and thus allow different design representations. An example of a parallel grammar is the discursive grammar (Duarte 2001).

E.g.: Malagueira houses (Duarte 2001), Yingzao Fashi (Li 2001), housing rehabilitation (Eloy and Duarte 2012).

2. *Sequential Grammar* (Knight 1999a): the original definition of this type states that rules are applied in a predetermined sequence. This definition can have further generalizations, as some grammars have different generation phases that must occur in sequence; or cases where different grammars are applied in sequence.

E.g.: Queen Anne (Flemming 1987) (two grammars), tableware (Castro e Costa and Duarte 2013) and De Stijl paintings (Knight 1989b) (three stages).

Implementation

This category classifies grammars according to the environment of computation (Knight 1999b).

1. *By-hand Implementation:* the grammar is not implemented in a digital environment. It provides a deeper understanding of the working of creating and testing rules.

E.g.: De Stijl paintings (Knight 1989b), Mughul gardens (Stiny and Mitchell 1980), Hepplewhite chair-back (Knight 1980).

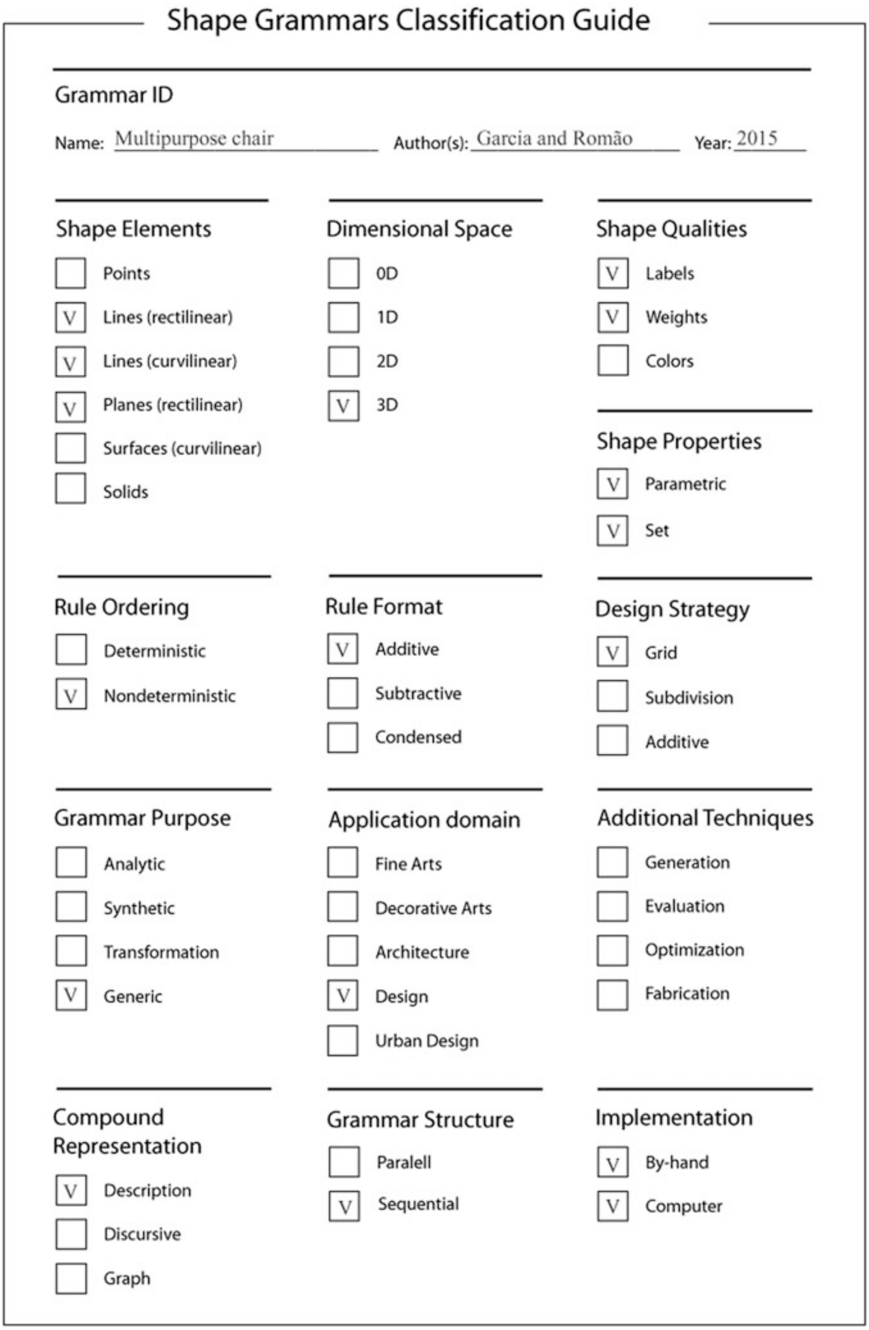
2. *Computer Implementation*: the grammar is implemented in a digital environment, or tested in an interpreter. It allows for a more rapid and bigger exploration of possibilities.

E.g.: Queen Anne houses (Flemming 1987), Malagueira houses (Duarte 2001), Cola-cola bottles (Chau et al. 2004).

Results

This section presents a classification diagram, in the form of a check-list (Fig. 9). The template is filled with an example of categorization of a specific shape grammar (Garcia and Romão 2015). Notice that the classifications can be applied to the whole or to sections of a shape grammar; hence the user can select more than one type in each category (or none).

Only the main types are presented. Some subtypes could be included, such as: curve type (arcs/Bézier curves); nondeterminism type (1/2/3); and application subdomain (e.g. product design). The basic elements and dimensional space of labels and weights are not included.



Shape Grammars Classification Guide

Grammar ID

Name: Multipurpose chair Author(s): Garcia and Romão Year: 2015

Shape Elements

- [ ] Points
- [V] Lines (rectilinear)
- [V] Lines (curvilinear)
- [V] Planes (rectilinear)
- [ ] Surfaces (curvilinear)
- [ ] Solids

Dimensional Space

- [ ] 0D
- [ ] 1D
- [ ] 2D
- [V] 3D

Shape Qualities

- [V] Labels
- [V] Weights
- [ ] Colors

Shape Properties

- [V] Parametric
- [V] Set

Rule Ordering

- [ ] Deterministic
- [V] Nondeterministic

Rule Format

- [V] Additive
- [ ] Subtractive
- [ ] Condensed

Design Strategy

- [V] Grid
- [ ] Subdivision
- [ ] Additive

Grammar Purpose

- [ ] Analytic
- [ ] Synthetic
- [ ] Transformation
- [V] Generic

Application domain

- [ ] Fine Arts
- [ ] Decorative Arts
- [ ] Architecture
- [V] Design
- [ ] Urban Design

Additional Techniques

- [ ] Generation
- [ ] Evaluation
- [ ] Optimization
- [ ] Fabrication

Compound Representation

- [V] Description
- [ ] Discursive
- [ ] Graph

Grammar Structure

- [ ] Paralell
- [V] Sequential

Implementation

- [V] By-hand
- [V] Computer

Fig. 9 Shape grammar classification guide

Discussion

The classification that is conducted in this paper elucidates how existing knowledge can be categorized based on several criteria, and thus launch a new aid for forthcoming grammar developments.

The categorization provides a systematic review of the known types. The developed framework stimulates new, important questions such as the main gaps in the overall theory, and some specific issues raised by each type. Within the given example of a shape grammar categorization, one can easily frame the work in progress, and to prove the novelty of the grammar.

Considering that shape grammars are currently under development, it would be profitable for this categorization process to become collaborative. That would allow to join other types that are not mentioned in the paper, and to include other forthcoming types. Furthermore, future work could be directed to the statistics on how many grammars were developed within each type, a timeline illustrative of all the mentioned types, and a database of examples.

Acknowledgements Funding for this research was provided by FCT (Fundação para a Ciência e Tecnologia) under PhD grant SFRH/BD/77927/2011 and by CIAUD (Centro de Investigação em Arquitectura Urbanismo e Design).

References

Agarwal M, Cagan J (1998) A blend of different tastes: the language of coffeemakers. Environ Plan B Plan Des 25:205–226

Agarwal M, Cagan J, Constantine K (1999) Influencing generative design through continuous evaluation: associating costs with the coffeemaker shape grammar. Artif Intell Eng Des Anal Manuf 13:253–275

Ang MC, Chau HH, McKay A, Pennington A (2006) Combining evolutionary algorithms and shape grammars to generate branded product design. Springer, Berlin, pp 521–539

Barros M (2015) Mass customization in the furniture design industry: the case of thonet chairs. Ph. D. thesis, Universidade de Lisboa

Beirão JN, Duarte JP, Stouffs R (2011) Creating specific grammars with generic grammars: towards flexible urban design. Nexus Netw J 13:73–111

Cagan J, Mitchell W (1993) Optimally directed shape generation by shape annealing. Environ Plan B Plan Des 20:5–12

Castro e Costa E, Duarte J (2013) Tableware shape grammar: towards mass customization of ceramic tableware. In: Proceedings of the 31st ECAADe conference

Chau HH, Chen X, McKay A, Pennington A (2004) Evaluation of a 3D shape grammar implementation. In: Des. Comput. Cogn. Kluwer Academic Publishers, Dordrecht, pp 357–376

Chomsky N (2002) Syntactic structures. Mounton de Gruyer, Berlin

Duarte JP (2001) Customizing mass housing: a discursive grammar for Siza's Malagueira houses. Ph.D. thesis, Massachusetts Institute of Technology

Eloy S, Duarte JP (2012) A transformation grammar-based methodology for housing rehabilitation. Des Comput Cogn 12:301–320

Ertelt C, Shea K (2008) Generative design and CNC fabrication using shape grammars. In: Proceedings of the ASME design engineering technical conference 2008

Flemming U (1987) More than the sum of parts: the grammar of Queen Anne houses. Environ Plan B Plan Des 14:323–350

Garcia S, Romão L (2015) A design tool for generic multipurpose chair design. In: City—new technol. Future Built Environ. 16th international conference on CAAD future 2015 Sao Paulo, July 8–10 2015. Springer, pp 600–619

Gips J (1975) Shape grammars and their uses: artificial perception, shape generation and computer aesthetics. Springer Basel AG, Basel

Hsiao S-W, Chen C-H (1997) A semantic and shape grammar based approach for product design. Des Stud 18:275–296

Jowers I, Earl C (2011) The implementation of curved shape grammars. Environ Plan B Plan Des 38:616–635

Jowers I, Earl C (2015) Extending the algebras of design. Nexus Netw J 17:1–16

Knight T (1980) The generation of Hepplewhite-style chair-back designs. Environ Plan B Plan Des 7:227–238

Knight T (1981) The forty-one steps. Environ Plan B 8:97–114

Knight T (1989a) Color grammars: designing with lines and colors. Environ Plan B Plan Des 16:417–449

Knight T (1989b) Transformations of De Stijl art: the paintings of Georges Vantongerloo and Fritz Glarner. Environ Plan B Plan Des 16:51–98

Knight T (1999a) Shape grammars: six types. Environ Plan B Plan Des 26:15–31

Knight T (1999b) Applications in architectural design, and education and practice. Report for the NSF/ MIT workshop on shape computation. Massachusetts Institute of Technology, Cambridge

Knight T (2003) Computing with emergence. Environ Plan B Plan Des 30:125–155

Koning H, Eizenberg J (1981) The language of the prairie: Frank Lloyd Wright's prairie houses. Environ Plan B 8:295–323

Krishnamurti R, Stouffs R (1993) Spatial grammars: motivation, comparison, and new results. In: CAAD Futures '93: Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures. CAAD Futures '93. Amsterdam: North-Holland, pp. 57–74

Lee HC, Tang MX (2004) Evolutionary shape grammars for product design. In: The seventh international conference generation art. Politecnico di Milano University, pp 309–322

Li A (2001) Teaching style grammatically, with an example from Chinese architecture. Proc Math Des 2011:270–277

McCormack J, Cagan J (2004) Speaking the Buick language: capturing, understanding, and exploring brand identity with shape grammars. Des Stud 25:1–29

Mendes L, Beirão JN, Duarte JP, Celani G (2013) A Bottom-up social housing system described with shape grammars: the case of Belapur low-income housing development. In: Proceedings of the 31st ECAADe conference. pp 705–714

Orsborn S, Cagan J, Pawlicki R, Smith RC (2006) Creating cross-over vehicles: defining and combining vehicle classes using shape grammars. Artif Intell Eng Des Anal Manuf 20:217–246

Pauwels P, Strobbe T, Eloy S, De Meyer R (2015) Shape grammars for architectural design: the need for reframing. In: CAAD future 2015, pp 507–526

Pugliese M, Cagan J (2002) Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar. Res Eng Des 13:139–156

Sass L (2005) Wood frame grammar: CAD scripting a wood frame house. In: CAAD futures. 2005. Springer, Berlin, pp 383–392

Shea K, Cagan J (1999) The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. Des Stud 20:3–23

Stiny G (1977) Ice-ray: a note on the generation of Chinese lattice designs. Environ Plan B 4:89–98

Stiny G (1980a) Kindergarten grammars: designing with Frobel's building gifts. Environ Plan B Plan Des 7:409–462

Stiny G (1980b) Introduction to shape and shape grammars. Environ Plan B Plan Des 7:343–351

Stiny G (1981) A note on the description of designs. Environ Plan B 8:257–267
Stiny G (1982) Spatial relations and grammars. Environ Plan B Plan Des 9:113–114
Stiny G (1990) What is a design? Environ Plan B Plan Des 17:97–103
Stiny G (1992) Weights. Environ Plan B Plan Des 19:413–430
Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. In: Proceedings of IFIP congress 1971
Stiny G, Gips J (1978) An evaluation of palladian plans. Environ Plan B 5:199–206
Stiny G, Mitchell W (1978) The palladian grammar. Environ Plan B Plan Des 5:5–18
Stiny G, Mitchell W (1980) The grammar of paradise: on the generation of Mughul gardens. Environ Plan B 7:209–226

From Shape Computations to Shape Decompositions

Djordje Krstic

Abstract This is a third paper in a series on shape decompositions, which are seen here as a means to grasp computations with shapes. That is, decompositions that arise in computations with shapes and may serve to explain them are investigated. Due to certain isomorphisms, computations with discrete and topological decompositions are carried out in parallel with shape computations thus providing insights into the latter. In particular, discrete decompositions grasp the transition of intuitive spatial computations envisioned by the designers into symbolic ones that could be carried out by a computer. Some counting has been done showing that even simple spatial computations require a great many symbols in order to be turned into the symbolic ones. It is interesting to explore the converse: turning complex symbolic computations with vast numbers of symbols into the simpler spatial ones with shapes. This may prove promising in tackling big data problems.

Introduction

This paper continues the study of decompositions of shapes in the context of design theory evolving around shape grammars. In the previous papers (Krstic 2004, 2008) and their extended versions (Krstic 2005, 2010) the emphasis was on the use of shape decompositions as shape approximations. Here decompositions appear in a new role, as tools for investigating computations with shapes. Such computations are of the outmost importance for design as computing with shapes is exactly what designers do when they draw, erase move, tweak, or pick shapes in order to complete their designs. Before moving any further some background information on shapes, their decompositions, and framework(s) for computations with them may prove useful.

D. Krstic (✉)
Krstic Design, Calabasas, USA
e-mail: gkrstich@aol.com

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_14

Shapes, Decompositions, Algebras, and Grammars

Shapes are important part of our everyday experience and play role in many human activities, but most crucially in design. Designers (engineers, architects, tradesmen, artists…) manipulate shapes—physical or abstract—to create new ones for different purposes. Design theory, which started with shape grammars (Stiny and Gips 1972) [and (Stiny 2006) for a recent account], formalizes these activities in terms of shapes and computations with them. The former are seen as sets of basic elements points, lines, planar segments, and solids—of dimension 0, 1, 2, and 3, respectively—while the latter are carried on in the framework of algebras for shapes (Stiny 1991) [and (Krstic 2014) for a recent account]. The algebras are equipped with Boolean operations of sum + and difference −, which model what designers do when they draw and erase shapes. The operations are based on the relation of partial order on shapes and work differently than familiar ones-with numbers. A shape is greater or equal (≥) to another shape if it is the only one we see when both are drawn. Shapes are added by simply drawing a new shape on top of the existing ones. However, only parts of the added shape not shared with the original shapes appear to be added to the drawing—not the whole shape. In contrast, when this new shape is subtracted, by erasing it from the drawing, only parts of the original shapes not shared with the erased shape remain—and not the whole original shapes. If $\boldsymbol{a}$ and $\boldsymbol{b}$ are shapes then expressions $\boldsymbol{a} + \boldsymbol{b} = \boldsymbol{a} + (\boldsymbol{b} - \boldsymbol{a})$ and $(\boldsymbol{a} + \boldsymbol{b}) - \boldsymbol{b} \leq \boldsymbol{a}$ reflect the previous statements. Needless to say that if $\boldsymbol{a}$ and $\boldsymbol{b}$ are numbers the first expression would be an inequality while the second would be an equality. Two more Boolean operations are useful for shapes: product, defined as $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a} - (\boldsymbol{a} - \boldsymbol{b}) = \boldsymbol{b} - (\boldsymbol{b} - \boldsymbol{a})$, and symmetric difference ⊕, given by $\boldsymbol{a} \oplus \boldsymbol{b} = (\boldsymbol{a} - \boldsymbol{b}) + (\boldsymbol{b} - \boldsymbol{a})$.

Designers use differently oriented shapes, which may vary in size and may also be mirrored or moved. To model that, an algebra has to be equipped with Euclidian transformations that could act on shapes by rotating, reflecting, translating or scaling them. Transformations could also be combined to produce compound transformations or to cancel each other. Two consecutive 30° rotations amount to a 60° one, which could be canceled by the same rotation in the opposite direction. In order to handle this, group operations of composition and inverse are added to the Boolean ones.

Finally, an algebra of shapes $\boldsymbol{U}_{ij}$ manipulates two sorts of objects: shapes with basic elements of dimension i defined in a space of dimension $j \geq i$ ($i, j \in \{0, 1, 2, 3\}$), and Euclidean transformations that could act on shapes. Shapes together with Boolean operations amount to the Boolean part of $\boldsymbol{U}_{ij}$ while transformations with group operations define its group part. The two parts interact via group action, which is the only two-sorted operation in the algebra. Group action takes a transformation, say $\boldsymbol{t}$, and a shape, say $\boldsymbol{a}$, to produce the transformed shape $\boldsymbol{t}(\boldsymbol{a})$.

One of defining characteristics of shapes, which distinguishes them from sets, is that they come unanalyzed rendering any division into parts possible. Where finite sets have only finitely many subsets, finite shapes—with basic elements of dimension 1 and above—have infinitely many parts to choose from. And that is exactly what we do when we try to describe shapes. We analyze and structure them in terms of their certain parts in effect creating *shape decompositions*. The latter are "…sets of shapes that show how their sums are divided into parts of certain kinds" (Stiny 1991). Decompositions thus serve as shape approximations and one should be able to compute with them in the same way as with unanalyzed shapes. Algebras of decompositions facilitate that. They come in two flavors: as set algebras $\boldsymbol{D}_{ij}$ and complex algebras $\wp(\boldsymbol{U}_{ij})$. Both combine transformations the same way $\boldsymbol{U}_{ij}$ does and both have the same group action operation: an extension of the one from $\boldsymbol{U}_{ij}$ to finite sets of shapes. That is, transformation $\boldsymbol{t}$ acts on decomposition A to produce decomposition $\boldsymbol{t}(A) = \{\boldsymbol{t}(\boldsymbol{a})|\ \boldsymbol{a} \in A\}$. The algebras differ in the way they combine decompositions. Set algebra $\boldsymbol{D}_{ij}$ does it using set operations of union and difference. Consequently, it treats shapes that are elements of decompositions as atoms or symbols without taking advantage of their spatial properties or combining them. In contrast, $\wp(\boldsymbol{U}_{ij})$ relies on operations from $\boldsymbol{U}_{ij}$ to combine elements of decompositions exhaustively. Operations + and − from $\wp(\boldsymbol{U}_{ij})$ are extensions of shape operations to direct products of decompositions. If X and Y are decompositions then $X + Y = +(X \times Y) = \{\boldsymbol{x} + \boldsymbol{y}|\ \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$ and $X - Y = -(X \times Y) = \{\boldsymbol{x} - \boldsymbol{y}|\ \boldsymbol{x} \in X, \boldsymbol{y} \in Y\}$. In their most general form decompositions are finite sets of arbitrary shapes without an apparent structure. However, they may come structured in an algebraic fashion to become lattices, topologies, and Boolean algebras, to name a few, or they could have some other structure like consisting of pairwise discrete shapes. The latter are *discrete decompositions* characterized by $\boldsymbol{a} \cdot \boldsymbol{b} = 0$, for any two different elements $\boldsymbol{a}$ and $\boldsymbol{b}$ of such a decomposition.

Algebras of shapes provide a framework for shape grammars, which are rule based production systems, equivalent to Turing machines and capable of generating designs in design languages. A shape grammar consists of a finite set of replacement rules where each rule is an ordered pair of shapes, say $(\boldsymbol{a}, \boldsymbol{b})$, customarily written as $\boldsymbol{a} \rightarrow \boldsymbol{b}$. Such a rule is applied (or acts) on an existing design, say shape $\boldsymbol{c}$, and changes it by replacing a transformed version of $\boldsymbol{a}$, or $\boldsymbol{t}(\boldsymbol{a})$, with $\boldsymbol{t}(\boldsymbol{b})$, provided $\boldsymbol{t}$ is a transformation that makes $\boldsymbol{t}(\boldsymbol{a})$ a part of $\boldsymbol{c}$. The replacement computation is done in the framework of algebra $\boldsymbol{U}_{ij}$ according to the formula

$$\boldsymbol{c}' = [\boldsymbol{c} - \boldsymbol{t}(\boldsymbol{a})] + \boldsymbol{t}(\boldsymbol{b}), \tag{1}$$

where $\boldsymbol{c}'$ is the altered shape.

Rules are applied in an exhaustive fashion to generate a new design. The process of generating a new design with shape grammar or shape *derivation* with grammar starts with an action of the initial rule of the form $0 \rightarrow \boldsymbol{b}$. Because empty shape 0 is a part of every shape, the choice of transformation $\boldsymbol{t}$ is not critical and one can pick it so to make the resulting shape $\boldsymbol{t}(\boldsymbol{b})$ of the desired size, orientation and position. A derivation is complete when no rule—other than the initial one-could be applied, or when some other previously defined conditions are met. The resulting shape is a design in the language specified by the grammar.

Computations and Decompositions

It has been mentioned earlier that describing a shape leads to its decomposition. It seems that even looking at one does the same. According to some vision theories shapes are recognized—as figures against the ground—based on how easily they could be broken into parts (Hoffman and Singh 1997). Moreover, this is done almost instantaneously via pre-attentive processing. It further seems that whenever we try to do something with shapes they end up structured. Computing with shapes is no exception. Any computation with unanalyzed shapes implies decomposing each of these in some way. These acquired structures may serve to explain the computation.

Let, for example, two shapes $\boldsymbol{a}$ and $\boldsymbol{b}$ engage in a computation carried out in an algebra $\boldsymbol{U}_{ij}$. Certain parts of $\boldsymbol{a}$ and $\boldsymbol{b}$ are readily distinguishable just from the fact that the two are combined-regardless of what the computation might be. Namely, the greatest part of $\boldsymbol{a}$ that is also a part of $\boldsymbol{b}$ or $\boldsymbol{a}\cdot\boldsymbol{b}$, the greatest part of $\boldsymbol{a}$ that does not have parts of $\boldsymbol{b}$, or $\boldsymbol{a} - \boldsymbol{b}$, and the greatest part of $\boldsymbol{b}$ that does not have parts of $\boldsymbol{a}$, or $\boldsymbol{b} - \boldsymbol{a}$. This leads to the respective decompositions of $\boldsymbol{a}$ and $\boldsymbol{b}$, $A = \{\boldsymbol{a}\cdot\boldsymbol{b}, \boldsymbol{a} - \boldsymbol{b}\}$ and $B = \{\boldsymbol{a}\cdot\boldsymbol{b}, \boldsymbol{b} - \boldsymbol{a}\}$. Because each of the shapes may also be combined with itself, shapes $\boldsymbol{a} = \boldsymbol{a}\cdot\boldsymbol{a} = \boldsymbol{a} + \boldsymbol{a}$, $\boldsymbol{b} = \boldsymbol{b}\cdot\boldsymbol{b} = \boldsymbol{b} + \boldsymbol{b}$, and $0 = \boldsymbol{a} - \boldsymbol{a} = \boldsymbol{b} - \boldsymbol{b}$ become available. Hence, the enlarged decompositions $A = \{\boldsymbol{a}, \boldsymbol{a}\cdot\boldsymbol{b}, \boldsymbol{a} - \boldsymbol{b}, 0\}$ and $B = \{\boldsymbol{b}, \boldsymbol{a}\cdot\boldsymbol{b}, \boldsymbol{b} - \boldsymbol{a}, 0\}$. Now, let the computation be the sum $\boldsymbol{a} + \boldsymbol{b}$. Shapes $\boldsymbol{a}$ and $\boldsymbol{b}$ are already structured—as A and B—so their sum should follow. Because $\boldsymbol{a}$ and $\boldsymbol{b}$ are parts of their sum, its decomposition C should at least include A and B, or $C = A \cup B$, but may also include some other shapes. The structure of A and B suggests inclusion of shapes $\boldsymbol{a} + \boldsymbol{b}$ and $(\boldsymbol{a} + \boldsymbol{b}) - (\boldsymbol{a}\cdot\boldsymbol{b}) = \boldsymbol{a} \oplus \boldsymbol{b}$, which makes $C = \{\boldsymbol{a} + \boldsymbol{b}, \boldsymbol{a} \oplus \boldsymbol{b}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a} - \boldsymbol{b}, \boldsymbol{b} - \boldsymbol{a}, \boldsymbol{a}\cdot\boldsymbol{b}, 0\}$. This is clearly a Boolean algebra generated by two generators $\boldsymbol{a}$ and $\boldsymbol{b}$, and relation $\boldsymbol{a} + \boldsymbol{b} = 1$, where 1 is the Boolean unit. Including more relations makes C smaller.

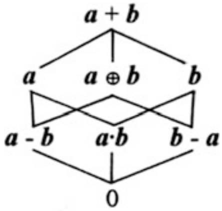
For example, if $\boldsymbol{a} \leq \boldsymbol{b}$ holds, then $C = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{b} - \boldsymbol{a}, 0\} = B$.

As a consequence of the computation, all of the shapes involved $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{a} + \boldsymbol{b}$ ended up structured in decompositions A, B and C, respectively.

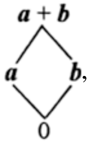
Argument Lattices and Argument Decompositions

The previous reasoning may be generalized to explain an arbitrary computation with shapes defined in $\boldsymbol{U}_{ij}$. Any shape that plays a role in a computation may be decomposed as above. That is, by recognizing the parts that emerge in combinations of arguments of the computation. Such decompositions are *argument decompositions*, and their upper bound is an *argument lattice*.

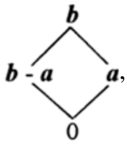
Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be the arguments of a computation resulting in shape $\boldsymbol{c}$. Further, let a Boolean algebra $\boldsymbol{L}$ be constructed with $\boldsymbol{a}$ and $\boldsymbol{b}$ as generators, and the set of relations between the generators that includes $\boldsymbol{a} + \boldsymbol{b} = 1$. Algebra $\boldsymbol{L}$ includes all of the shapes distinguishable in any of the computations with $\boldsymbol{a}$ and $\boldsymbol{b}$ as arguments. That is, all of the shapes obtainable as combinations of $\boldsymbol{a}$ and $\boldsymbol{b}$ in $\boldsymbol{U}_{ij}$. Decompositions A, B, and C obtained as sets of elements of $\boldsymbol{L}$ that are parts of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, respectively, include all of the parts of $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, that are recognized in the computation. Decompositions A, B, and C are therefore *principal ideals* of $\boldsymbol{L}$ generated by the respective shapes—where principal ideal $\mathbf{I}(\boldsymbol{x})$ generated by $\boldsymbol{x} \in L$ is the set of all elements of $\boldsymbol{L}$ smaller than or equal to $\boldsymbol{x}$. Thus the values of A and B, remain the same for all of the computations while C changes. For $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{b}$, C is as above while for $\boldsymbol{c} = \boldsymbol{a} - \boldsymbol{b}$, $\boldsymbol{c} = \boldsymbol{b} - \boldsymbol{a}$, $\boldsymbol{c} = \boldsymbol{a}{\cdot}\boldsymbol{b}$, and $\boldsymbol{c} = \boldsymbol{a} \oplus \boldsymbol{b}$, C becomes $\{\boldsymbol{a} - \boldsymbol{b}, 0\}$, $\{\boldsymbol{b} - \boldsymbol{a}, 0\}$, $\{\boldsymbol{a}{\cdot}\boldsymbol{b}, 0\}$, and $\{\boldsymbol{a} \oplus \boldsymbol{b}, \boldsymbol{a} - \boldsymbol{b}, \boldsymbol{b} - \boldsymbol{a}, 0\}$, respectively. Even trivial computations imply decompositions like $C = A$ if $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{a}$ or $\boldsymbol{c} = \boldsymbol{a}{\cdot}\boldsymbol{a}$ and $C = \{0\}$ if $\boldsymbol{c} = \boldsymbol{a} - \boldsymbol{a}$, $\boldsymbol{c} = \boldsymbol{b} - \boldsymbol{b}$, $\boldsymbol{c} = \boldsymbol{a} \oplus \boldsymbol{a}$ or $\boldsymbol{c} = \boldsymbol{b} \oplus \boldsymbol{b}$. The eight element Boolean algebra $\boldsymbol{L}$



includes all of the possible computations with shapes $\boldsymbol{a}$ and $\boldsymbol{b}$ as arguments, provided that no other relation—except $\boldsymbol{a} + \boldsymbol{b} = 1$—exists. Additional relations between the arguments make $\boldsymbol{L}$ smaller like Boolean algebras,



,



, and

, which correspond to relations $\boldsymbol{a}{\cdot}\boldsymbol{b} = 0$, $\boldsymbol{a} \leq \boldsymbol{b}$, and $\boldsymbol{a} = \boldsymbol{b}$, respectively.

It is easy to extend the notions above to computations with any number of arguments.

Definition 1 Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$ be a finite set of shapes defined in $\boldsymbol{U}_{ij}$, and let R be a set of relations among the elements of X. The Boolean algebra $Ar(X)$ generated—with the aid of operations from $\boldsymbol{U}_{ij}$—by generators X and relations $R \cup \{\sum X = 1\}$ is the *argument lattice* for all of the computations with $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n$ as arguments.

Definition 2 Let $\boldsymbol{x}$ be an element of $Ar(X)$. Principal ideal $\mathbf{I}(\boldsymbol{x})$ of $Ar(X)$ generated by $\boldsymbol{x}$ is the *argument decomposition* of $\boldsymbol{x}$, with respect to any computation with $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, … $\boldsymbol{x}_n$ as arguments.

Let $p(X) = \boldsymbol{a}$ be a computation, where $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$, and p is an n-ary polynomial of $\boldsymbol{U}_{ij}$. Shape $\boldsymbol{a}$ is clearly an element of the argument lattice $Ar(X)$, because the latter contains all of the shapes that are values of n-ary polynomials of $\boldsymbol{U}_{ij}$ for X. Argument decomposition $\mathbf{I}(\boldsymbol{a})$ together with decompositions $\mathbf{I}(\boldsymbol{x}_k)$, where $k = 1, 2, \ldots n$, may serve to explain computation $p(X) = \boldsymbol{a}$. If some shape $\boldsymbol{y}$ other than $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, … $\boldsymbol{x}_n$, or $\boldsymbol{a}$ plays a role in explaining the computation, $\mathbf{I}(\boldsymbol{y})$ may be considered as well, assuming that $\boldsymbol{y} \in Ar(X)$. Examples of such computations are examined in the next section, while $\boldsymbol{y} \notin Ar(X)$ case is left for future study.

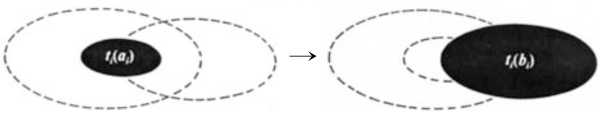
It is interesting to note that in some algebras of shapes $Ar(X)$ may fail to be a Boolean algebra. In particular, properties of algebras for weighted shapes (Stiny 1992) depend (also) on properties of weights, which may not be Boolean resulting in $Ar(X)$ being a lattice, but not a Boolean one. This motivates the use of the term argument lattice instead of, say, argument Boolean algebra.

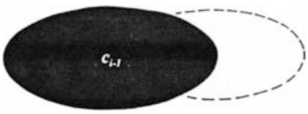
For example, Stiny (1992) introduced algebras of weighted lines with numeric weights standing for line thicknesses. For weights u and v, where $v \le u$ he specifies sum $u + v = v + u = u$, product $u{\cdot}v = v{\cdot}u = v$, and difference $v - u = 0$ in a Boolean fashion, while treating difference $u - v$ as an arithmetic one. Suppose that two spatially identical lines, but with different thicknesses—say lines l_1 and l_2 with respective thickness of 3 and 2—engage in a computation of some sort. Related argument lattice $Ar(\{l_1, l_2\})$ has $l_1 + l_2 = l_1$ as the unit, empty shape $l_1 - l_1 = l_2 - l_2 = 0$ as zero, and should also have the complements of the lines. The complement of l_2 is, in a Boolean fashion, line $l_2' = 1 - l_2 = l_1 - l_2$ of thickness 1 and it should, again in a Boolean fashion, satisfy identities $l_2 + l_2' = 1$ and $l_2{\cdot}l_2' = 0$. However, l_2' fails both identities because $l_2 + l_2' = l_2 \neq l_1 = 1$ and $l_2{\cdot}l_2' = l_2' \neq 0$, rendering $Ar(\{l_1, l_2\})$ a non-Boolean lattice.

Derivations and Decompositions

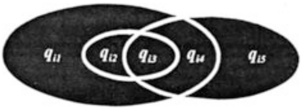
As mentioned earlier, spatial kinds of computations in $\boldsymbol{U_{ij}}$ emerge in the course of a shape derivation with a shape grammar. The derivation starts with action $\boldsymbol{r_1}$ of the initial rule $0 \rightarrow \boldsymbol{b_1}$ resulting, according to (1), in shape $\boldsymbol{c_1} = \boldsymbol{t_1}(\boldsymbol{b_1})$. Other rule actions follow producing a new shape at each step so that in an n-step derivation, a sequence $(\boldsymbol{c_i})_{i=1,\ldots,n}$ of n shapes is produced by the sequence $(\boldsymbol{r_i})_{i=1,\ldots,n}$ of n rule actions. In accordance with (1), each shape $\boldsymbol{c_i}$ in the sequence is produced from the previous shape $\boldsymbol{c_{i-1}}$ via $\boldsymbol{c_i} = [\boldsymbol{c_{i-1}} - \boldsymbol{t_i}(\boldsymbol{a_i})] + \boldsymbol{t_i}(\boldsymbol{b_i})$. The latter computation has three arguments $\boldsymbol{c_{i-1}}$, $\boldsymbol{t_i}(\boldsymbol{a_i})$, and $\boldsymbol{t_i}(\boldsymbol{b_i})$, which together with relation $\boldsymbol{t_i}(\boldsymbol{a_i}) \le \boldsymbol{c_{i-1}}$ generate argument lattice $Ar(\boldsymbol{r_i})$. The sequence of argument decompositions $\mathbf{I}(\boldsymbol{c_{i-1}})$, $\mathbf{I}(\boldsymbol{t_i}(\boldsymbol{a_i}))$, $\mathbf{I}(\boldsymbol{t_i}(\boldsymbol{b_i}))$ describes this computation.

For example, shape $\boldsymbol{c_i}$ 

defined in $\boldsymbol{U_{22}}$ is produced when rule 

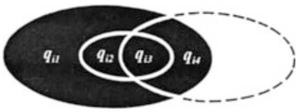
acts on shape $\boldsymbol{c_{i-1}}$ 

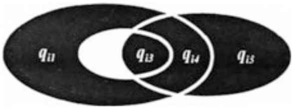
. The related argument lattice $Ar(\boldsymbol{r_i})$ is an atomic

Boolean algebra whose set of atoms $At(Ar(\boldsymbol{r_i}))$ has five elements  defined by:

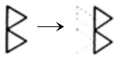
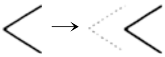
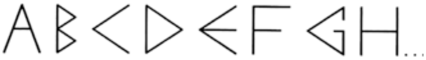
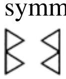
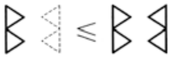
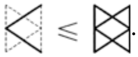
$$
\begin{aligned}
\boldsymbol{q}_{i1} &= \boldsymbol{c}_{i-1} - \boldsymbol{t}_i(\boldsymbol{a}_i + \boldsymbol{b}_i) \\
\boldsymbol{q}_{i2} &= \boldsymbol{t}_i(\boldsymbol{a}_i - \boldsymbol{b}_i) \\
\boldsymbol{q}_{i3} &= \boldsymbol{t}_i(\boldsymbol{a}_i \cdot \boldsymbol{b}_i) \\
\boldsymbol{q}_{i4} &= \boldsymbol{c}_{i-1} \cdot \boldsymbol{t}_i(\boldsymbol{b}_i - \boldsymbol{a}_i) \\
\boldsymbol{q}_{i5} &= \boldsymbol{t}_i(\boldsymbol{b}_i) - \boldsymbol{c}_{i-1}
\end{aligned}
\tag{2}
$$

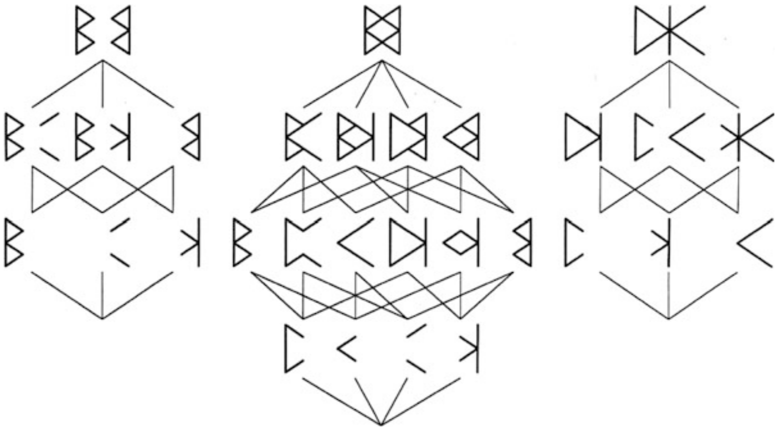
The lattice has $2^5 = 32$ elements, and because the only relations among the generators are $\boldsymbol{c_{i-1}} + \boldsymbol{t_i}(\boldsymbol{a_i}) + \boldsymbol{t_i}(\boldsymbol{b_i}) = 1$ and $\boldsymbol{t_i}(\boldsymbol{a_i}) \le \boldsymbol{c_{i-1}}$, $Ar(\boldsymbol{r_i})$ is the biggest an argument lattice for a single rule action can get. The argument decompositions of shapes $\boldsymbol{c_{i-1}}$ and $\boldsymbol{c_i}$ are atomic Boolean algebras with four atoms each. Atoms

 $(\boldsymbol{q_{i1}}, \boldsymbol{q_{i2}}, \boldsymbol{q_{i3}}, \boldsymbol{q_{i4}})$ are that of $\mathbf{I}(\boldsymbol{c_{i-1}})$ while atoms

 $(\boldsymbol{q_{i1}}, \boldsymbol{q_{i3}}, \boldsymbol{q_{i4}}, \boldsymbol{q_{i5}})$ belong to $\mathbf{I}(\boldsymbol{c_{i-1}})$.

An n-step derivation may be seen as a single computation resulting in shape $\boldsymbol{c_n}$ defined recursively by $\boldsymbol{c_i} = [\boldsymbol{c_{i-1}} - \boldsymbol{t_i}(\boldsymbol{a_i})] + \boldsymbol{t_i}(\boldsymbol{b_i})$, and satisfying relations $\boldsymbol{t_i}(\boldsymbol{a_i}) \le \boldsymbol{c_{i-1}}$ for $i = 1, 2, 3, \ldots n$. This creates an opportunity of representing all of the shapes in the derivation as respective argument decompositions related to this computation. Its argument lattice $\mathrm{Ar}(C_n)$ is generated with set $\{\boldsymbol{t_1}(\boldsymbol{b_1}), \boldsymbol{t_2}(\boldsymbol{a_2}), \boldsymbol{t_2}(\boldsymbol{b_2}), \ldots \boldsymbol{t_n}(\boldsymbol{a_n}), \boldsymbol{t_n}(\boldsymbol{b_n})\}$ of $2n - 1$ generators satisfying $n - 1$ relations of the form $\boldsymbol{t_i}(\boldsymbol{a_i}) \le \boldsymbol{c_{i-1}}$ $(i = 2, 3, \ldots n)$.

For example, two rules  and  that translate letters B and C from linear font  along their respective axes of symmetry may be used sequentially in a derivation that starts with initial rule →  having shape with two B letters and ends with shape  spelling DK. This may seem paradoxical, because BB is transformed by translating a B and a C shape to produce DK. However, shape  that is produced by applying the rule that translates B has a number of different elements from the font as parts. These include  (B, C, D, I, K, M, and 1). The second rule picks C from the shape and moves it to the right, leaving D and I behind. The translated C combines with I to produce K, so that DK emerges. The derivation is given by sequence . The derivation's argument lattice is produced with five shapes  as generators, and two relations  and . It has six atoms  and 64 elements. Sequence of argument decompositions



describes the derivation.

Properties of Argument Decompositions

An argument lattice $Ar(X)$ is a finite Boolean algebra if its elements are shapes defined in $\boldsymbol{U}_{ij}$. Consequently, argument decomposition $\mathbf{I}(\boldsymbol{a})$ for $\boldsymbol{a} \in Ar(X)$ is also a Boolean algebra because it is an ideal of a Boolean algebra. The argument lattice itself is an argument decomposition, namely, the one generated by the sum of the generators $\sum X$, or $Ar(X) = \mathbf{I}(\sum X)$. Note that $\sum X = 1 \in Ar(X)$ by the definition of $Ar(X)$. Argument decompositions are also topological decompositions of shapes (Stiny 1994; Krstic 2008, 2010), and any topological decomposition that is a Boolean algebra is an argument lattice for some set of generators. If this is a non-trivial topological decomposition—that is, the one with more than two elements—then it may be generated by more than one set of generators. Consequently, it may represent more than one argument lattice.

For example, an atomic Boolean algebra may be generated by the set of its atoms, as well as by the set of its dual atoms.

It is often more convenient to deal with sets of atoms than with argument decompositions. This is especially true in longer computations where an argument lattice may become gargantuan. It is then useful to be able to produce the set of atoms of some argument decomposition without having the decomposition itself. The following proposition provides for that.

Proposition 1 *Let S be a subset of X. Shape $\alpha(S)$ defined by $\alpha(S) = \prod S - \sum(X - S)$ is either an atom of $Ar(X)$ or 0.*

It is clearly so, because the product of $\alpha(S)$ with any shape generated by S is $\alpha(S)$ and with any shape generated by $X - S$ is 0.

The set of atoms of $Ar(X)$ can, according to the proposition above, be obtained as the union of α shapes for all subsets of X or

$$At(Ar(X)) = \cup_{S \in \wp(X)} \alpha(S) - \{0\} \tag{3}$$

The possibility of constructing the atoms of an argument lattice without constructing the lattice itself, justifies abbreviation $At(X) = At(Ar(X))$.

Argument lattice $Ar(X)$ is clearly a subalgebra of $\boldsymbol{U_{ij}}$ and the same is true for each of its argument decompositions $\mathbf{I}(\boldsymbol{x})$, $\boldsymbol{x} \in Ar(X)$. It is less clear that set $\mathbf{I}(Ar(X)) = \{\mathbf{I}(\boldsymbol{x}) | \ \boldsymbol{x} \in Ar(X)\}$ of argument decompositions is a subalgebra of $\wp(\boldsymbol{U_{ij}})$ and the set of sets of their atoms $At(\mathbf{I}(Ar(X))) = \{At(\mathbf{I}(\boldsymbol{x})) | \ \boldsymbol{x} \in Ar(X)\}$ is a subalgebra of $\boldsymbol{D_{ij}}$. Moreover, all three subalgebras $Ar(X)$, $\mathbf{I}(Ar(X))$, and $At(\mathbf{I}(Ar(X)))$ are isomorphic.

Proposition 2 *Let $Ar(X)$ be an argument lattice.*

i) Set $\boldsymbol{I}(Ar(X))$ is a subalgebra of $\wp(\mathbf{U_{ij}})$ isomorphic to $Ar(X)$.
ii) Set $At(\boldsymbol{I}(Ar(X)))$ is a subalgebra of $\mathbf{D_{ij}}$ isomorphic to $\boldsymbol{I}(Ar(X))$.
iii) $Ar(X)$ and $At(\boldsymbol{I}(Ar(X)))$ are isomorphic Boolean algebras.

Note, that the proof of the proposition above is given in Krstic (1996) (pages 142–144) and will be omitted here.

The proposition allows for computations with shapes to be duplicated with their discrete and argument decompositions. Let p_U, $p_\wp$, and p_D be the same polynomial p defined in $\boldsymbol{U_{ij}}$, $\wp(\boldsymbol{U_{ij}})$, and $\boldsymbol{D_{ij}}$, respectively. The computation described by p may then be carried out in parallel by p_U, $p_\wp$, and p_D in all of the algebras above. This is represented by the commutative diagram

$$\begin{tikzcd}
Ar(X) \arrow[r, shift left, "\mathbf{I}"] \arrow[d, "p_U"'] & \mathbf{I}(Ar(X)) \arrow[l, shift left, "\mathbf{I}^{-1} = \sum"] \arrow[r, shift left, "At"] \arrow[d, "p_\wp"] & At(\mathbf{I}(Ar(X))) \arrow[l, shift left, "At^{-1} = cl_+"] \arrow[d, "p_D"] \\
Ar(X) \arrow[r, shift right, "\mathbf{I}"'] & \mathbf{I}(Ar(X)) \arrow[l, shift right, "\mathbf{I}^{-1} = \sum"'] \arrow[r, shift right, "At"'] & At(\mathbf{I}(Ar(X))) \arrow[l, shift right, "At^{-1} = cl_+"']
\end{tikzcd},$$

which may also serve to illustrate Proposition 2.

It is interesting to note that computations conducted in $\boldsymbol{D_{ij}}$ with the set of arguments $At(\mathbf{I}(X))$ do not take advantage of spatial properties of shapes. That is, if the shapes that are elements of every $At(\mathbf{I}(\boldsymbol{x}))$, where $\boldsymbol{x} \in X$ are substituted with symbols, the computation will remain the same. Moreover, $At(\mathbf{I}(\boldsymbol{x}))$ is the smallest decomposition with this property.

Proposition 3 *Let Ar(X) be the argument lattice for all of the spatial computations carried out in* $\boldsymbol{U}_{ij}$ *with X as the set of arguments. Set* $At(\mathbf{I}(Ar(X)))$ *contains the least decompositions of elements of Ar(X) that allow for these computations to be carried out with symbols.*

According to Proposition 2, iii, a spatial computation with X as the argument set, carried out in $\boldsymbol{U}_{ij}$ may be duplicated with symbols in $\boldsymbol{D}_{ij}$. Sets $Ar(X)$ and $At(\mathbf{I}(Ar(X)))$ are the least upper bounds for all of such spatial and symbolic computations, respectively. Because $Ar(X)$ is the least Boolean algebra that contains X, and $At(\mathbf{I}(Ar(X))) = \wp(At(X))$ elements of $At(\mathbf{I}(Ar(X)))$ are the least decompositions of elements of $Ar(X)$ that allow for all of the computations with shapes—described above—to be duplicated with sets of symbols.

Size of Argument Lattice

The size of an argument lattice, generated by some set X of n generators, depends on the relations among the generators. $Ar(X)$ is always smaller than a free Boolean algebra generated by an n-element set. That is, X is never entirely "free" of relations, as relation $\sum X = 1$ is always assumed. In the most general case this is the only relation and $Ar(X)$ is maximum.

Proposition 4 *An argument lattice related to some computation with n different arguments has* 2^k *elements, where k is an integer that satisfies the following:*

i) $k = 2^n - 1$ *in the maximum case, and*
ii) $k = \log_2 n$ *in the minimum one.*

Let $Ar(X)$ be such a lattice. Because $Ar(X)$ is a finite Boolean algebra, k is the number of its atoms. Let S be a subset of X, the shape $\alpha(S)$ defined in Proposition 1 is either an atom of $Ar(X)$ or 0. In the first case, the only relation between elements of X is $\sum X = 1$ so that $\alpha(\emptyset) = \prod\emptyset - \sum X = 1 - 1 = 0$. Hence, $\alpha(\emptyset)$ is not an atom of $Ar(X)$. However, all of the other subsets of X yield atoms so that $Ar(X)$ ends up with $k = 2^n - 1$ atoms. Note that $\prod\emptyset = 1$ is often assumed in topology; however, in set theory it is considered undefined. If the latter approach is preferred, then $\alpha(\emptyset)$ becomes undefined, so again it is not an atom. In the minimum case, one has to look for the smallest set of atoms that will generate all n elements of X, which is clearly $k = \log_2 n$.

It is obvious from the proposition above that $Ar(X)$ can have a huge range of sizes. For example, an argument lattice for a computation with seven arguments may range from 8 elements to 2^{127} elements. This also shows how a spatial computation with shapes may become cumbersome when duplicated with symbols.

When a computation is conducted in the framework of shape grammars, some new relations between the arguments are introduced. Namely, $\boldsymbol{t_i}(\boldsymbol{a_i}) \leq \boldsymbol{c_{i-1}}$ for each $2 \leq i \leq m$, where m is the number of steps in the derivation. These relations affect the size of a maximum argument lattice for an m-step derivation. It becomes smaller than the maximum argument lattice for computations with the same number of arguments $n = 2\,m - 1$.

Proposition 5 *The argument lattice for an m-step derivation has at most 2^k elements, where $k = 2^{2(m-1)} + 2^{m-1} - 1$.*

Let $Ar(C)$ be an argument lattice for an m-step derivation C. $Ar(C)$ is generated by the set $X = \{\boldsymbol{t_1}(\boldsymbol{b_1}), \boldsymbol{t_2}(\boldsymbol{a_2}), \boldsymbol{t_2}(\boldsymbol{b_2}), \ldots \boldsymbol{t_n}(\boldsymbol{a_n}), \boldsymbol{t_n}(\boldsymbol{b_n})\}$ of 2 m − 1 generators. For every subset S of X there is a shape $\alpha(S)$, which is either an atom of $Ar(C)$ or the empty shape, in accordance with Proposition 1. There are 2^{2m-1} such shapes so that the number of atoms of $Ar(C)$ is $k = 2^{2m-1} - v$, where v is the number of instances of $\alpha(S)$ that are 0. For each $l = 1, 2, \ldots m - 1$ define decomposition $\beta(l) = \{\boldsymbol{t_1}(\boldsymbol{b_1}), \boldsymbol{t_2}(\boldsymbol{b_2}), \ldots \boldsymbol{t_l}(\boldsymbol{b_l})\}$ consisting of all the shapes that are added in the first l steps of the derivation. Shape $\boldsymbol{c_l}$ is clearly a part of $\sum\beta(l)$ and because relation $\boldsymbol{t_{l+1}}(\boldsymbol{a_{l+1}}) \leq \boldsymbol{c_l}$ holds it follows that $\boldsymbol{t_{l+1}}(\boldsymbol{a_{l+1}}) \leq \sum\beta(l)$. Consequently, if some $S \subseteq X$ is such that $\boldsymbol{t_{l+1}}(\boldsymbol{a_{l+1}}) \in S$ and $\beta(l) \subseteq X - S$ then $\alpha(S) = 0$. Every subset of $X - \beta(l)$ that has $\boldsymbol{t_{l+1}}(\boldsymbol{a_{l+1}})$ as an element, satisfies this condition. Because $X - \beta(l)$ has $2\,m - 1 - l$ elements, there are exactly 2^{2m-2-l} such subsets for each l, which makes $\sum_{l=1,\ldots m} 2^{2(m-1)-l} = 2^{2(m-1)} - 2^{\,m-1}$ the total of such subsets. It was shown earlier that $\alpha(\emptyset) = 0$ so that $v = 2^{2(m-1)} - 2^{\,m-1} + 1$ and $k = 2^{2m-1} - (2^{2(m-1)} - 2^{\,m-1} + 1) = 2^{2(m-1)} + 2^{\,m-1} - 1$.

Continuity

Stiny (1994) introduces topological decompositions of shapes as means to explain computations with shape grammars. He uses continuous mappings—describing a certain aspect of rule actions—to analyze the shapes in a derivation. Continuous mappings guarantee that the analysis proceeds without incompatible divisions of the analyzed shapes i.e. in a continuous fashion. As mentioned earlier, argument decompositions are topological decompositions and are also used for analyzing the shapes in a derivation. Because they belong to the argument lattice of the derivation the divisions of the analyzed shapes are compatible, rendering the analysis continuous. Moreover, such argument decompositions are upper bounds for all of the topological decompositions describing the derivation in a continuous fashion. The latter decompositions also belong to the argument lattice of the derivation, which is the upper bound of the argument decompositions.

Application of Argument Decompositions

Argument decompositions as descriptions of computations with shapes provide an important link between different systems that manipulate shapes spatially and symbolically. This is especially interesting when the computations are conducted within the framework of shape grammars.

Most of the intuitive appeal of shape grammars stems from the fact that they treat shapes as spatial objects. Spatial properties of shapes, like emergence-that are often used by designers-are fully appreciated by shape grammars and associated algebras. In some cases, however, emergence does not play any role in design. It is then possible to treat shapes symbolically as in computer-aided design, where shapes are divided into predetermined parts to suit some data structures. Stiny (1982) introduces set grammars, a formalism similar to shape grammars, except that it regards shapes as symbols. Both formalisms are of equal computational power and the former can be used in place of the latter whenever it is possible to describe shapes in terms of kits of parts. “Even so, set grammars are perhaps better behaved as algorithms; they are particularly close relatives of the symbolic production systems of computer science” (Stiny 1982). Set grammars are well suited for generating designs from a given (predetermined) vocabulary of shapes. For example, designs composed with Froebel’s building blocks (Stiny 1980), or buildings produced in a building system. They treat shapes as sets of symbols manipulating them with the set operations of $\boldsymbol{D_{ij}}$. “Those wishing to combine the multidimensional properties of shape grammars with the symbolic properties of standard production systems for their own special computer applications should find set grammars an ideal compromise. Others more interested in the spatial properties of designs will still find shape grammars the better bet” (Stiny 1982).

Given the differences between the two formalisms, it seems promising to analyze computations with shapes conducted in one of the formalisms by duplicating them in the other. This can answer the questions of what makes a symbolic computation look like a spatial one, or more importantly, how a spatial computation can be recast as a symbolic one. A computation carried out with sets of shapes symbolically-like computations carried out in set grammars—may easily be duplicated to appear as if it was carried out with shapes spatially, like computations carried out in shape grammars. It is sufficient to substitute sets of shapes with their sums. The former are manipulated in $\boldsymbol{D_{ij}}$ and the latter in $\boldsymbol{U_{ij}}$. It, however, takes some work to go the other way.

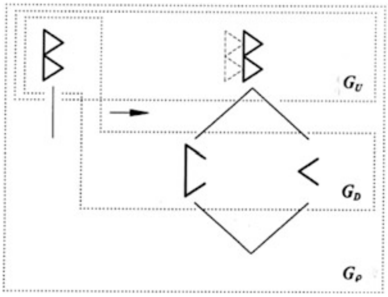
When designs are generated with a shape grammar, there is no such thing as a predetermined vocabulary of shapes that is necessary to define a set grammar. However, such a vocabulary can be distinguished after a computation has taken place.

Let C be an n-step derivation with shape grammar $\boldsymbol{G_U}$, carried out in $\boldsymbol{U_{ij}}$, and let $Ar(C)$ be its argument lattice. According to Proposition 3, iii, this computation with shapes may be duplicated with sets of atoms of their argument decompositions. Set $At(\mathbf{I}(Ar(C)))$ is then the least upper bound for all of such computations, in accordance with Propositions 2 and 3. Because $At(\mathbf{I}(Ar(C))) = \wp(At(C))$ set of atoms $At(C)$ of $Ar(C)$ may serve as a vocabulary of shapes to define a set grammar $\boldsymbol{G_D}$. Shapes $\boldsymbol{c_1}, \boldsymbol{c_2}, \ldots \boldsymbol{c_n}, \boldsymbol{t_1}(\boldsymbol{b_1}), \boldsymbol{t_2}(\boldsymbol{a_2}), \boldsymbol{t_2}(\boldsymbol{b_2}), \ldots \boldsymbol{t_n}(\boldsymbol{a_n}), \boldsymbol{t_n}(\boldsymbol{b_n})$ are then represented by the sets of atoms of their respective argument decompositions: $At(\mathbf{I}(\boldsymbol{c_1}))$, $At(\mathbf{I}(\boldsymbol{c_2}))$, … The rules of $\boldsymbol{G_U}$ are transformed into the rules of $\boldsymbol{G_D}$ so that if $\boldsymbol{a_k} \to \boldsymbol{b_k}$, $k = 1, \ldots n$, is a rule of the former, $At(\mathbf{I}(\boldsymbol{a_k})) \to At(\mathbf{I}(\boldsymbol{b_k}))$ is a rule of the latter, where $\mathbf{I}(\boldsymbol{a_k})$ and $\mathbf{I}(\boldsymbol{b_k})$ are obtained as $\boldsymbol{t_k^{-1}}(\mathbf{I}(\boldsymbol{t_k}(\boldsymbol{a_k})))$ and $\boldsymbol{t_k^{-1}}(\mathbf{I}(\boldsymbol{t_k}(\boldsymbol{b_k})))$, respectively. Therefore, argument decompositions may serve as a means for converting manipulations with shapes familiar to designer, into the computations with sets of symbols understandable to a computer. Moreover, argument decompositions, with all of the possible spatial combinations of their atoms, explore the spatial possibilities within the constraints imposed by the symbolic treatment of shapes. This justifies recasting computations with shapes into computations with argument decompositions.

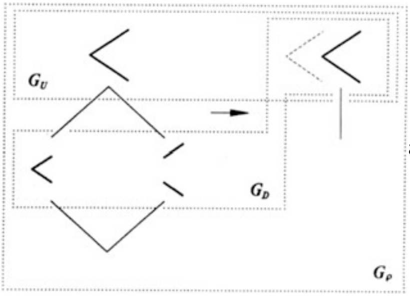
Let C, $\boldsymbol{G_U}$, $Ar(C)$ and $\boldsymbol{G_D}$ be as above. The computation describing C carried out in $\boldsymbol{U_{ij}}$ can be duplicated in $\wp(\boldsymbol{U_{ij}})$. The new grammar $\boldsymbol{G_\wp}$ has rules of the form $\mathbf{I}(\boldsymbol{a_k}) \to \mathbf{I}(\boldsymbol{b_k})$.

It is, therefore, possible to see shapes in three different ways: as spatial entities in $\boldsymbol{U_{ij}}$, as sets of symbols in $\boldsymbol{D_{ij}}$, and as combinations of both in $\wp(\boldsymbol{U_{ij}})$. Computations with shapes are then carried out by three different grammars $\boldsymbol{G_U}$, $\boldsymbol{G_D}$, and $\boldsymbol{G_\wp}$, respectively.

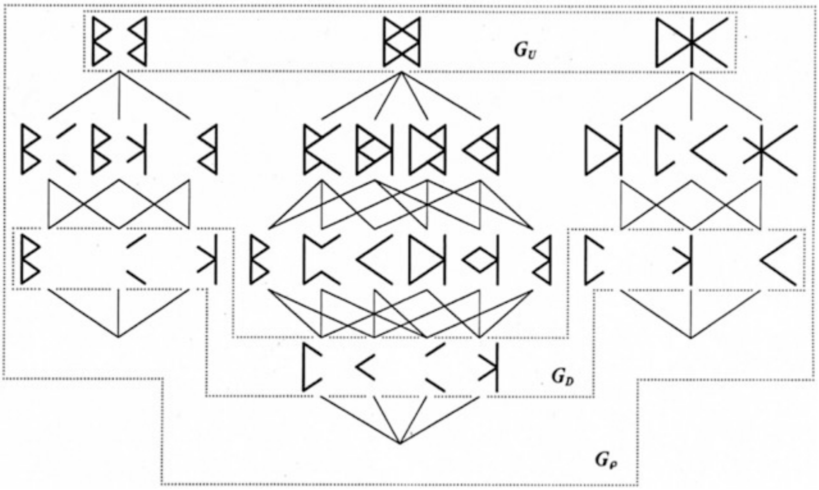
The 3-step derivation from the previous example may be carried out by grammars $\boldsymbol{G_U}$, $\boldsymbol{G_D}$, and $\boldsymbol{G_\wp}$ so that rules

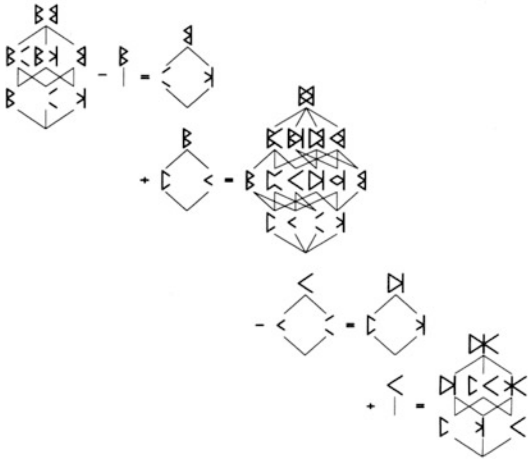


and

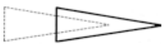


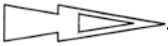
act to produce triple derivation.

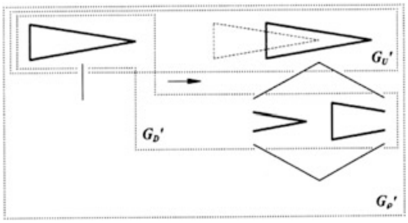


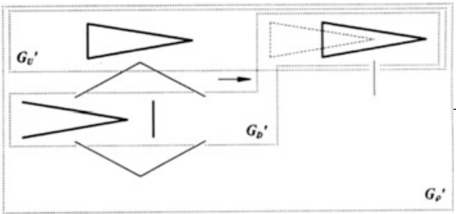
Computations  support the 3-step derivation by grammar $G_{\wp}$ carried out with argument decompositions in algebra $\wp(U_{ij})$.

Grammars G_U, G_D, and $G_{\wp}$ have the same number of rules. In fact, each rule of G_U is matched with the corresponding rules in G_D and $G_{\wp}$. However, this may not always be the case. Whenever the same rule is used more than once in a derivation by G_U, it is possible for each of its applications to be paired with a different rule in G_D and $G_{\wp}$.

For example, let $G_U{}'$ be a simple grammar with $\rightarrow$  the initial rule and rule  $\rightarrow$  that translates a triangle. In the first step of a derivation with $G_U{}'$ the initial rule introduces shape

 which is then altered by the other rule acting on the left triangle of the shape so that  is produced. This shape is further altered by the same rule, this time acting on the emergent triangle (in the middle), to produce the final arrowhead shape .

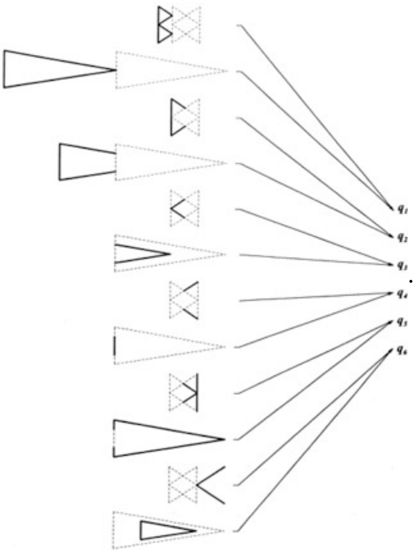
When the computation above is duplicated in $\boldsymbol{D_{ij}}$ and $\wp(\boldsymbol{U_{ij}})$ two new grammars $\boldsymbol{G_D}'$ and $\boldsymbol{G_\wp}'$ emerge. The argument lattice $Ar(C')$ of the derivation has 64 elements and six atoms as in the previous example. Similarly, the grammars $\boldsymbol{G_D}'$ and $\boldsymbol{G_\wp}'$ have three rules each. However, $\boldsymbol{G_U}'$ has only two rules where in the previous example $\boldsymbol{G_U}$ had three. The rule that translates a triangle in $\boldsymbol{G_U}'$ becomes 

and 

—two rules. Note how the $\boldsymbol{G_U}'$ parts of both rules are the same.

These two example may provide some insights on what is gained by converting spatial computations of $\boldsymbol{U_{ij}}$ to symbolic ones of $\boldsymbol{D_{ij}}$ and $\wp(\boldsymbol{U_{ij}})$.

From the "spatial" point of view, the only similarity between the two derivations is that they are of the same length. Otherwise, they may look quite different. The shapes are different, the rules are different, even their number in one grammar is different than in the other. However, from the "symbolic" point of view, the two computations look pretty much the same. Both argument lattices $Ar(C)$ and $Ar(C')$ have 64 elements and six atoms each. Decompositions of the relevant shapes in both of the computations have the same respective numbers of elements, and $Ar(C')$ can be obtained from $Ar(C)$ and vice versa by a simple exchange of their atoms. Moreover, one grammar can be transformed into the other. Let the atoms of $Ar(C)$ and $Ar(C')$ be mapped onto the set of symbols $\{\boldsymbol{q_1}, \boldsymbol{q_2}, \ldots \boldsymbol{q_6}\}$ as follows



. The rules are obtained via argument decompositions of the related shapes: $t_1(b_1)$ for the initial rule, $t_2(a_2)$ and $t_2(b_2)$ for the second, and $t_3(a_3)$ and $t_3(b_3)$ for the third rule. The initial rule emerges when $\rightarrow q_1 + q_4 + q_5$ is transformed via t_1^{-1}, the second when $q_1 \rightarrow q_2 + q_3$ is transformed via t_2^{-1}, and the third when $q_3 + q_4 \rightarrow q_6$ is transformed via t_3^{-1}. Therefore, the rules are:

i) $\rightarrow t_1^{-1}(q_1 + q_4 + q_5)$
ii) $t_2^{-1}(q_1) \rightarrow t_2^{-1}(q_2 + q_3)$
iii) $t_3^{-1}(q_3 + q_4) \rightarrow t_3^{-1}(q_6)$

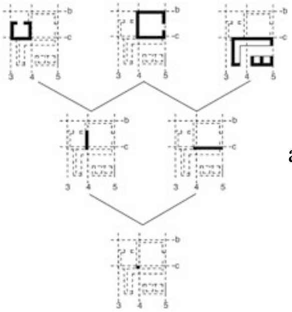
To get the rules of G_U and G_U' in their usual spatial form, it is sufficient to compute i), ii), and iii) with the appropriate values for atoms and transformations. Coincidently, when ii) and iii), are computed in U_{ij} with the values from $Ar(C')$ identities $t_2^{-1}(q_1) = t_3^{-1}(q_3 + q_4)$ and $t_2^{-1}(q_2 + q_3) = t_3^{-1}(q_6)$ emerge resulting in G_U' having one rule where G_U has two.

This example suggests the possibility of using argument lattices and argument decompositions to provide for algebraic machinery to facilitate transformations of shape grammars and languages of designs. The program of transformations in design developed by Terry Knight (1994) may be enriched by these tools. They may also lead to some kind of algebraic classification of shape grammars.

There is a possibility to use argument lattices for constructing new building systems. This can be achieved by the following program:

i) Define one or more shape grammars for desired buildings;
ii) Generate a number of prototype buildings using the grammars;
iii) Take the atoms of the derivations in ii) as the building blocks of the system.

This program is truly design oriented; it starts with designs and leads to building systems. It differs from the usual approach where building systems are developed around certain building technologies. Papers (Krstic 2008, 2010) provide a glimpse into this possibility. There, a plan of a residence and the outlines of the rooms—which are the design elements—are used to extract the walls and joints between the walls—which are the structural elements. Computations



are carried out in a $\boldsymbol{U_{22}}$ algebra, where the product operation is used twice to extract the walls and again to single out their joint.

Given that even simple spatial computations with shapes require big sets of symbols to be carried out "symbolically" it may be interesting to explore the converse: turning complex computations with vast numbers of symbols into simpler spatial ones with shapes. This could help in visualizing complex problems and prove promising in tackling big data. This approach is not without precedents. Stouffs and Krishnamurti (Stouffs and Krishnamurti 2004) extend the shape representation logic-of shape grammars theory—"to other, non-geometric, data, without necessarily considering non-spatial information as attributes to shapes." Elsewhere (Krstic 2008, 2010), topological decompositions of shapes yield some functional (non-spatial) information. For example, information on which component of a broken object needs to be replaced and which parts of the broken component could be salvaged is obtained via simple interior/closure arithmetic in a $\boldsymbol{U_{ij}}$ algebra.

References

Hoffman D, Singh M (1997) Salience of visual parts. Cognition 63:29–78

Knight TW (1994) Transformations in design: a formal approach to stylistic change and innovation in visual arts. Cambridge University Press, Cambridge

Krstic D (1996) Decompositions of shapes. Ph.D. thesis UCLA

Krstic D (2004) Computing with analyzed shapes. In: Gero JS (ed) Design computing and cognition'04. Springer, Berlin, pp 397–416

Krstic D (2005) Shape decompositions and their algebras. AIEDAM 19:261–276

Krstic D (2008) Approximating shapes with topologies. In: Gero JS, Goel AK (eds) Design computing and cognition'08. Springer, Berlin, pp 81–100

Krstic D (2010) Approximating shapes with hierarchies and topologies. AIEDAM 24:259–276

Krstic D (2014) Algebras of shapes revisited. In: Gero JS (ed) Design computing and cognition '12. Springer, Berlin, pp 361–376

Stiny G (1980) Kindergarten grammars: designing with Frobel's building gifts. Environ Plan B 7:251–343

Stiny G (1982) Spatial relations and grammars. Environ Plan B 9:113–114
Stiny G (1991) The algebras of design. Res Eng Design 2:171–181
Stiny G (1992) Weights. Environ Plan B 19:413–430
Stiny G (1994) Shape rules: closure, continuity, and emergence. Environ Plan B 21:S49–S78
Stiny G (2006) Shape: talking about seeing and doing. The MIT Press, Cambridge
Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. In: Freiman CV (ed) Information processing 71. North-Holland, Amsterdam, pp 1460–1465
Stouffs R, Krishnamurti R (2004) Data views, data recognition and design rules. In: Gero JS (ed) Design computing and cognition'04. Springer, Berlin, pp 219–238

Automated Induction of General Grammars for Design

Mark Whiting, Jonathan Cagan and Philip LeDuc

Abstract Grammars are useful for representing systems of design patterns, however formulating good grammars is not straightforward in many contexts due to challenges of representation and scope. This challenge has been identified as one of the 3 goals for computerized use of shape grammars: grammar inference. This work introduces a highly flexible mechanism for inducing viable general grammars from a computational representation of a designed context. This mechanism is evaluated with several common types of devised media of increasing complexity based on dimensionality: 1D (e.g., text), 2D (e.g., PCB layout, building plans), many dimensional (which in abstract can generally be used to represent product, system, platform or service designs), and, against a set of grammar properties necessary for a grammar acquisition method to be useful: accuracy, variability, repeatability and conciseness. This work shows complete enumeration over possible grammars in the 1D case and a continuum of approaches for higher dimension data sets that are demonstrative of grammars in design.

Introduction

Formally, grammars define the scope of content and structure that constitutes a language. People use grammars naturally through human language but also strictly when communicating with computers and other computational systems. Examples of the latter can be very simple, such as the reverse polish notation (Ball 1978), a syntax used to efficiently input arithmetic to a calculator by listing all operands of an expression before their operator. But they can also be more complex, such as parsing programming languages, or even extremely complicated, such as elements that formally convey subtleties of human design, like Shape Grammars (Gips and Stiny 1972; Stiny 1980) and Graph Grammars (Ehrig et al. 1973).

M. Whiting · J. Cagan (✉) · P. LeDuc
Carnegie Mellon University, Pittsburgh, PA, USA
e-mail: cagan@cmu.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_15

Statements in programing languages generally have unique interpretations for a given language, for example “for i in range(10): print(i)”, is always interpreted in the same way by python. On the other hand, Design representation, either geometrical (in shape grammars) or ontological (in graph grammars), tend to not be uniquely interpretable (Benrós et al. 2015). This is exacerbated by the fact that designs are often made by humans, and not perfectly logical systems, who sometimes unwittingly incorporate internal inconsistencies of rule interpretation, for example inconsistent execution of a brand identity. In other words, humans face similar challenges adhering to a formal grammar when designing as they do when communicating with spoken languages. This is noticeable in languages like English where the growth of the language has led to new spelling rules as well as some surprising grammatical divergences. Similarly, development of a design grammar often requires adding more rules as special cases are introduced. A shape grammar for a house can not include a bathtub until a specific rule detailing a bathtub is added to the grammar. Furthermore, this is true, even if the house only needs one instance of the bathtub.

The difference between rote knowledge of a grammar and native speaker familiarity with it is stark. Although poetry generally exemplifies the valid use of a formal language grammar, it is also an example of a higher level of cultural use of the grammar, that is not easy to explicitly document. As a result, native speakers can capture this with significantly more ease and repeatability than formal lingual grammars do. This contrast is relevant with grammars used in design too. For example, although it is possible to explicitly define a shape grammar based on an existing design in order to make more designs of a similar style, it is often hard to know when there are enough rules to be able to reliably use the grammar as a generative tool that will effectively demonstrate the style. Human designers, on the other hand, can easily mimic the design style directly. One aspect of this challenge is that many grammars are over simplified because factors they aim to represent are too subtle to be incorporated manually. For example, the grammar mentioned above that now has a rule for a bathtub, but the rule creator may not bother to add rules for specifically locating the feet of the tub; yet these need to be located precisely to make the tub functional. Formally, this challenge involves inducing latent grammatical traits of the natural grammar (this can be thought of as the human designer’s copycat design), and representing those in the formal shape grammar (an optimized, complete, and rule based design).

Bridging between natural grammars and formal grammars has also been a focus in automated language translation (Ding and Martha 2005). Probabilistic Grammars are a schema used when specific grammatical representations can’t be computed so a field of representations is used with probability weightings (Stolcke and Omohundro 1994). These approaches afford induction of grammars from representative datasets in very specific contexts, but do not effectively support inducing grammars from generalized data; in other words, they do not resolve situations with no syntactical coding. As a result, they are not particularly suitable for making grammars in design more available or powerful at this time, because they would require input data with some structural coding, but may become useful as more dynamic approaches emerge. An example of this is demonstrated by Talton et al. (2012) showing that with coded design data, machine translation methods can be effectively used to generate grammars.

Other efforts to conduct automated grammar inference have been discussed since early in shape grammar literature (Evans 1971), but there remain no complete solutions. In addition to the reasons mentioned above, generating grammars computationally is difficult because it is computationally challenging, it requires assessing many possible rules, and because there are not precise objectives or heuristics for determining the single best grammar out of a range of possible grammars (Königseder and Kristina 2015; Gips 1999). As a result, high-level sketches for shape grammar automation have been proposed (Gips 1999), and some parts of the process have been automated with statistically generated rules (Talton et al. 2012; Orsborn et al. 2008). However, these do not sufficiently solve the issue for arbitrary problems, which is the underlying challenge in inducing general grammars for design.

The aim of this work is to introduce an approach for generating general grammars through induction from arbitrary, pattern driven design data, one of the three potential tasks for applying computation to grammars as identified by Gips (1999). To do this, a means for evaluating grammar completeness and efficiency is established, and a method for inducing lexical grammars is introduced. This method is extended for more complex data types, of the same type as those used in grammars for design for things such as layout problems, and product, service and platform style design problems. Both methods are then assessed on real and manufactured data sets.

Grammar Metrics

In design, a grammar's effectiveness can be judged first by its ability to communicate style, and then by its ability to do so in an optimal way, from a computational performance perspective. As a mechanism for conveying style, there are at least 3 critical previsions for effectiveness, adapted from Stiny and Mitchell (1978): (1) it should clearly convey the common elements of stylistically similar artifacts, (2) it should be complete enough to facilitate determining if an artifact is within a style or not, and (3) it should be usable to generate new artifacts that adhere to the style but were not part of the style corpus.

The *accuracy* of a grammar, defined as how well the grammar affords representing the input materials used to establish it, is intended to evaluate Stiny and Mitchell's first point above. However, quality of representation is somewhat binary with non-probabilistic grammars because there is no margin of error—it's either correct, or not correct—so for the sake of this work, accuracy is assumed to be a requirement of any grammar. Without this feature, the grammar is not even reliable enough to reproduce its own input and it should be rejected.

The *variability* of a grammar, on the other hand, interpreting Stiny and Mitchell's third point, is defined as how well a given grammar can be used to generate new examples of the style or construction rules embedded in the grammar. Again, with non-probabilistic grammars, a grammar either offers variability, or does not, so this will be considered a necessary condition for a grammar to be accepted.

Another aspect of a grammar system is the expectation that a similar grammar could be achieved from different observed artifacts. A practical example of this is if one builds a lingual grammar for English from two different instances of the New York Times Newspaper, there would be some hope that the contents of the two grammars would be similar and functionally interchangeable. We term this as the *repeatability* of a grammar, or the likelihood that grammars A and B will be identical when grammar B is learnt based on material generated from grammar A. In practice, grammar variability means that most data can be represented by many different valid grammars, so it is better to require that there is a grammar B in the set of grammars learnt from material generated by grammar A, such that grammar A and B are functionally identical.

Repeatability is not exactly the same as Stiny and Mitchell's second point, because for repeatability it is assumed that grammars of two different artifacts are comparable, as opposed to determining the degree to which one grammar serves two different artifacts. However, these two approaches end up being computationally similar in the worst case, because applying a grammar to an existing artifact is approximately as hard as developing a new grammar from that artifact. This work does not prove that claim, but it was found in practice, and since our definition of repeatability is stricter than Stiny and Mitchell's second point, that has been the preferred metric.

Computational complexity of grammars is a well studied challenge (Slisenko 1982; Yue and Krishnamurti 2013), and determining if two different grammars can have identical output, with only a difference in complexity is nontrivial. Prioritizing *conciseness* in generated grammars can be established by adhering to the information axiom found in Axiomatic Design (Suh 1990); if two designs are otherwise equal, choose the simpler one. When learning grammars, and after ensuring they have accuracy, variability and repeatability, the next priority is to establish that the selected grammar is the most simple. In practice, this is convenient for computational complexity but also because it implies that more salient information is stored per grammar rule, so arguably, it can demonstrate more nuance in alternative outputs.

Together, accuracy, variability, repeatability and conciseness offer a complete metric for computational optimality as well as effective communication of style. Achieving the first two is a necessary condition for a grammar to be considered effective, then the latter two offer helpful insight when deciding which grammar representation best suits a given data set. These will be used as key metrics for determining the effectiveness of grammar induction methods outlined below.

Inducing Grammars

To induce a grammar is to use the content of a sample dataset and inductive reasoning to derive what grammar rules define that dataset. A stark example of this is trying to learn the structural rules of a foreign language simply by parsing text in that language. It is not an impossible task, but one at which humans are not very good. An algorithm, on the other hand, can be made to tackle this problem with precision and tireless sampling.

Text can be classified as data with one dimension of structural order that follow some set of grammar rules and has some standardized segmentation symbol, usually a space or punctuation. In this work, data of this type is described as 1D, for one dimension of connection. A token in this form of data is a nuclear unit of the data, such as a word in text or a data object in a more complex dataset. Grammar rules are replacement rules in which one specific token can be replaced by one or more other specific tokens to implement the rule.

Our algorithm, detailed in Fig. 1, begins by ranking the set of segments on the token level. The algorithm is rank schema agnostic but for the purposes of introducing the method, token frequency will be considered directly proportional to a token's rank. The token ranking is used to determine a higher level segmentation, to initiate a recursive process and generate rules that represent the grammar. Recursion

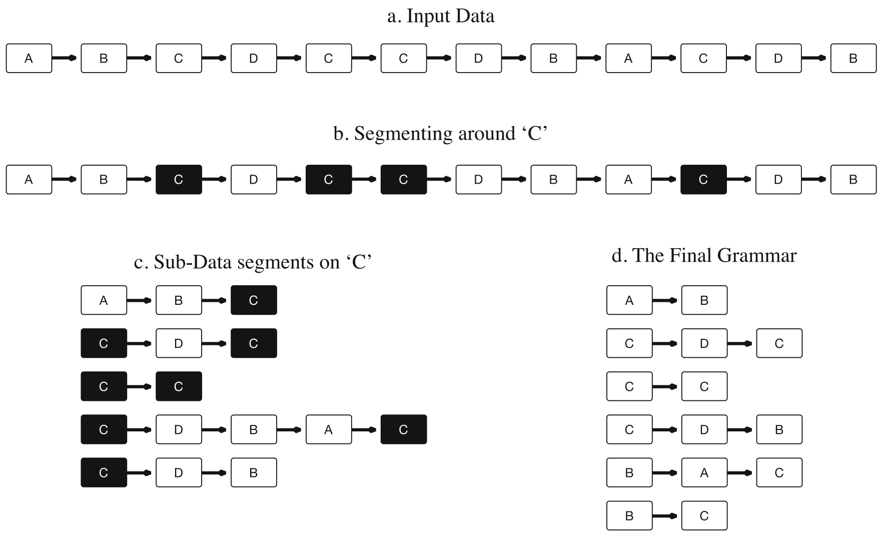


Fig. 1 The algorithm being used on one dimensionally connected data for the first segmentation level. **a** Input data is parsed. **b** Data is segmented around a specific repeated token, in this case 'C'. **c** Sub-data segments are formed from the segmentation. **d** After recursive runs with all tokens (not shown), a final grammar is achieved

1. Parse dataset into tokens.
2. Rank tokens using specified ranking heuristic.
3. While sub-datasets are larger than a grammar rule size constant:
    1. Split the dataset into smaller sub-datasets at each instance of the highest remaining ranked token.
    2. Remove the token just used from the ranking.
4. Return set of sub-datasets as grammar rules.

Fig. 2 Pseudocode of the 1D grammar induction algorithm. Each token of data is assumed to be connected to only 2 others, one before it, and one after it, and there is assumed to be an implicit ordering governed by those connections, as is the case with text in English

can incorporate the entire set of unique tokens, or can be stopped earlier, depending on how granular the grammar representation is required to be for a specific application.

Tokens appearing only once in a dataset are treated as singletons. To make these usable, it is necessary for their final tokens to include the previous least common token grouping, so they are clustered and incorporate relevant positional information to be useable in the grammar.

As shown in Fig. 2, intuitively, the data are broken into parts based on most frequently repeated elements, these parts are then broken into smaller parts based on the second most frequently repeated elements. This goes on until no repeated elements remain. The eventual result of this process is a set of element clusters, that when applied in the right order, can reproduce the input data exactly.

By using alternative rankings, an exhaustive set of valid grammars can be generated for a given dataset without any other awareness about the dataset. However, in practice, measurable properties of tokens are generally more practical for ranking. In other words, token clusters of any pairing rule can be the final rules after generating a grammar, just by using different token ranking mechanisms. This brings a lot of flexibility, but means that choosing the appropriate way to do ranking is an important step and has a critical influence on which grammars are most useful for specific situations.

Working with Interconnected Data

The most common forms of data with one dimension of connection are text, audio and video. In design there are often more dimensions of connection, for example both content and position play a key role in designing the layout of a graphic work, this might be considered two dimensional (2D). Products and service systems can incorporate even more dimensions of interconnection (nD) to represent their physical, functional, temporal and cognitive complexities. However, once a design

1. Parse graph into tokens.
2. Rank tokens using specified ranking heuristic.
3. Enumerate unique paths through the graph as datasets
4. While sub-datasets are larger than a grammar rule size constant:
    1. Split the dataset into smaller sub-datasets at each instance of the highest remaining ranked token.
    2. Remove the token just used from the ranking.
    3. If any resulting sub-dataset has been previously processed, discard it.
5. Return set of sub-dataset units as grammar rules.

Fig. 3 Pseudocode of the nD grammar induction algorithm. All tokens are assumed to have some connection to other tokens in the graph, but no other assumptions are made about how tokens are connected

is represented by a graph that is more complicated than just a list, it can't be assessed quite the same way. Although not all design issues can be expressed as graphs, shape grammars and graph grammars are both formalisms which can be. For this reason, this work assumes that separate algorithmic approaches are only necessary for one dimension and more than one dimensions of connection, so only the algorithms in Fig. 2, for one dimension, and Fig. 3, for more dimensions, are demonstrated.

With the same terminology used in the one dimensional case, it is possible to treat data with more dimensions of interconnection as systems of tokens that are connected in a node graph made up of tokens with interconnections as edges, the nD case. Just as was done in the one dimensional case, rankings can be used to find critical points about which to segment the graph, however, the graph can be segmented in more than one way at each node. Because how the graph is segmented drastically influences which strings of tokens are parsed, it also greatly impacts which grammar rules are formed, independent of the ranking used. This is problematic for the sake of comparison and repeatability.

By enumerating paths through the data graph, all alternative interpretations of the data are incorporated and separately evaluated. The algorithm used in the 1D case is then applied to each path and a grammar is then devised as the set of rules from all the rules derived in this process. Figure 3 shows the nD algorithm while Fig. 4 shows an example graph with 100 nodes and 10 unique tokens, labeled with letters 'A–J'. This example was generated with a grammar with a relatively small number of rules, one being that letters datasets could exist in their labeled alphabetical order, 'ABCDEFGHIJ'. Figure 5 shows the rules used to generate the sample data in Fig. 4, and also shows rules induced from that data using this algorithm.

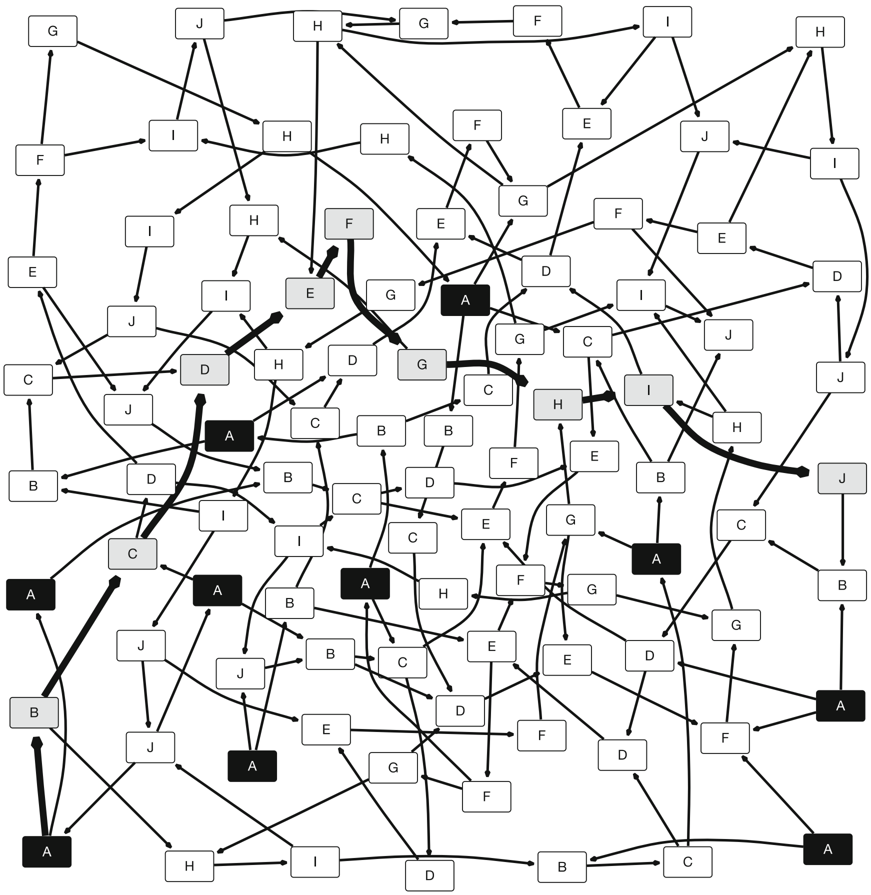


Fig. 4 Sample experimental nD graph with 100 data objects of 10 different types labeled with *letters A–J* and ‘A’ objects are darkened to demonstrate that a simple segmentation rule is not easy to implement in a graph. 9 other objects are greyed with stronger edge markings to show a possible tour of part of the graph that highlights one of the grammar rules used to generate this data, objects ordered alphabetically by title, ABCDEFGHIJ

Because many paths through the data could have similarities, many of the sub-datasets could be identical between paths. As a result, at any point that segmentation leads to a sub-dataset that has already been induced by the algorithm, it can be discarded and the algorithm can move onto the next step. This process helps reduce the processing time of the algorithm and reduces it significantly with respect to the number of edges in the graph.

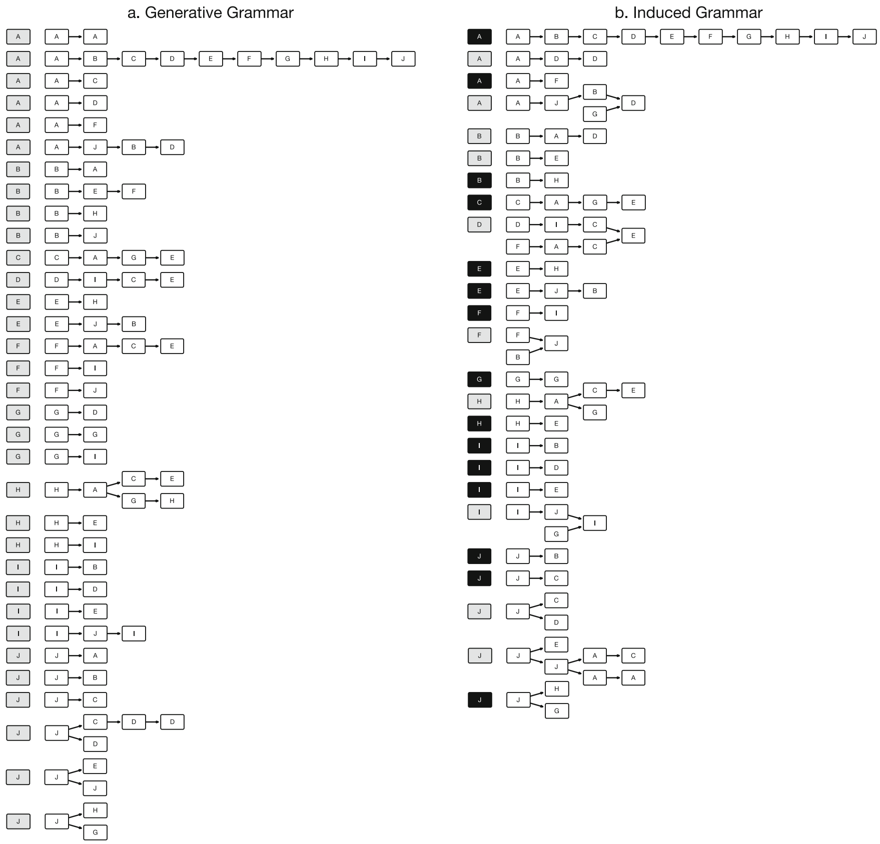


Fig. 5 **a** The grammar used to generate the graph in Fig. 4, made up of 33 rules. **b** A grammar induced from the graph in Fig. 4 using the grammar induction algorithm resulting in 25 rules. 60% of the rules in the induced grammar are identical (marked with *black* starting token) to ones in the generated grammar

Trial Data and Evaluation

Evaluations were conducted with two types of data: (1) artificial grammatically defined data objects of arbitrary complexity for 1D and nD cases, (2) randomly selected text snippets from the Corpus of Contemporary American English (Davies 2008) with special characters removed for 1D cases only. Without reliable reference results to compare to, it is difficult to make strong claims about grammar accuracy with corpus data. By comparing artificial data with corpus data it was possible to verify matching effectiveness between the two. Randomly generated grammar rules were used to build the artificial data sets, as opposed to purely random data, to ensure that there were sufficient patterns for the grammar induction algorithm to find. Figure 5 shows an example of the grammar rules used to generate the example graph in Fig. 4, as well as an output grammar from the grammar induction algorithm.

Trials were run with small (n = 100), medium (n = 10,000) and large (n = 1,000,000) numbers of nodes to induct into a grammar, for each of one (1D), two (2D), and many dimensional (nD) cases. These evaluation sizes were chosen to emulate problems of vastly different levels of complexity. The number of unique tokens in each case was 10% of the number of nodes in that trial. Trial data objects consisted of random strings.

The implementation was in Python 3 and interconnections were managed with the NetworkX library (2015). All trials were run locally on commodity hardware with 8 cores and in instances in which processing lasted longer than one hour, trials were cut short. One hour was used as the time-limit after it was found that giving significantly more time tended to not show different results. For example, the 2D Large case had also not finished after 24 h. Because of the computational complexity of higher dimensional cases, this meant that medium and large trials were not practical to run for nD cases.

Results

Results are reported on trial runs of generated datasets for specific complexity levels in Table 1. Reported values are the percent of resulting grammars that express accuracy, variability and repeatability while reported values for conciseness are the

Table 1 Grammar metrics

Trial	Accuracy (% achieved)	Variability (% achieved)	Repeatability (% achieved)	Conciseness (% rules/n)
Real World	100	100	80	50
1D Small	100	100	100	50
1D Medium	100	100	100	50
1D Large	100	100	100	50
2D Small	100	100	50	83.5
2D Medium	100	100	50	91.7
nD Small	100	100	10	98.2

Evaluations with runtime over 1 h omitted. Accuracy and variability are necessary conditions. Repeatability is desired at 100% and conciseness, reported as a percentage ratio of grammar tokens to nodes, has an expected arithmetic asymptote of approximately 50%

percentage of the ratio of rules to total data-points, which agrees with computational bounds.

As a necessary condition of any grammar to be selected was that it would achieve accuracy and variability, all evaluations reported achieved those goals. Unlisted evaluations such as 2D Large were too complex to achieve results within the time limit, but there is no evidence that they would not converge given enough time.

Repeatability varied over the course of experimental cycles. In the trial data cases, it varied proportionally to connectedness with higher dimensionality leading to significantly fewer repeatable grammars. This is assumed to be the case because selecting for concise grammars may sometimes lead to certain grammar features being represented more than once. As is mentioned in Talton et al. (2012), it can be very difficult to determine the difference between two grammars for the same data set. For the nD Small case this is not a problem but for larger cases this might mean that it becomes almost impossible to rely on only searching for unique rules over all resulting grammars when the algorithm is run.

Conciseness performed best on one dimensional cases but showed significantly less favorable results in more complex evaluations. This is attributable to the number of plausible and actual node connected pairs increasing significantly with more dimensionality. As a result, however, it means that the inducted grammars are too large to be useful and are likely to include many rules that are near duplicates of each other. Removing these is nontrivial and would make the current approach take orders of magnitude longer to compute.

Conclusion

Inducing a general grammar defining a dataset is difficult in that it requires gaining a sense of the content and the structure of the data simultaneously. This work focused on studying how structural rules can be learnt with a straightforward algorithm and evaluated its performance with 4 grammar metrics: accuracy, variability, repeatability and conciseness. Results were promising for every metric with simple grammar induction problems, however, when dealing with highly connected datasets, grammar conciseness and computational complexity exceeded a computation threshold leading to reduced value. The algorithm would take much longer than an hour to complete and the size of the grammar became close to the number of tokens in the set trial. This method is able to induce a compelling grammar for a wide variety of structured datasets, including those found in traditional uses of grammars in design. However, it is not yet practically usable for many of the more sophisticated examples of these because of the very large number of possible grammars to calculate and the amount of time computing all those options takes.

Immediate areas for potential refinement include better measures of conciseness, refined bounds for the high dimensional cases, and more nuanced exploration of rankings to improve graph segmentation. Additionally, further experimentation with this approach as computational performance is improved will offer insight into its strengths and weaknesses for design.

This work has introduced a first step in automating the use of grammars. By considering multiple dimensions of interconnection, it has also addressed many of the issues in automatically using grammars for design. With more effort in this area, grammars could become widely used for generative design and as an analytical tool.

References

Ball JA (1978) Algorithms for RPN calculators, 1st edn. Wiley, Cambridge. ISBN 0-471-03070-8

Benrós D, Eloy S, Duarte JP (2015) Re-inventing ceramic tiles: using shape grammars as a generative method and the impact on design methodology. In: Proceedings of the 16th International Conference CAAD Futures, pp 467–480

Davies M (2008) The corpus of contemporary American English: 450 million words, 1990–present. http://corpus.byu.edu/coca/

Ding Y, Martha P (2005) Machine translation using probabilistic synchronous dependency insertion grammars. In: Proceedings of the 43rd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics

Ehrig H, Michael P, Hans JS (1973) Graph-grammars: an algebraic approach. In: IEEE conference record of 14th annual symposium on switching and automata theory, 1973. SWAT'08. IEEE

Evans TG (1971) Grammatical inference techniques in pattern analysis. Softw Eng 2:183–202

Gips J, Stiny G (1972) Shape grammars and the generative specification of painting and sculpture. In: Freiman CV (ed) Information processing, vol 71. North-Holland, Amsterdam, pp 1460–1465

Gips J (1999) Computer implementation of shape grammars. In: NSF/MIT workshop on shape computation, vol 55

Königseder C, Kristina S (2015) A method for visualizing the relations between grammar rules, performance objectives and search space exploration in grammar-based computational design synthesis. In: ASME 2015 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers

NetworkX (2015) networks.org

Orsborn S, Cagan J, Boatwright P (2008) A methodology for creating a statistically derived shape grammar composed of non-obvious shape chunks. Res Eng Des 18(4):163–180

Slisenko AO (1982) Context-free grammars as a tool for describing polynomial-time subclass of hard problems. Inf Process Lett 14(2):52–56

Stiny G (1980) Introduction to shape and shape grammars. Environ Plann B 7:343–351

Stiny G, Mitchell WJ (1978) The Palladian grammar. Environ Plann B 5:5–18

Stolcke A, Omohundro S (1994) Inducing probabilistic grammars by Bayesian model merging. Grammatical inference and applications. Springer, Berlin, pp 106–118

Suh NP (1990) The principles of design. New York: Oxford University Press. pp 147–188

Talton J et al (2012) Learning design patterns with bayesian grammar induction. In: Proceedings of the 25th annual ACM symposium on user interface software and technology. ACM

Yue K, Krishnamurti R (2013) Tractable shape grammars. Environ Plan 40(4):576–594

Generative Shape Design Using 3D Spatial Grammars, Simulation and Optimization

Luca Zimmermann, Tian Chen and Kristina Shea

Abstract Advancements in 3D printers are challenging engineers to design ever more complex, customizable and unique products. This paper presents a method that facilitates design by combining 3D spatial grammars, structural simulation and optimization. The proposed method is generic and illustrated here through the design of wheels for inline skates since they have both aesthetic and functional requirements. A new spatial grammar for wheel spoke design is described that integrates constraints from additive manufacture such that the wheels can be directly fabricated. Next, the necessary adjustments to enable automated FE simulation with invariant boundary conditions are shown. The design selection process during generation is driven by simulated annealing optimization and a spatial grammar specific neighborhood definition is introduced for shape modification. Results presented for the case of the inline skate wheel show promise both in automatically generating many different yet valid concepts and in obtaining a structurally optimized design.

Introduction

Technological advancement in the Additive Manufacturing (AM) offers new opportunities for engineers to design and fabricate complex structural and mechanical products (Stanković et al. 2015). It has also been recognized that manual design with current CAD technology is inadequate in leveraging the possibilities offered. In particular, the geometric complexity is both difficult to envision in a manual design process and difficult to model in CAD (Gibson et al. 2010).

While the benefits of a human design process are numerous, human designers have some difficulties that may lead a design to be suboptimal, such as design fixation (Purcell and Gero 1996) and confirmation bias (Hallihan et al. 2012). These challenges lead the authors to automate the design conception and evaluation

L. Zimmermann (✉) · T. Chen · K. Shea
ETH Zuerich, Zurich, Switzerland
e-mail: lucaz@ethz.ch

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_16

process through a combination of a 3D spatial grammar formalism, Finite Element (FE) simulation and Simulated Annealing (SA) optimization.

Design is a process where the designer creates form and structure that possess the required behavior and function (Starling and Shea 2005). On an ad hoc basis, a human designer is able to tailor the interpretation of these requirements to the specific stages of design. A designer is also able to implicitly guide the design to a state of sufficient quality before formal evaluation begins (Antonsson and Cagan 2005). However, in automated design space exploration, the given requirements must be formulated in a way that can be understood and satisfied by the exploration method. In this paper, we focus on the size, shape and topology design of one component within an assembly and forgo the design generation of assemblies and systems.

Competing methods have been proposed in the fields of art, design, architecture, and engineering. Continuous topology optimization (Bendsøe and Kikuchi 1988) in particular offers potential to take advantage of the capabilities offered by AM (Brackett et al. 2011). Given a solid volume, the method optimizes for the distribution of mass within this volume so as to achieve a defined goal (e.g. minimum compliance) subject to defined constraints (e.g. allowable stress range) (Bendsøe and Kikuchi 1988).

While continuous topology optimization generates an optimized design given the constraints, there are several challenges to be overcome if the generated design is to be physically fabricated. Aside from the technical need for post-processing, e.g. to adhere to AM constraints, the method produces only one design for a given optimization model that the user cannot directly influence, e.g. the form. If the designer's intention is to explore the design space with requirements that cannot be quantified in terms of engineering performance, such as aesthetics, and directly incorporate AM constraints, a more suitable method is needed.

By dissociating design generation from design evaluation, one is able to use a number of methods for each, thereby allowing for much greater freedom in design. Numerous researchers have proposed methods in which the design is parametrized (Woodbury 2010). These parameters are then varied systemically using a meta-heuristic algorithm (Rutten 2010) to achieve an optimum. One inherent disadvantage of such a system is that when the number of variables increases, the algorithm may have difficulty in converging to an optimum. Conversely, with few variables, the design space becomes unnecessarily limited.

A grammar system provides an alternative to parametric modeling in generating a design. In particular, the spatial grammar formalism (Stiny 1980) is one such rule-based system for shape generation. Connecting such a system to SA optimization is explored in this work. The authors demonstrate the flexibility of both the generative system and the optimization method, and the synergy between the two when solving design challenges involving both function and form constraints.

Here, the authors expand the general approach of the 2.5D structural shape grammar proposed by Shea and Cagan (1997) to the continuous domain using a primitive-based 3D spatial grammar interpreter, spapper, presented by Hoisl and Shea (2011). Whereas Shea and Cagan proposed a method for the optimization of

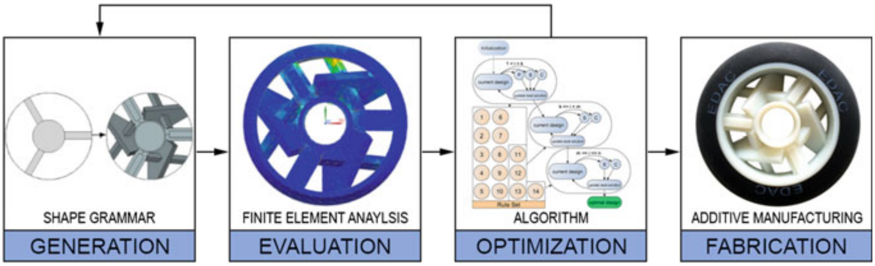


Fig. 1 Proposed process from generation to evaluation, optimization and fabrication using the design task of inline skate wheel spoke design as an example

truss structures, here a method is proposed to optimize continuous structural geometries.

In this paper, a spatial grammar is used to generate wheel design variants, the resulting designs are evaluated with structural FE simulation, and Simulated Annealing with a rule-based neighborhood definition is used for selection towards generating optimized designs. The complete procedure as shown in Fig. 1 starts from design generation, transforms the model automatically to an FE simulation model that provides evaluation for the optimization and finally the optimized designs can be directly fabricated.

The paper begins with a brief review of spatial grammars, the spatial grammar interpreter spapper and further relevant work. Then the method is described in general terms following the procedure outlined in Fig. 1. First, the spatial grammar developed for this design task and implementation is discussed. Next, the preparation of the CAD-based design model for FE analysis along with its automation is detailed. The subsequent section discusses the SA implementation when combined with the spatial grammar. Section four presents the design of an inline skate wheel, focusing on the spokes, to demonstrate the effectiveness and versatility of the proposed method for both form and function design. The paper finishes with a conclusion and future outlook.

Background

The proposed method consists of four distinct components, three of which are introduced and summarized separately in this section. The description of SA is omitted here but a comprehensive overview is given in (Dowsland and Thompson 2012).

Spatial Grammars

Stiny and Gips (1971) first introduced the shape grammar formalism in 1971. A typical shape grammar consists of four components, namely a set of shapes, rules and labels, and a starting shape. The rules and the shapes within the current working shape (CWS) are built from elements in the vocabulary. The formalism is defined as a transformation of the CWS by replacing a set of matching, in terms of geometry and spatial relations, shapes with replacement shapes according to a set of predefined rules. As stated in the introduction, the shape grammar formalism provides a more powerful alternative to parametric design, and has been shown to generate more and novel solutions in comparison (Chen and Shea 2015).

spapper—A Parametric, 3D Spatial Grammar Interpreter

To reduce the difficulty associated with implementing the spatial grammar formalism (Krishnamurti and Stouffs 1993) and thereby widening its adoption, Hoisl and Shea (2009) developed a 3D spatial grammar interpreter. As a plugin to FreeCAD, spapper links grammar development with a CAD system that offers primitive modeling. Features of spapper include rule development, 3D primitive recognition and replacement, labels and enumeration of the search space through automatic rule selection. spapper is used in this work to generate the initial and the neighboring solutions in an automated manner.

Two wheel spoke grammars have been implemented using spapper. The first was used to demonstrate the capability of spapper (Hoisl and Shea 2011), and the second attempted to generate both existing and novel designs of automotive wheel spokes. It was proposed that such a grammar system can generate designs that have more variety than using a parametric system as well as incorporate constraints from AM so that the designs can be automatically fabricated (Chen and Shea 2015). In these examples, limited quantitative evaluation took place.

Combining a Spatial Grammar with Optimization

Combining a spatial grammar with an optimization algorithm has been investigated with success. As exemplified by Ang et al. (2006) who proposed a method for product development, a spatial grammar is used to generate the shapes, and an optimization algorithm in conjunction with a volume-based objective function is used for selection.

Shea and Cagan developed a spatial grammar system that explores the design space of truss structures from 2D to 2.5D and 3D (Shea and Cagan 1997; Shea et al.

1997). With limited and reversible rules, they were able to traverse the design language with SA to generate designs that can be better than human designs in terms of optimization and novelty. This combination is coined shape annealing (Cagan and Mitchell 1993). The design evaluation involves structural simulation and is limited to 3D pin-jointed truss structures. The method itself however is unique in that topologies are grown and optimally directed, as opposed to members being removed from a ground structure (Saitou et al. 2005). In addition, it has been shown to be an effective asset to designers by expanding the designers' search space (Shea and Cagan 1999). In this paper, the concept of shape annealing is extended by utilizing a 3D spatial grammar interpreter and a FE solver with 3D brick elements, rather than truss elements.

Method

This section first presents the spatial grammar developed for the generation of inline skate wheel spokes. This serves as a concrete example of the method. The method itself however is generic so that 3D spatial grammars intended to generate other products can be used in the design synthesis. The second portion of this section describes the necessary adjustments to enable automated FE simulation and optimization.

Spatial Grammar

The rule system consists of 14 rules, 13 of which are illustrated in Fig. 2. The generative power of these rules results from two special properties. The first can be seen in the middle of Fig. 2: The left hand side (LHS) is the same shape for each rule. Although certain rules constrain the LHS box to have a minimum length, every rule is guaranteed to be applicable at any time of the generation process. Thus, the rule sequence can be classified as unrestricted (Knight 1999). The second detail is the absence of labels in the rules. Labels are introduced in spatial grammars as indicators of further, often non-spatial information, in addition to the shape. For example, if the LHS of two or more rules is the same, but they can only be applied in certain sequences or spatial relations, the designer adds a label in order to distinguish the two. The more labels a designer uses, the more control is exerted on the rule system, but the creative potential and generative power of the spatial grammar are more restricted. The concept for this wheel grammar is to let the grammar generate shapes as freely as possible, which is why labels are omitted in the rules (Table 1).

The validity of designs needs to be ensured without the use of labels in the rules. This is achieved mostly by the parameter boundaries but also by the possible selection of rules, e.g. rule 14 is only applied at the very end of a run as it can

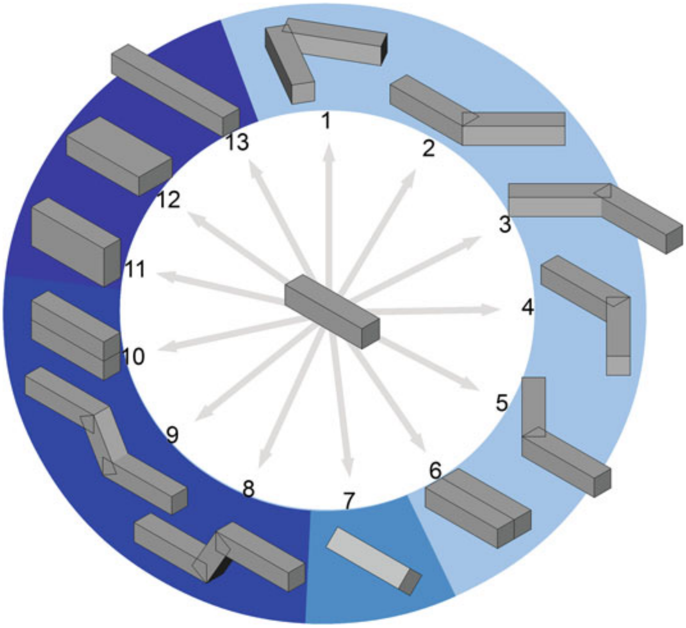


Fig. 2 Spatial grammar rules of the wheel spokes

Table 1 Description of each rule of the inline skating wheel grammar

Set A: In-plane rules, rotation only around the axis of the wheel	
1	Duplicates a box, rotates the resulting boxes by an angle
2[a]	Adds to the end of the box a new box rotated counter-clockwise
3[a]	Adds to the beginning of the box a new box rotated counter-clockwise
4[a]	Same as rule 2 but rotates the new box clockwise
5[a]	Same as rule 3 but rotates the box clockwise
6	Duplicates the box and places it adjacent to the original
Set B: In-plane rotation rule	
7	Rotates a box around its longitudinal axis
Set C: Out-of-plane rules	
8	Adds to the end of a box a pitched smaller box with an upward translated box at its end
9	Same as rule 8 but downwards
10	Same as rule 6 but in vertical direction
Set D: Sizing rules	
	Changes the height (11), width (12), and length (13) of a box
Set E: Removal rule	
14[b]	Removes one box from the design

(continued)

Table 1 (continued)

[a]	Shape detection in spapper functions by recognizing an origin (and thus a label) placed at a specific corner of each box, thereby embedding directionality. Rules 3 to 5 are therefore included to remove the dependence on this origin, as they represent all the in-plane combinations of rule 2 without any labels. Rules 7, 11, 12 and 13 are not affected because the rules do not change the location of the center of mass for a box. The same is true for rules 6 and 10, where the new box is placed adjacent to the original, but both boxes are translated so that their interface lies on the middle axis of the original box. Rules 8 and 9 do not cover all possibilities
[b]	This rule is needed as a post-processing rule, and is omitted in Fig. 2. SA may accept worse solutions, for example, boxes that add weight but do not contribute to the strength of the structure. These boxes can be filtered out at the end of the optimization process

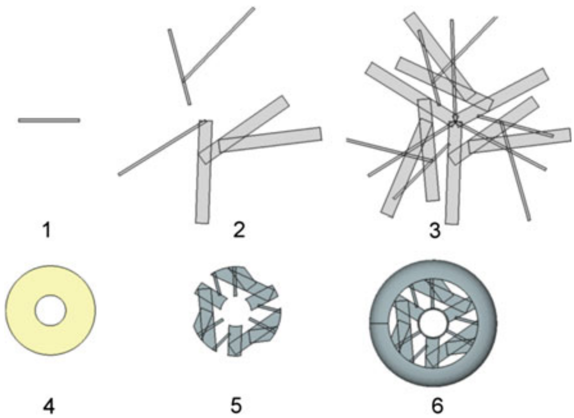


Fig. 3 Generation steps of a sample wheel

disconnect the rim and hub of the wheel. The process from the initial shape to a final design that can be evaluated numerically is shown in Fig. 3.

The initial shape is one radially placed box, to which the rules are applied from 1 to 2. Rather than having a rule that copies each box around the wheel like Hoisl and Chen (Chen and Shea 2015; Hoisl and Shea 2009), the generated pattern is replicated in the post-processing step 3, assuming rotational symmetry as an aesthetic constraint. This is done parametrically, i.e. possible numbers of copies around the wheel range from 3 to 8, and is denoted as NC. The part shown in step 4 is imported and intersected with the pattern, and the resulting geometry (5) is then united with hub and rim (6). The rule format is additive with regard to shapes (Knight 1999), but is hard to further categorize due to the absence of labels. The vocabulary consists of primitives, but the grammar could employ other vocabularies. As this grammar is not subject to any other constraints, it can be categorized as near unrestricted. As such, an extensive design space is expected.

$$N_{Sol} = N_C N_P N_{Rules} \sum_{n=1}^{N_{Appl}} n! \tag{1}$$

Equation 1 approximates how many designs can theoretically be generated. The smallest step spapper assigns to free parameters is integer size. If for example a free parameter ranges from 1.5 to 10 mm, possible values that spapper uses are $2, 3, \ldots, 10$. The wheel rules are parametric; each rule can generate approximately $NP = 1300$ different shapes on average. The number of copies around the hub N_C equals 6. Finally, on average, there are as many generated boxes as the number of rule applications (N_{Appl}). In total, for three rule applications, the design alternatives sum to approximately 900,000 designs. This would increase with the number of rule applications.

In comparison to other methods, one of the important advantages is that spatial grammars can include constraints. The authors include Design for Additive Manufacturing (DfAM) constraints within each rule that are specific to the printing process as measured by Chen et al. (2015) (Table 2).

Due to constraints (1) and (2), the minimum thickness and width of the boxes are set to 2 mm in all rules. (3) is directly enforced in rules 8 and 9, which are the only ones that generate pitched boxes, thus the angular parameter limits of these rules are set to [45° : 90°]. However, the bigger issue with these rules is not the feasibility, but the validity of the resulting designs. The rule parameters in the plane of the wheel, e.g. length of the boxes, are designed such that the hub and rim stay connected. By introducing pitched members, this constraint is likely to be violated, because any rule that disconnects the pitched middle member of rules 8 or 9 from one of the horizontal members generates an invalid design. Thus, we need to prevent the application of any topology rule to the pitched middle members. If we worked with labels, this would be the point where the pitched box is assigned a label, which is then added to the LHS of all the rules that apply to the pitched box. For every such rule, the rule creation process with labels is iterative and tedious, and with every new introduction of a label, the rule system becomes more restricted and loses generative power (Knight 1999). This issue is resolved here by setting the minimum length on the LHS of all topology rules to be greater than the maximum length of the pitched members in rules 8 and 9. The maximum length of a pitched box is 12 mm, which is why the minimum length for said rules is set to 13 mm. The

Table 2 3D printing constraints, adopted from Chen et al. (2015)

	Dimensions	Range of validity
1	Minimum thickness in horizontal direction	$w \geq 1.5\,\text{mm}$
2	Minimum thickness in vertical direction	$h \geq 0.66\,\text{mm}$
3	Angle of overhang to eliminate support material	$\theta_{OH} \geq 45°$
4	Minimum interior angle	$\alpha \geq 10°$

minimum interior angle constraint (4) is implemented as a soft constraint, meaning that the rotational limits of spokes in rules 1–5 is set to 10° or 170°, respectively.

To prevent identical designs from being analyzed multiple times, each resulting geometry is subjected to a volume check, which accepts all solutions that exhibit a different volume than its predecessor. The generated wheels can then be imported into the FE simulation.

Simulation

After a design is generated, FE simulation analyzes the design and calculates the objective value that is used by the optimization algorithm. The automation involves replacing the previously simulated design with the current output by the spatial grammar interpreter and then simulating the new design. During this process, the simulation software must ensure that boundary conditions (BCs) and the surfaces to which they are applied remain invariant when the underlying geometry changes. This is achieved by dividing the overall design into multiple smaller components, so that the boundary components remain in the simulation and only the spatial grammar output (SGO) is replaced.

This division usually requires the definition of contact parameters between the components. However, since the components are assumed to be rigidly connected, the degrees of freedom on the contacting surfaces between components are made rigid with respect to each other in the simulation model. The top diagram in Fig. 4 shows a generic, standardized simulation, in which the design is fixed on the left side and a distributed force is applied to the right side. To separate the SGO from the invariant components in the model, one must distinguish two situations. In the first case, where the contact surface of the SGO lies exactly on the surface of the invariant component, they can just be connected (A on Fig. 4 lower). The second

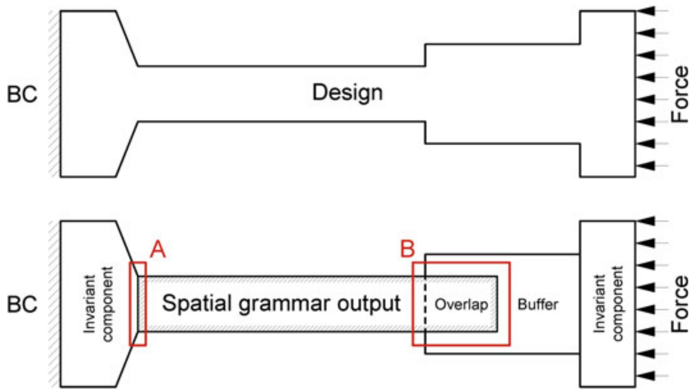


Fig. 4 Generic simulation (*upper*) and possible separation of the part (*lower*)

situation B is illustrated in Fig. 4 lower, where the surface of the SGO is either unknown or lies within the invariant volume such that there is an overlap. A buffer component remedies the situation, but requires a two-step process, because it and the SGO must be united before they can be connected to the invariant component. This second solution offers more flexibility depending on the design task.

Optimization

Spatial grammars pose unique challenges for optimization algorithms. For a general application of optimization in spatial grammar systems, a nonlinear search space is anticipated. In addition, the nature of spatial grammar often creates a discontinuous optimization task. This is a consequence of the rules' LHS/RHS nature and the transformation from one state to another by adding and removing shapes and their associated variables. Lastly, the selected optimization algorithm must be able to provide a reasonably good optimum given a solution space that is multi-modal.

Depending on the class of the spatial grammar (Knight 1999), generated search spaces can be vast. Variety typically results from a parametric rule set and not from the sheer number of applicable rules. Due to a spatial grammar's ability to generate many solutions with only a few rule applications, it would be beneficial for the selected algorithm to optimize with a breadth-prioritized strategy. Two possibilities that meet the requirements are SA and Genetic Algorithms (GAs).

It may be natural to encode the genotype of a GA as the rule application sequence, e.g. $\mathrm{bin}(135)$ would result in rules $1 \rightarrow 3 \rightarrow 5$ applied in that order. The disadvantage of such an encoding is that the values of the free parameters in a parametric rule set would be unknown. It would be necessary to update the gene after the application of each parametric rule to record the selected parameters. Second, in spatial grammars, in contrast to L-systems, a rule when executed only applies to one matching LHS shape, but not all. Therefore, to maintain a one-to-one mapping between the genotype and the phenotype, the ID of the matching shape in the CWS must also be recorded. Such genotype may increase in length prohibitively, and the implementation of genetic operations becomes overwhelmingly complex and ineffective.

SA has been employed to optimize large search spaces (Dowsland and Thompson 2012) and has been shown to work well with spatial grammars (Shea et al. 1997; Reddy and Cagan 1995). It is a robust algorithm that can be tailored easily to different optimization problems, including discrete, nonlinear and multi-modal search spaces. Therefore, SA is chosen as a suitable optimization algorithm for this task.

Two parts of the SA algorithm need special attention in their implementation, namely the definition of a neighboring solution and state dependent rule sets. The requirements of a neighbor are that the next solution is close in proximity, but far enough that all possible solutions within the entire space can be reached. To support a general application of the method, we define the neighborhood as follows.

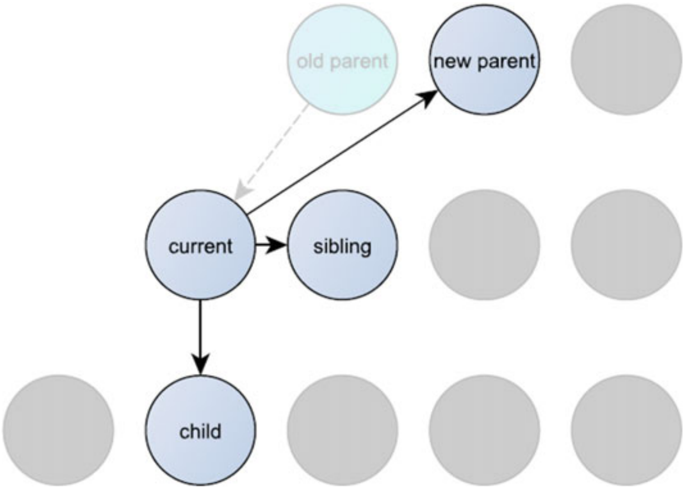


Fig. 5 Neighborhood definition for spatial grammar based SA

We proceed with the above-mentioned rule sequence $1 \rightarrow 3 \rightarrow 5$, which is denoted as "current" solution in Fig. 5. There are three possible neighbors: a parent solution, a sibling or a child. A child solution would mean an additional rule is applied to the current solution, e.g. $1 \rightarrow 3 \rightarrow 5 \rightarrow \mathbf{7}$. A sibling replaces the last applied rule with a different one, e.g. $1 \rightarrow 3 \rightarrow \mathbf{4}$. A parent solution is defined by replacing the previous rule and removing the current, e.g. $1 \rightarrow \mathbf{2}$. In case of a parametric rule set, neighbors with the same rule sequence, e.g. $1 \rightarrow \mathbf{3}$ or $1 \rightarrow 3 \rightarrow \mathbf{5}$, could generate a new solution as the parameters may change. If this is not the case, then the choice needs to be restricted so that the same rule is not reselected. This generic neighborhood definition can be applied to parametric or nonparametric rule sets, and to grammar systems ranging from basic to unrestricted (Knight 1999). The distinction between different rule sets, e.g. topological and sizing rules, poses a challenge in combining grammar systems and optimization (Königseder and Shea 2014).

Deciding which rules to apply in which stage of the optimization is a delicate operation, and the best strategy differs from task to task. Therefore, it is important to allow designers to influence such a strategy. Figure 6 depicts the rule structure using the inline skating wheels as an example. In the first stage of the algorithm, topology and sizing rules (1–13) are applicable, whereas the rule set is reduced to sizing rules only (11–13) in the second stage. In the final stage, the removal rule is applied.

With SA, the neighborhood definition may change at different stages of the algorithm as well. "P" (parent), "S" (sibling) and "C" (child) in Fig. 6 can be tailored to the designer's needs; if the sizing rules in the second stage are reversible, there would be no need to include parental neighbors (sibling neighbors may also

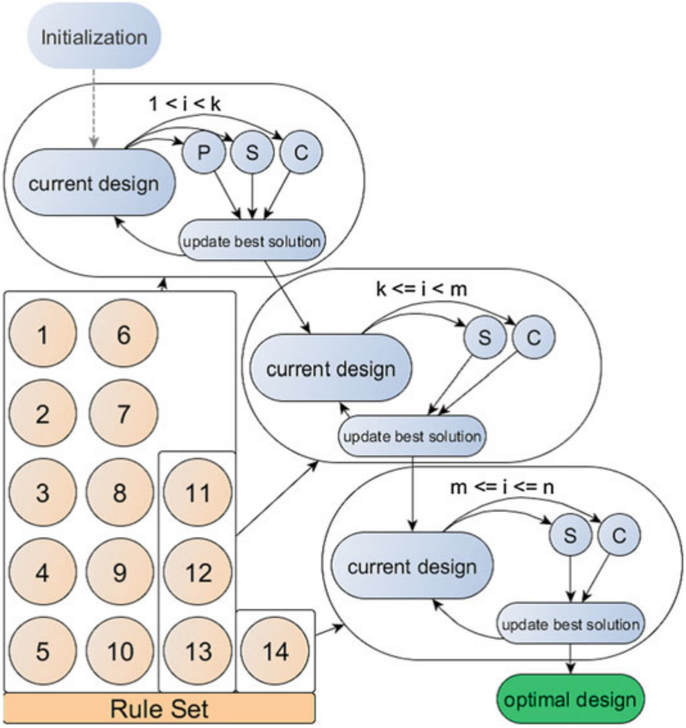


Fig. 6 Possible stage structure for different rules and neighbors

be omitted, but enabling them keeps the solution concise). Neighbors, rules and stages can be arranged in any combination.

Wheels of Inline Speed Skates

The proposed method is applied to the task of designing wheels of inline skates. These wheels provide a good example of a product that requires both engineering performance and aesthetic appeal as well as a good example for the state-of-the-art 3D printing with multiple materials used in the same part. The “tire” can be printed with a compliant elastomer, whereas the spokes use a stiff plastic (Boyer 2000). The printed specimen can be tested for structural integrity and performance.

For inline speed skates, the wheels are ≥100 mm in diameter and 24 mm across. The wheel designs, focusing on the spokes, are generated using spapper in FreeCAD and are then sent to Siemens NX 8.5 for FE simulation. The load case is adapted from car wheel simulations (Giger and Ermanni 2005; König 2004; Stearns et al. 2006) and ice speed skating (De Boer et al. 1986, 1987), which was shown to exhibit the same load case as inline skating (De Boer et al. 1987). Each wheel is simulated with linear FEA (both material and geometry) four times to guarantee structural soundness for left and right cornering, and is completed in 2 min on an Intel Xeon E31245 with 8 GB RAM. The geometry is meshed with unstructured tetrahedrons to provide flexibility. Mesh density is calculated automatically and the average size of an element is 5 mm.

Pre-process and FE Simulation

This section shows the necessary division of the wheel for the FEA simulation, specifies the objective function and the parameter definition for SA, and finally discusses the results of the process.

The wheel is divided into four parts (Fig. 7, A. hub, B. SGO, C. rim, D. tire). The SGO (B) is connected to the hub on the inside and the rim on the outside. Conventional wheels have a stiff rim beneath the compliant tire (Boyer 2000), hence these components are separated from each other. Because the SGO is generated through intersection, its boundaries are predefined and thus do not need a buffer part for either the hub or the rim. Hub, rim and tire are always present in the simulation file, and the SGO is replaced at each iteration and united with hub and rim. The load case is shown in Fig. 8. The inner surface of the hub (A) is restrained from translation and rotation. Two different load cases are defined to account for both straight skating and cornering. In both cases the force is applied on the outer surface of the tire where ground forces attack, but its direction and magnitude differ. A 100 kg skater is assumed; this mass is distributed onto four wheels evenly, resulting in 250 N of nominal force. A motion factor amplifies the load depending on the load case (De Koning et al. 1987). The distribution of the force into its vertical and horizontal components depend on the angle of contact θ_E (De Boer et al. 1986, 1987) (Table 3).

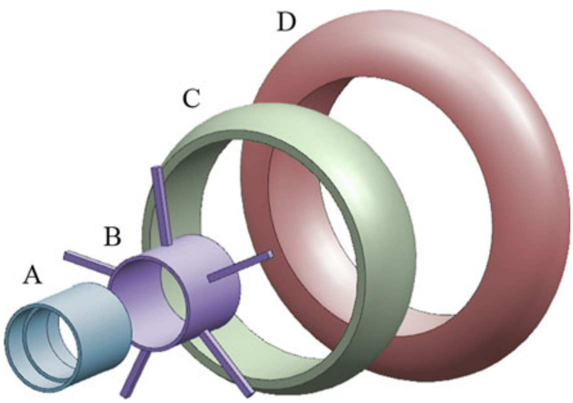


Fig. 7 Division of the roller blade wheel in multiple components

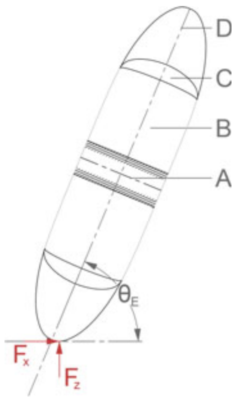


Fig. 8 Schematics of load and boundary conditions

Table 3 Summary of applied loads

	Straight	Cornering
Nominal force	250 N	250 N
Motion factor	1.50	1.05
F_{design}	375 N	263 N
$F_{\text{design,X}}$	218 N	173 N
$F_{\text{design,Z}}$	305 N	198 N
θ_{E}	35.5°	41.2°

Simulated Annealing

As mentioned before, the SA structure used here is shown in Fig. 6. Since the optimization takes place in two major stages (the third stage with rule 14 does not need a separate definition), the objective function definition and the parameters change between the stages. A safety factor of 4 is applied (Ehrenstein and Erhard 1984; Krishnamachari 2002; Mascarenhas et al. 2004), leading to a maximum allowable stress of $\sigma_{\text{adm}} = 10.6\,\text{MPa}$. This constraint is implemented through an exterior penalty method with a penalty factor of 1×10^7. Both the stress and volume are normalized, where $V_{\max} = 81,505\,\text{mm}^3$ is the volume of a solid wheel (Table 4).

$$\sigma_{obj} = \frac{\sigma}{\sigma_{adm}}; \quad V_{obj} = \frac{V}{V_{max}} \tag{2}$$

$$p = \begin{cases} \sigma_{obj} & if\, \sigma_{obj} \leq 1 \\ \left(\sigma_{obj} - 1\right) \times 10^7 & if\, \sigma_{obj} > 1 \end{cases} \tag{3}$$

Table 4 Objective and SA parameters

Stage	Objective function	SA parameters
1: Topology and sizing	$\Omega_1 = p + \gamma V_{\mathrm{obj}}$ Where $\gamma = 20$	$\mathrm{T}_0 = 10^8$ Fraction = 0.85 Number of iterations = 800 Inner loop size = 39
2: Sizing	$\Omega_2 = V_{\mathrm{obj}}$	$\mathrm{T}_0 = 0.25$ Fraction = 0.80 Number of iterations = 200 Inner loop size = 24

Hence, the formulation of the optimization is the following:

$$\text{Minimize } \Omega \tag{4}$$

$$\mathit{Subject\ to}\ \sigma \leq 10.6\,\mathit{MPa} \tag{5}$$

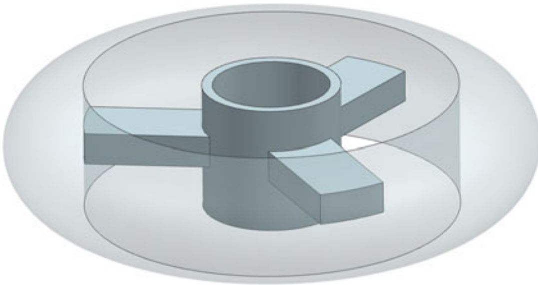
The multiplier γ in the first objective function equalizes the influence of the volume and the stress objectives on the algorithm. $\gamma = 20$ is determined through heuristics. The inclusion of the stress function p in the first objective function stems from the additive nature of the spatial grammar. Without the stress, after a feasible solution is found, only volume subtraction would guarantee acceptance, which would go against the additive rule set.

The temperature schedule follows an exponential reduction, with starting temperatures of $T_0 = 10^8$ for stage one and $T_0 = 0.25$ for stage two, which in both cases is the maximum possible change in the objective; this follows a standard implementation (Dowsland and Thompson 2012). The number of moves within the inner loop is 39 in the first stage, equal to the number of all neighbors for one solution (13 rules for parents, siblings and children), offering the possibility for the algorithm to test every possible neighbor. Since the sizing stage includes only 200 iterations, a faster decrease rate in comparison to the first stage is needed. The three sizing rules need more trials per rule relative to the topology stage, which is set to eight, resulting in 24 inner loop moves. The displacement does not need to be included in the objective function because it is proportional to the stress and does not exceed a maximum of 1.5 mm for feasible solutions.

Results

The above SA optimization is performed 20 times. The best found solution for these 20 runs is shown in Fig. 9 and its volume is 6.42% of a fully filled wheel. The design with the highest objective function value has a volume fraction of 7.96%, the mean volume fraction over all runs is 7.16% and the standard deviation is 0.38 %.

Fig. 9 Solution with the best objective function value in 20 runs



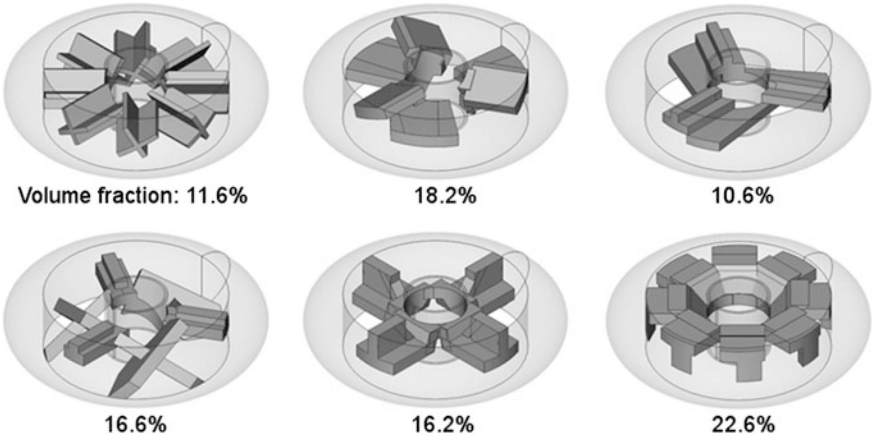


Fig. 10 Collection of selected wheel designs

The best wheels from the other 19 runs follow the same design strategy, all featuring simple, radially placed spokes. The best solution is shown to obtain the same shape as a continuous topology optimized wheel with the same load case and volume fraction. Another important outcome of the developed method is the variety that is generated. Figure 10 shows a small collection of chosen designs to illustrate the unique designs that arise from the proposed method. The shown wheels all fulfill the stated requirements and may have more aesthetic appeal and capitalize on the capabilities of AM.

The last step of the proposed method is the printing of the designed wheels. A chosen design is selected and fabricated on a Stratasys Polyjet 500 Connex 3. The spokes, hub and rim are printed with VeroWhite+, the tire is printed with FLX9870. The wheels are attached to a pair of skates and the assembly is tested both statically and in motion (Fig. 11).

Fig. 11 Selected wheel printed and assembled onto the skates

Discussion

The authors demonstrate a method that systematically explores the design space in an optimally directed way. As a framework, it yields the possibility for a designer to exchange any of the four components (generation, analysis, optimization and fabrication) for others independently.

In the wheel example, the solution that is optimized for structural performance alone may not be what the designer wants. In contrast to a direct optimization method such as continuous topology optimization, in addition to the optimized design, the designer is also provided with a range of 'good' designs that may have other desirable qualities such as aesthetics. With consumer products, the ability to generate a variety of functional designs can be invaluable.

The biggest time investment for a designer to create such a system is the creation of the spatial grammar and the set-up of the FE simulation. There currently exists no automatic generation of spatial grammar rules, although this has been investigated in research in the past (Schnier and Gero 1996), but it is possible to improve the simulation automation. A designer would still have to simulate once, but could then specify the components to replace, the boundaries and possible buffer parts in a program that translates this for the method.

One future step is to add aesthetics and spatial uniqueness as an objective as opposed to a beneficial side effect of using a stochastic optimization method. Optimizing for minimum volume and constraining stress guarantees functional solutions, but interesting designs are found rather as by-products.

Conclusion

In this work, we propose a design method that is able to generate a suite of structurally optimized designs for a product with geometry that may be difficult to conceive by human designers in a comparable amount of time and that is feasible to fabricate with Additive Manufacturing. The method integrates a spatial grammar interpreter with FE simulation and stochastic optimization for general structural forms composed of primitives. Label-free spatial grammar rules with constraints encoded in each rule, some of which embed AM constraints, are used to generate alternative designs for an inline skate wheel. FEA is performed on each design and the process is automated through the definition of invariant boundary geometries. A modified Simulated Annealing routine is used to drive the process towards optimally directed designs. In particular, a neighborhood definition tailored to the spatial rule application process is proposed to explore the design space. The results for an inline skate wheel that can be directly printed on a multi-material 3D printer illustrate that the approach can generate optimized designs for structural performance and, as a result of using a stochastic optimization method, a variety of alternative, feasible designs that can be selected for their aesthetics.

References

Antonsson EK, Cagan J (2005) Formal engineering design synthesis. Cambridge University Press, Cambridge

Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224

Boyer G (2000) Multi durometer wheel for in-line skates. Google Patents

Brackett D, Ashcroft I, Hague R (2011) Topology optimization for additive manufacturing. In: Proceedings of the solid freeform fabrication symposium, Austin, TX

Cagan J, Mitchell WJ (1993) Optimally directed shape generation by shape annealing. Environ Plan B 20:5

Chau MCAHH, Alison M, De Pennington ALAN (2006) Combining evolutionary algorithms and shape grammars to generate branded product design. Design Computing and Cognition’06. Springer, Netherlands, 521–539. http://link.springer.com/chapter/10.1007/978-1-4020-5131-9_27

Chen T, Shea K (2015) Computational design-to-fabrication using spatial grammars: automatically generating printable car wheel design variants. In: Proceedings of the 20th international conference on engineering design, vol 2. Design Society, Glasgow, pp 35–44

Chen T, Fritz S, Shea K (2015) Design for mass customization using additive manufacture: case-study of a balloon-powered car. In: DS 80-4 proceedings of the 20th international conference on engineering design (ICED 15), vol 4: Design for X, Design to X, Milan, Italy, 27–30 Jul 2015

De Boer RW et al (1986) Characteristic stroke mechanics of elite and trained male speed skaters. Int J Sport Biomech 2:175–185

De Boer RW et al (1987a) Biomechanical aspects of push-off techniques in speed skating the curves. Int J Sport Biomech 3(1):69–79

De Boer R et al (1987b) Physiological and biomechanical comparison of roller skating and speed skating on ice. Eur J Appl Physiol 56(5):562–569

De Koning JJ, De Boer RW, De Groot G, Van Ingen Schenau GJ, Koning D, De Boer RW et al (1987) Push-Off force in speed skating. Int J Sport Biomech 3:103–9. http://journals.humankinetics.com/doi/abs/10.1123/ijsb.3.2.103?journalCode=ijsb

Dowsland KA, Thompson JM (2012) Simulated annealing. In: Handbook of Natural Computing. Springer Berlin, Heidelberg, 1623–1655. http://link.springer.com/referenceworkentry/10.1007%2F978-3-540-92910-9_49

Ehrenstein GW, Erhard G (1984) Designing with plastics: a report on the state of the art. Hanser Publishers, Cincinnati

Gibson I, Rosen DW, Stucker B (2010) Additive manufacturing technologies. Springer, Berlin
Giger M, Ermanni P (2005) Development of CFRP racing motorcycle rims using a heuristic evolutionary algorithm approach. Struct Multidiscip Optim 30(1):54–65
Hallihan GM, Cheong H, Shu L (2012) Confirmation and cognitive bias in design cognition. In: ASME 2012 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
Hoisl F, Shea K (2009) Exploring the integration of spatial grammars and open-source CAD systems. In: DS 58-6: proceedings of ICED 09, the 17th international conference on engineering design, vol 6, Design methods and tools (pt. 2), Palo Alto, CA, USA, 24–27 Aug 2009
Hoisl F, Shea K (2011) An interactive, visual approach to developing and applying parametric three-dimensional spatial grammars. Artif Intell Eng Des Anal Manuf 25(04):333–356
Knight TW (1999) Shape grammars: six types. Environ Plan B 26:15–32
König O (2004) Evolutionary design optimization: tools and applications. Eidgenössische Technische Hochschule [ETH], Zürich. http://e-collection.library.ethz.ch/view/eth:27401
Königseder C, Shea K (2014) Strategies for topologic and parametric rule application in automated design synthesis using graph grammars. In: ASME 2014 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
Krishnamachari S (2002) Recommended factors of safety and related considerations (167). In: ANTEC-conference proceedings
Krishnamurti R, Stouffs R (1993) Spatial grammars: motivation, comparison, and new results. http://papers.cumincad.org/cgi-bin/works/Show?cb3c
Mascarenhas W, Ahrens C, Ogliari A (2004) Design criteria and safety factors for plastic components design. Mater Des 25(3):257–261
Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17(4):363–383
Reddy G, Cagan J (1995) An improved shape annealing algorithm for truss topology generation. J Mech Des 117(2A):315–321
Rutten D (2010) Evolutionary principles applied to problem solving. In: AAG10 conference, Vienna
Saitou K et al (2005) A survey of structural optimization in mechanical product development. J Comput Inf Sci Eng 5(3):214–226
Schnier T, Gero JS (1996) Learning genetic representations as alternative to hand-coded shape grammars. Artificial Intelligence in Design'96. Springer Netherlands, 39–57. http://link.springer.com/chapter/10.1007/978-94-009-0279-4_3
Shea K, Cagan J (1997) Innovative dome design: applying geodesic patterns with shape annealing. Artif Intell Eng Des Anal Manuf 11(05):379–394
Shea K, Cagan J (1999) The design of novel roof trusses with shape annealing: assessing the ability of a computational method in aiding structural designers with varying design intent. Des Stud 20(1):3–23
Shea K, Cagan J, Fenves SJ (1997) A shape annealing approach to optimal truss design with dynamic grouping of members. J Mech Des 119(3):388–394
Stanković T et al (2015) A generalized optimality criteria method for optimization of additively manufactured multimaterial lattice structures. J Mech Des 137(11):111405
Starling AC, Shea K (2005) A parallel grammar for simulation-driven mechanical design synthesis. In: ASME 2005 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
Stearns J et al (2006) Understanding the influence of pressure and radial loads on stress and displacement response of a rotating body: the automobile wheel. Int J Rotat Mach. https://www.hindawi.com/journals/ijrm/2006/060193/abs/
Stiny G (1980) Introduction to shape and shape grammars. Environ Plann B 7(3):343–351
Stiny G, Gips J (1971) Shape grammars and the generative specification of painting and sculpture. In: IFIP congress, no. 2
Woodbury R (2010) Elements of parametric design. ISBN-13: 978-0415779876, ISBN-10: 0415779871

Part V
Design Cognition—Design Behaviors

Comparing Two Approaches to Studying Communications in Team Design

Hao Jiang and John S. Gero

Abstract This paper explores intragroup communication in team design using data collected from a protocol study. Two units of analysis are introduced, (1) at a coarse level: turn-taking of utterances during conversations, and (2) at a fine level: design issues on the basis of the FBS ontologically-based coding scheme. These basic elements of team design activities (i.e., conversational turns and design issues) are then interconnected using Goldschmidt's Linkography method. The proposed two methods are demonstrated and compared through a case study of product design meeting. Results indicate that, for the purpose of structure-based analysis, the measurements derived from the turn-taking model are able to largely resemble the measurements derived from the FBS-based model, though the former model could be achieved with much less labor than the latter model. However, content-based analysis could only be conducted by using the more sophisticated FBS-based method.

Introduction

To deal with the increasing complexity of contemporary design problems, the design professions have moved from mainly an individual activity towards predominantly a team-based activity. Studying team design activities, particularly verbal communications and interactions between designers as well as other stakeholders, therefore becomes one of the important emerging interests of design research. For example, the 7th Design Thinking Research Symposium (DTRS7) "analysing design meetings" (McDonnell and Lloyd 2009) and the 10th Design Thinking Research Symposium (DTRS10) "analysing design review conversa-

H. Jiang (✉)
Zhejiang University, Hangzhou, Zhejiang, China
e-mail: 0012171@zju.edu.cn

J.S. Gero
University of North Carolina at Charlotte, Charlotte, NC, USA

J.S. Gero
George Mason University, Fairfax, VA, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_17

tions" (Adams and Siddiqui 2013) both focus on multidisciplinary design collaborations through conversation.

The shift from individual's work to teamwork introduces new dimensions beyond a simple increase of the number of participants. Olson et al. (1992) found that in various small team design sessions only 40% of the time on average was spent on direct discussion of the design. Social issues, e.g., information sharing, collective learning and cognitive consensus, need to be explicitly considered when studying team design activities (Mohammed and Dumville 2001).

This paper focuses on detailed analyses of verbal communications of a small design team during one or a few design sessions. This is a typical situation for the use of the protocol analysis methodology. Many studies have adapted protocol analysis methods to explore shared mental models of the design team (Bierhals et al. 2007), multimodal communication channels (Eris et al. 2014), and some other team design issues. However, the majority of current cognitive studies into team design activities relied on the research paradigms developed from the cognitive studies of individual designers. They tended to consider the design team as a whole and primarily focused on the overall performance of the teams, rather than through the lens of communication between multiple participants. A more appropriate approach for modeling team design activities is required to facilitate a better exploration of the nature of team design.

This paper reviews several communication models and methods that attempt to elucidate the structure of communication during a team-based activity. It then synthesizes the merits of these methods and proposes two approaches to studying communications in team design. These two methods are then demonstrated and compared in a product design meeting case study using the DTRS7 common dataset.

Studying Communications in Team Design

Modeling Communication as Conversational Turn-Taking

Verbal communication provides a primary means for interpersonal information exchanges among team members (Krauss and Fussell 1996). To become a valid form of communication or team design activity, a necessary condition is the involvement of more than one participant. Turn-taking of conversations is a fundamental feature of interpersonal communication infrastructure (Stivers et al. 2009; Schegloff 2007). To a large extent, information is sequentially transmitted from one participant of communication to another with minimal time gaps or overlaps. Sacks et al. (1978) classic turn-taking model captured the structure of communication, rather than its contents. It was selected to form the basis for our initial approach to modeling team design communication. Irrespective of the content of utterances, conversational turns are then used as basic elements for a simple team communication model. We further encode conversational turns with the roles of designers, and explore the communication patterns from a social perspective.

Gero and Kan (2009) demonstrated this turn-taking model in a case study of an engineering design meeting. They applied first-order Markov chain analysis to explore the interpersonal interactions between adjacent conversational turns. The transitional trend between turns revealed the formation of sub-teams during the team design activities.

Using Linkography to Connect Turns Beyond Adjacency

Gero and Kan's (2009) turn-taking model is a syntactic model, based on the assumption that a successive design action has potential connection to its immediately preceding one. In reality, the conceptual dependency of a design action exists beyond adjacent ones (Suwa and Tversky 1997). A semantic modeling approach should construct the connection between two design actions on the basis of their semantic relationship, rather than the adjacency in terms of temporal relationship.

Linkography, developed by Goldschmidt (1990, 2014), is an example of semantic modeling methods. It delineates the structure of the designing process in terms of a network of nodes and links. The basic element of Goldschmidt's Linkography is called "design moves", i.e., an incremental change of the design status (Goldschmidt 1995). The links among design moves are constructed on the basis of critical assessments of the semantic meaning contained in those moves (Goldschmidt 1990, 2014). It enables constructing connections beyond adjacent moves and also allows one move to connect with more than one other move, or to have no connection at all (isolated or "orphan" move). Several measurements (e.g., critical moves, and link density) have been developed to analyze different properties of the designing process using this approach.

Though not developed for the purpose of analyzing team design activities, the idea of the designing process as a network of interconnected design actions is valuable for modeling team design activities. We thus integrate Goldschmidt's (1990, 2014) Linkography with Gero and Kan's (2009) turn-taking model: on the basis of their semantic meaning, a conversational turn could be conceptually linked to another turn which has been taken either immediately preceding it or anywhere in the past.

FBS-Ontologically-Enhanced Linkography

The sequence of conversational turn-taking could be seen as a structure of communication that provides an environment for the acts of designing. A social-linguistic turn-taking model specifies that a turn may consist of many turn-constructional units

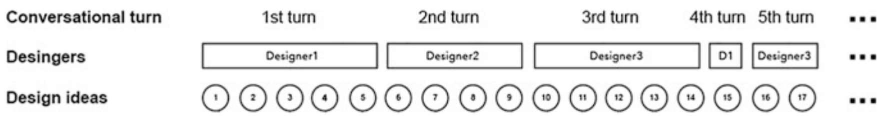


Fig. 1 A sequence of conversational turns and design ideas in a team design process

(e.g., sentences or clauses) (Sacks et al. 1978; Ford and Thompson 1996). A designer often generates more than one piece of design information in a conversational turn. Information items could thus be seen a unit of analysis nested in the conversational turn, Fig. 1.

The design information exchanged during team communication could be coded and examined in a variety of ways. The Function-Behavior-Structure (FBS) ontology (Gero 1990; Gero and Kannengiesser 2004) and the corresponding coding scheme (Gero et al. 2011) provide a validated means to categorize design information, independent of specific contexts in which designing takes place. The FBS ontology describes the designing process in terms of function (F) of a designed object (its teleology); behavior (B) of that object (either derived [Bs] or expected [Be] from the structure), and structure (S) that represents the components of an object and their compositional relationships. These ontological classes are augmented by requirements (R) that come from outside the designer and description (D) that is the document of any aspect of designing, Fig. 2.

Kan and Gero (2009) proposed a Linkography study based on the FBS-based coding scheme. The integration of Linkography method and categorization of design ideas should produce a richer understanding of design cognition. We consider conversational turns as a superimposed structure to design issues. The Linkography of design issues could also apply to examine relationships between designers.

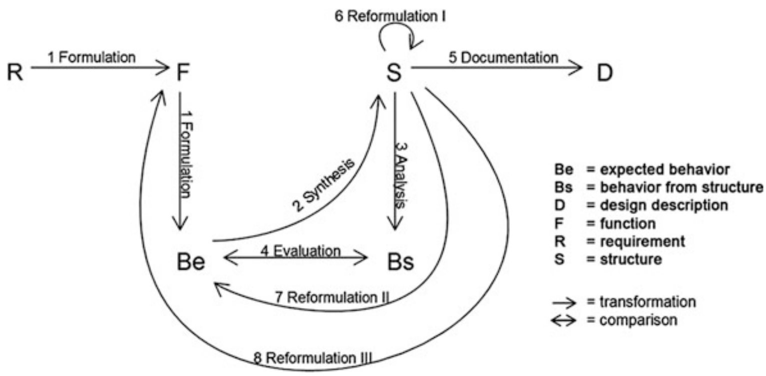


Fig. 2 The FBS ontology with the resultants design processes delineated as transitions between the ontological constructs (after Gero and Kannengiesser 2004)

Two Linkography-Based Methods of Examining Communication in Team Design

Based on a simple turn-taking model, Linkography and the FBS-ontologically based coding scheme, we propose two approaches to studying communication in team design. They aim to cover major structure and content-based analyses. Both methods adopt Goldschmidt's Linkography to reveal the structure of designing process. The distinction is in the unit of analysis. The two methods respectively use conversational turns and design issues as basic building blocks. Figure 3 shows a graphical representations of communication in team design when applied with our methods.

Linkography of conversational turns is a simpler method. Determining conversational turns from a transcript of team design is straightforward as the unit of analysis is the individual conversational utterance. The segmentation process can be untaken semi-automatically and inter-rater reliability of segmentation is sufficiently high for reliable results. As turn-taking is defined independent of utterances' semantic meaning, this unit of analysis is largely homogeneous in nature, except for the label of speakers.

The unit of analysis of FBS-based linkography is design issues. Empowered by the FBS-based coding scheme, both design issues and transitions between design issues are associated with ontologically-defined categories. A content-based analysis is thus possible using the linkograph of design issues. The benefits of Linkography of design issues come with a cost. The FBS-based segmentation and

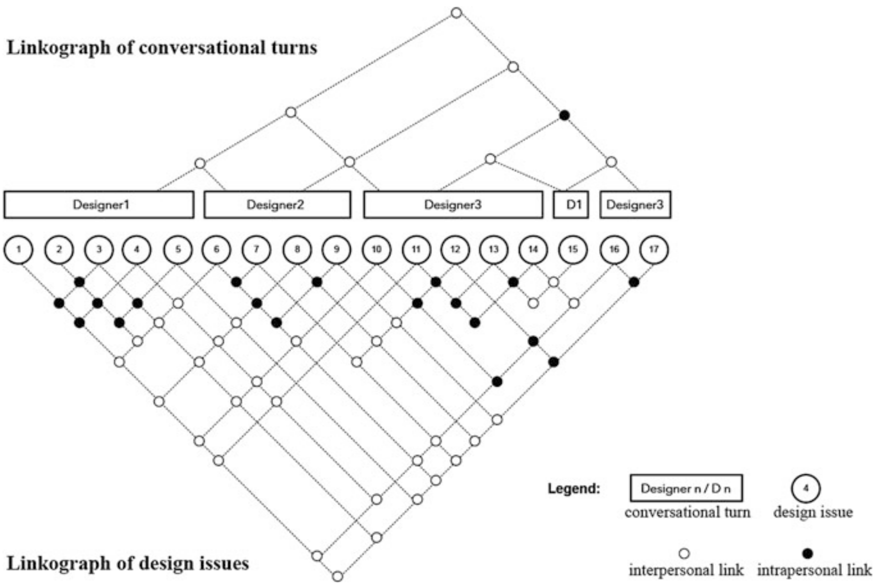


Fig. 3 Linkograph of conversational turns and linkograph of design issues

coding of designing process into design issues requires critical judgments about the scope of a design issue as well as the ontological category of individual issues. This process can be very time-consuming and labor-intensive. The Delphi method (Gero and McNeill 1998) is usually required to increase the coding reliability. A larger number of design issues, compared to the relatively smaller number of conversational turns in a same protocol, also demands more effort to construct the linkograph of design issues.

The main purpose of this paper is to examine to what extent the Linkography of conversational turns could resemble the characteristics of Linkography of design issues. If the differences between the measurements derived from these two methods are minimal, we could apply Linkography of conversational turns as a quick-and-dirty alternative to the more sophisticated Linkography model of design issues.

Case Study: Engineering Design Meeting from DTRS7

These two Linkography-based methods were performed on a seven-person team design session, using data from the DTRS7 dataset (McDonnell and Lloyd 2009). The source data was a video of design meetings taking place in a product design practice, Fig. 4. The meeting lasted approximately one and half hours. A multidisciplinary design team was asked to brainstorm ideas for solving technical

Fig. 4 Four camera digital recording of the design session

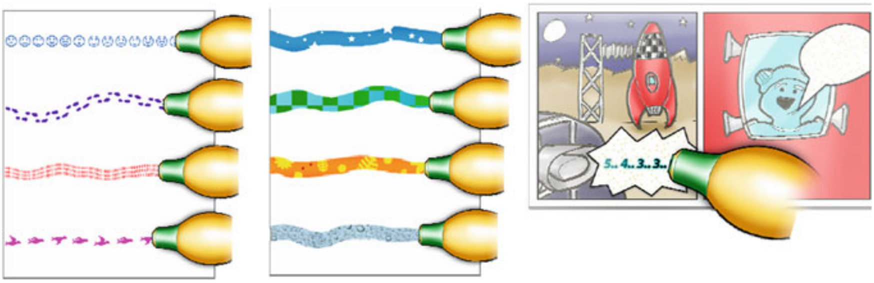


Fig. 5 Illustrations used in the session showing the function and expected behavior of the thermal printing pen being designed

issues in the design of a thermal printing pen. Figure 5 illustrates what the thermal pen was intended to do.

The team consisted of a business consultant, who acted as the moderator (Allan), three mechanical engineers (Jack, Chad and Todd), an electronics business consultant (Tommy), an ergonomist (Sandra), and an industrial design student (Rodney). They were all professionals from the same company and the student, Rodney, was on an internship with the company. They knew each other well and were efficient at collaborating. The details of the meeting setting can be found in McDonnell and Lloyd (2009).

This dataset has been examined by a number of researchers, e.g., (Gero and Kan 2009; Kan and Gero 2011; Gero et al. 2014). Previous findings are compared with current results.

Overall Comparisons Between Two Linkography Models

A simple and crude way to compare these two Linkography-based models is to visualize them in a graphic form, i.e., as a linkograph, and visually inspect the geometric patterns of these two linkographs. A linkograph can be generated by software such as LINKODER (Pourmohamadi and Gero 2011).

This is followed by overall comparison of Goldschmidt's (Goldschmidt 2014) Linkograph metrics: (1) link density, the ratio of the number of links to the number of segments (turns or design issues); (2) forward link (a transition connecting a segment to a subsequent one) and backward link (a transition connecting a segment to a previous one).

Linguistic turn-taking models focus on transitions between an adjacent pair of turns (Sacks et al. 1978; Wooffitt 2005). We then examined the proportions of adjacent turns/issues link.

Communication operates on both intrapersonal and interpersonal levels (Krauss and Fussell 1996). Figure 3 distinguishes two broad categories of interpersonal

Table 1 Coding scheme for links of Linkography of design issues

Category	Sub-category	Description
Intrapersonal link	Intra1 (immediate)	A design issue to its immediately preceding issue generated by the same participant
	Intra2 (same turn)	A design issue to an issue that is generated by the same participant in the same conversational turn, but not immediately adjacent
	Intra3 (cross turn)	A design issue to an issue that is generated by the same participant, but in a different conversational turn
Interpersonal link	Inter1 (immediate)	A design issue to its immediately preceding issue generated by a different participant
	Inter2 (adjacent turn)	A design issue to an issue that is not immediately preceding in the issue level, but is in the immediately preceding turn
	Inter3 (cross turn)	A design issue to an issue that is generated by a different participant, and beyond an adjacent conversational turn

links and intrapersonal links. A high percentage of interpersonal links would indicate a high interchange of opinions among collaborative designers. A high percentage of intrapersonal links may indicate a less successful team efforts, as each designer tends to concentrate on their own ideas and considers other participants' contributions less.

A more detailed categorization was imposed onto the Linkography of design issues, Table 1. This coding scheme takes the relative positions of two issues into consideration. It distinguishes conceptual dependency between recent turns/issues and those between remote turns/issues. For the Linkography of conversational turns, *Intra 1* and *Intra 2* types are not applicable, and *Inter 1* and *Inter 2* links will collapse into a single sub-category.

Transitional Analysis Between Designers

Dynamics between participants are key to communication. A more detailed comparison looked into the patterns of information exchange between designers, through examining transitions between designers. A transition between designers is unidirectional, from a designer with a forward link connecting to another with a backward link. We first examine whether some designers were more likely to initiate a transition and some others were more likely to follow team members' lead. This is done by a Pearson's Chi-square test of independence, comparing overall frequency distributions of designers before and after a transition.

Contingency tables were then generated for each Linkography model. Designers with a forward link are listed in rows, and the corresponding designers with a backward link in columns. The cells are frequency of transitions from designer with a forward link to designer with a backward link. Contingency tables help to reveal whether transitional relationship of designers are reciprocal, i.e. whether the probability of designer A transiting to designer B is statistically different from the probability of designer B to designer A. This can be done using the Bowker procedure (Koppel and Berenson 2006; May and Johnson 2001), a symmetry test of contingency table, which examines changes in the categorical responses before and after some treatment condition.

The similarity between contingency tables from the two Linkography models is then compared by a regression vector coefficient, R_V coefficient (Abdi 2006). It is a generalization of the Pearson's product-moment correlation coefficient for two matrices.

$$\mathrm{RV}\left(W_i, W_j\right) = \frac{trace\left(W_i, W_j\right)}{\sqrt{trace(W_i, W_i) \cdot trace\left(W_j, W_j\right)}}$$

where $trace\left(W_i, W_j\right)$, $trace(W_i, W_i)$ and $trace\left(W_j, W_j\right)$ are respectively generalized covariance coefficient between matrices W_i and W_j, and generalized variances for matrices W_i and W_j. The R_V coefficient takes values between 0 and +1. The closer R_V is to 1, the more similar are the two contingency tables. The significance test of R_V was conducted by Monte Carlo resampling (number of permutations = 5000). Statistical software XLSTAT (version 2015) was used for calculation.

If the above comparisons reveal these two models are similar, this will indicate that a model produced with significantly lower effort will suffice to reveal the structure of communication in team design. Should that be the case we will continue to examine any additional benefit when using Linkography of design issues, i.e., what benefits the labor-intensive FBS-coding will provide above the simpler turn-taking model.

Results and Discussion

Protocol Segmentation and Coding

Turn-Taking Model

The segmentation and coding of design protocols resulted in 1279 design issues and 563 conversational turns taken during this team design session. Table 2 shows the frequencies of conversational turns. Participants were not equally active in terms of how many times they addressed their options. Allan, the moderator of this meeting, contributed the most conversational turns, approximately one quarter of total turns. Todd (a mechanical engineer) and Tommy (an electronics business consultant)

Table 2 Frequency distribution of conversational turns

Designer	Role	Count	%
Allan	Moderator and business consultant	137	24.5
Chad	Mechanical engineer	57	10.1
Jack	Mechanical engineer	88	15.6
Todd	Mechanical engineer	117	20.7
Rodney	Industrial designer (intern)	27	4.8
Sandra	Ergonomist	32	5.7
Tommy	Electronics business consultant	105	18.6
Total	–	563	100.0

generated 21 and 19% of total turns respectively. The ergonomist (Sandra), and the student intern (Rodney) were relatively inactive during the conversations.

Design Issue Model

The frequencies of design issues, when applied with the FBS-based protocol segmentation and coding, are presented in Table 3. It shows that, for the categories of design issues, the structure issues were most discussed, representing 40% of total design issues. It is followed by two behavioral issues (expected behavior and behavior from structure), together representing approximately one half of total issues. Design description and function issues constitute the rest 10% of total issues. No requirement issue was identified in this session. It was thus removed from the following analyses.

Conversational turn can be seen as a superordinate unit of analysis to design issues. It is possible to calculate the frequencies of design issues embedded in the unit of analysis of designers. Table 4 shows that one conversational turn, on average (mean), contains 2.3 design issues (SD = 2.5). The number of design issues per turn is highly left-skewed. Approximately three quarters of the turns contain only one or two issues. There are only 12 conversational turns containing more than 10 design issue per turn. The longest turn is generated by the engineer Jack when he was elaborating the structure in the late phase of the session. This turn contains 30

Table 3 Frequency distribution of design issue

Design issue	Count	%
Requirement	0	0.0
Function	47	3.7
Expected behavior	275	21.5
Behavior from structure	369	28.8
Structure	512	40.0
Description	76	6.0
Total	1279	100.0

Table 4 Frequencies of design issue per designer

Designer	Frequencies of design issues		Frequencies of design issues per designer	
	Count	%	Mean	SD
Allan	307	24.0	2.24	2.40
Chad	129	10.1	2.26	2.42
Jack	225	17.6	2.56	3.51
Todd	248	19.4	1.93	1.38
Rodney	52	4.1	1.56	1.19
Sandra	50	3.9	2.12	2.06
Tommy	268	21.0	2.55	2.64
Total	1279	100.0	2.27	2.51

design issues (19 structure issues, 10 behavior from structure issues and 1 expected behavior issue). The second longest turn drops to 16 issues, when the moderator Allan addressed the beginning of the session (10 expected behavior issues, 3 structure issues, 2 behavior from structure issues and 1 function issue).

Though designers varied their involvement in the group design activities (measured in the number of conversational turns and design issues), a Kruskal-Walis one-way ANOVA shows the median numbers of design issue per conversational turn were not statistically different among designers, $\chi^2(6) = 6.47$, $p = 0.37$.

Comparison of the Overall Distributions of Designers

Pearson's Chi-square test for independence was performed to compare the overall frequency distributions of designers between the Linkography of conversational turns and Linkography of design issues. Results indicate the distribution differences between two models were not statistically significant, $\chi^2(6) = 5.56$, $p = 0.47$, Cramer's V = 0.055. We could use the measurement of either model to indicate designers' relative involvement during the team design communication.

Construction of Linkography

When applying the Linkography technique to interrelating segmented conversational turns and design issues, it generates 1879 links for the linkograph of conversational turns (Fig. 6), and 5088 links for the linkograph of design issues (Fig. 7). By presenting these two graphs to match their lengths and then overlapping them, the geometric features of these two graphs were visually similar.

Using Goldschmidt (2014) Linkography metrics, it was found that though total numbers of links varied, the link densities of two models were close, 3.34 and 3.98

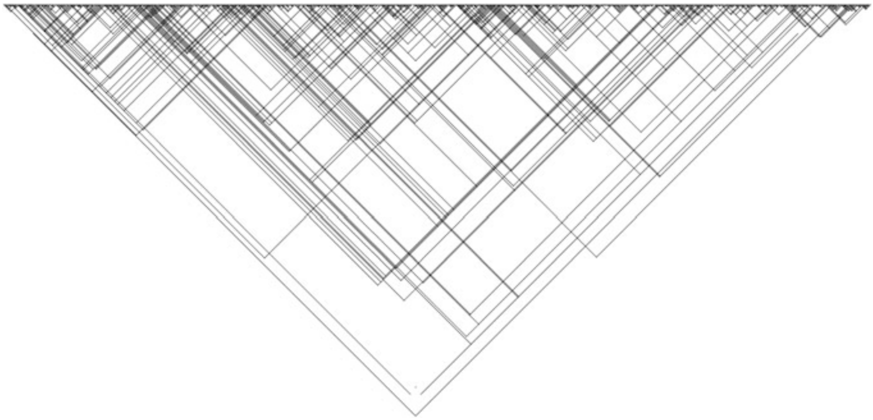


Fig. 6 Linkograph of conversational turns (generated by LINKODER)

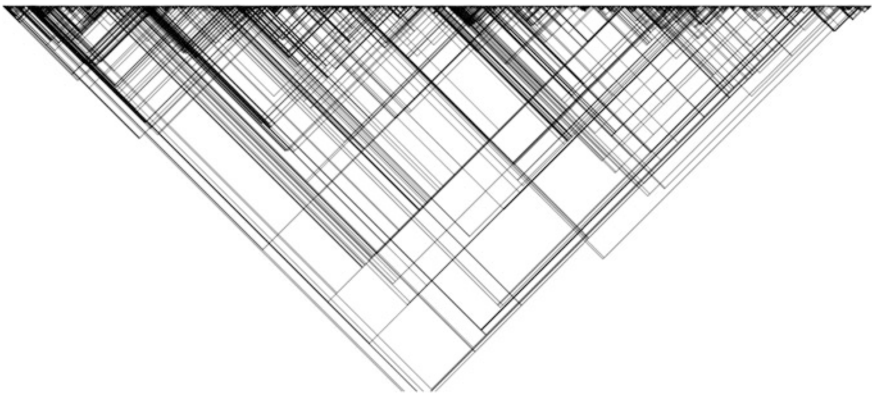


Fig. 7 Linkograph of design issues (generated by LINKODER)

Table 5 Summary of Linkography

Link measurement	Linkograph of conversational turns	Linkograph of design issues
Number of links	1879	5088
Link density	3.34	3.98
Link span (range)	1–514	1–1178
Link span (median)	4	4

respectively, Table 5. The distance between the two linked segments (link span) varied. A link may connect two immediately occurring segments, or connect a segment located nearer the beginning of the session and the other nearer the end of the session. The latter case happens when the team wrapped up the discussion and corresponded to the agendas set in the beginning of team design session. In general, the distributions of link spans were highly right skewed. The medians of link span were 4 for both linkograph of conversational turns and linkograph of design issues.

Directionality of Links

Each node (a conversational turn or design issue) of a linkograph may be associated with a forward link (a link connecting to a consequential node), a backward link (a link connecting with a preceding node), or both. Table 6 shows that the majority of nodes contain both forward and backward links, 91.5% for the linkograph of conversational turns and 92.4% for the linkograph of design issues. The frequency distributions of these three link types were not statistically different between two models, χ^2 (2) = 3.00, p = 0.23, Cramer's V = 0.040 (Table 6).

Adjacent Links

Syntactic models, which only consider the transition between adjacent pairs, are widely used in studies into designing processes, e.g., (Gero et al. 2014; Stempfle and Badke-Schaub 2002). Here 480 transitions between adjacent turns and 1100 transitions between adjacent design issues were identified. Applying the syntactic modeling approach, there are 562 links for the turn-taking model and 1278 links for the design issue model. Even though these syntactic links were constructed unrelated to the semantic meaning in the segments, the majority of these links were valid in the semantic modeling, 85.4 and 86.1%, Table 7. The proportion of adjacent turn links in the total number of semantic links was slightly higher than the proportion of adjacent issue links, 25.6 versus 21.6%. It shows the Linkography of design issues may enable more elaborate analysis of conceptual dependency beyond adjacent design issues.

Intrapersonal and Interpersonal Links

The Linkography of conversational turns compresses intrapersonal links within a turn, the intrapersonal percentage therefore would be lower than its counterpart of Linkography of design issues. A correction should be made by only considering

Table 6 Forward and backward links associate to the segments

Type of link	Linkograph of conversational turns		Linkograph of design issues	
	Count	%	Count	%
Bi-direction	515	91.5	1182	92.4
Forward only	9	1.6	30	2.3
Backward only	39	6.9	67	5.2
Total nodes	653	100	1279	100

Table 7 Transition between adjacent pairs of turns/design issues

	Linkography of conversational turns	Linkography of design issues
No. of adjacent links	480	1100
No. of syntactic links	562	1278
Adjacent link/syntactic links (%)	85.4	86.1
Adjacent link/total semantic link (%)	25.6	21.6

cross-turn links when comparing the relative frequency of intrapersonal links. There are seven participants in this team design session. If the transitional probabilities from one person to another are equal, the percentage of intrapersonal links is approximately 14.3% (1/7) of cross-turn links. The actual intrapersonal link percentage is 28.6%, almost twice of the hypothetically evenly distributed probability. This implies that, when designers refer to their previously generated design ideas, they are more likely to revisit their own ideas rather than their partners'.

For the Linkography of design issues, the frequency of intrapersonal links was 44.9% of total links, more than triple of this hypothetically evenly distributed probability (14.3%). It shows designers not only revisit their own ideas more often in terms of conversational turns, they also elaborate more of their own ideas than other participants' ideas.

Transitional Analysis Between Designers

A Pearson's Chi-square test of independence of the distribution comparison of forward links and backward links was conducted, Table 8. There was no statistical difference found in the Linkography of conversational turns, $p = 0.58$. The test was statistically significant for the Linkography of design issues, $p < 0.01$, but the small Cramer's V value (<0.1) indicated the difference was trivial.

Table 8 Comparison of nodes with forward and backward links

Statistic method	Linkography of conversational turns	Linkography of design issues
Pearson's Chi-square test of independence		
χ^2	4.72	19.35
df	6	6
p	0.580	0.004
Cramer's V	0.035	0.044
Bowker procedure		
B	17.50	59.10
df	21	21
p	0.680	0.000

The Bowker procedure showed that transitions in the Linkography of conversational turns were reciprocal, $p = 0.68$. A significant result was found in the Linkography of design issues, $p < 0.001$. But post hoc tests (Koppel and Berenson 2006) failed to detect specific discrepancies. We thus argue that transitions in the Linkography of design issues were essentially reciprocal.

The similarity of transitions between the two Linkography-based models was then quantitatively assessed using the R_V coefficient. Results confirmed that two transition matrices were strongly similar, $R_V = 0.85$, $p < 0.001$.

Comparing Linkography of Conversational Turns and Linkography of Design Issues

On the basis of the above comparisons, the characteristics of the Linkography of conversational turns, to a large extent, were similar to the Linkography of design issues in terms of the structure-based analysis. This implies that we could apply the Linkography of conversational turns as a relatively economic tool to explore structural patterns of communication in team design, e.g., relationships between designers.

On the other hand, the hierarchical feature of the Linkography of design issues enable us to explore the team communication beyond the turn-taking structure, and tap into semantic information exchanged during communication. For example, a richer categorization of links (Table 9) can be applied to explore the nuanced dynamics of team design communication. The transition between design issues can be automatically coded as eight ontologically design processes (Fig. 2), providing access to semantic aspects of concept developments. The following example illustrates a content-based analysis that can only be obtained using the more labor-intensive FBS-based method.

Exemplary Content-Based Analysis Using Linkography of Design Issues

Design issues are considered to be nested in the structure of conversational turns. We cross-tabulated the frequencies of design issues against designers, Table 10. Pearson's Chi-square test of independence shows that the frequency distribution of design issues was statistically associated with designers, χ^2 (24) = 83.80, $p < 0.001$, Cramer's V = 0.128.

A closer examination indicates the differences are related to the role of participants, using a battery of Z tests (adjust p value by Bonferroni method). The differences were mainly exhibited by the moderator Allan. He had a significantly higher percentage of expected behavior issue (29.6%) than the three mechanical

Table 9 Comparing two Linkography-based methods

	Linkography of conversational turns	Linkography of design issues
Unit of analysis	Conversational turns	Design issues
Nature of unit of analysis	Homogeneous in nature	Six ontological categories, e.g., function, expected behaviors (Fig. 2)
Segmentation and coding	• Straightforward and fast (semi-automatic) • Reliable (single coder is sufficient)	• Labor-intensive and time-consuming • Require Delphi method (Gero and McNeill 1998) to increase reliability (in this case, agreement between individual codings and final arbitrated code is approximately 84–87%)
Construction of links	Manually construct (relatively less labor-intensive, due to smaller number of turns than design issues)	Manually construct (labor-intensive, inter-ratter agreement is 80 + % in this case)
Nature of link	Homogeneous in nature	• Eight ontological categories, e.g., formulation, synthesis (Fig. 2) • Possible to categorize higher-order transitions
Intra- versus inter-personal links	No intrapersonal links within a turn	Richer categories (Table 1)
Scope of application	Structure-based analysis only	Structure-based analysis Content-based analysis

Table 10 Design issue frequencies for each designer

Designer	Design issues (%)					Problem-Solution index
	F	Be	Bs	S	D	
Allan	5.2	29.6	18.2	34.5	12.4	0.54
Chad	3.1	14.7	34.1	42.6	5.4	0.22
Jack	3.6	17.8	27.6	46.2	4.9	0.27
Todd	3.6	17.7	31.0	41.5	6.0	0.27
Rodney	0.0	15.7	45.1	39.2	0.0	0.19
Sandra	6.0	22.0	38.0	34.0	0.0	0.39
Tommy	2.6	23.1	32.8	39.6	1.9	0.35
Total	3.7	21.5	28.9	40.0	6.0	0.34

engineers (14.7–17.8%), and a lower percentage of behavior from structure issue of 18.2% than the rest of participants (32.2% in average). It seems that Allan was more focused on the design problems, and less evaluative. The problem-solution (P-S) index (Jiang et al. 2014) also shows Allan had a higher value of 0.54 than the other participants (0.30 in average). The role of the moderator may require him pulling back the discussion, from time to time, and aligning it with the set agendas. Allan also generated approximately a half of total description issues.

It is interesting to note that, there are no statistically significant differences among three mechanical engineers, Chad, Jack and Todd, $\chi^2(8) = 2.75$, $p = 0.95$. They are generally more focused on structure and behavior from structure issues. Using the FBS-based method is possible to detect how the participants perform the roles they are assigned through examining the FBS-coded communication contents as well as transitions between design issues.

Linkography of design issues, based on coding with the FBS ontology, should help to elucidate how design concepts are synthesized and evolve during communication in team design.

Conclusion

This paper proposed two Linkography-based methods of examining team design communication at two different levels of analysis units. The Linkography of conversational turns is an economic tool in term of the labor required to produce results. The relative objectivity of determining turn-taking of conversations make it much less labor-intensive than the Linkography of design issues, which requires critical assessment of the scope of a design issue and its ontological category. A series of model comparisons in a case study showed the characteristics of the Linkography of conversational turns to be similar to those of the Linkography of design issues if researchers are mainly concerned with structural patterns of communication in team design. Linkography of conversational turns should be sufficient for this structure-based analysis. But the Linkography of design issues should be selected if researchers want to extend their interests into content-based analysis. The more labor-intensive Linkography of design issues can provide deeper insights to team design communication. The hierarchical feature of model building blocks enables us to investigate both structure and content of team design communication. The selection of model should be based on the purposes of investigation and available resources.

Acknowledgements This research is supported in part by grants from the US National Science Foundation Grant Nos. CMMI–1161715 and EEC-1463873. 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

Abdi H (2006) R_V and congruence coefficients. In: Salkind NJ, Rasmussen K (eds) Encyclopedia of measurement and statistics, vol 2. Sage, Thousand Oaks, CA, pp 849–853

Adams RS, Siddiqui J (2013) Purdue DTRS—design review conversations database (TR-01-13). Retrieved from West Lafayette, IN USA

Bierhals R, Schuster I, Kohler P, Badke-Schaub P (2007) Shared mental models—linking team cognition and performance. CoDesign 3(1):75–94. doi:10.1080/15710880601170891

Eris O, Martelaro N, Badke-Schaub P (2014) A comparative analysis of multimodal communication during design sketching in co-located and distributed environments. Des Stud 35 (6):559–592. doi:10.1016/j.destud.2014.04.002

Ford CE, Thompson SA (1996) Interactional units in conversation: syntactic, intonational, and pragmatic resources for the management of turn. In: Ochs E, Schegloff EA, Thompson SA (eds) Interaction and grammar. Cambridge University Press, Cambridge, pp 134–184

Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11 (4):26–36

Gero JS, Kan JWT (2009) Learning to collaborate during team designing: some preliminary results from measurement-based tools. Paper presented at the ICoRD'09—international conference on research into design, national science seminar complex, Indian Institute of Science, Bangalore, India

Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Des Stud 25(4):373–391. doi:10.1016/j.destud.2003.10.010

Gero JS, McNeill T (1998) An approach to the analysis of design protocols. Des Stud 19(1):21–61. doi:10.1016/S0142-694X(97)00015-X

Gero JS, Kan JWT, Pourmohamadi M (2011) Analysing design protocols: development of methods and tools. In: Chakrabarti A (ed) Research into design. Research Publishing, Bangalore, pp 3–10

Gero JS, Kan JWT, Jiang H (2014) Distinguishing and measuring "Communication While Designing" and "Design Communication" in a design team. Paper presented at the international conference on human behavior in design (HBiD), Ascona, Switzerland

Goldschmidt G (1990) Linkography: assessing design productivity. Paper presented at the cybernetics and systems '90: 10th European meeting on CYBERNETICS AND SYSTEMS RESEARCH, Vienna

Goldschmidt G (1995) The designer as a team of one. Des Stud 16(2):189–209. doi:10.1016/0142-694X(94)00009-3

Goldschmidt G (2014) Linkography: unfolding the design process. The MIT Press, Cambridge

Jiang H, Gero JS, Yen C-C (2014) Exploring designing styles using a Problem-Solution Index. In: Gero JS (ed) Design computing and cognition DCC'12. Springer, Berlin, pp 87–104

Kan JWT, Gero JS (2009) Using the FBS ontology to capture semantic design information in design protocol studies. In: McDonnell J, Lloyd P (eds) About: designing: analysing design meetings. CRC Press, Boca Raton, pp 213–229

Kan JWT, Gero JS (2011) Learning to collaborate during team designing: quantitative measurements. In: Chakrabarti A (ed) Research into design. Research Publishing, Bangalore, pp 687–694

Koppel NB, Berenson ML (2006) Bowker Procedure. In: Salkind NJ, Rasmussen K (eds) Encyclopedia of measurement and statistics. Sage, New York, pp 107–111

Krauss RM, Fussell SR (1996) Social psychological models of interpersonal communication. In: Higgins ET, Kruglanski AW (eds) Social psychology: handbook of basic principles. Guilford Press, New York, pp 655–701

May WL, Johnson WD (2001) Symmetry in square contingency tables: tests of hypotheses and confidence interval construction. J Biopharm Stat 11(1–2):23–33. doi:10.1081/BIP-100104195

McDonnell J, Lloyd P (eds) (2009) About: designing: analysing design meetings. CRC Press, Boca Raton, FL

Mohammed S, Dumville BC (2001) Team mental models in a team knowledge framework: expanding theory and measurement across disciplinary boundaries. J Org Behav 22(2):89–106. doi:10.1002/job.86

Olson GM, Olson JS, Carter MR, Størrbsten M (1992) Small group design meetings: an analysis of collaboration. Hum Comput Interact 7(4):347–374

Pourmohamadi M, Gero JS (2011) LINKOgrapher: an analysis tool to study design protocols based on FBS coding scheme. Paper presented at the 18th international conference on engineering design (ICED’11), Copenhagen, Denmark

Sacks H, Schegloff EA, Jefferson G (1978) A simplest systematics for the organization of turn-taking for conversation. In: Schenkein J (ed) Studies in the organization of conversational interaction. Academic Press Inc., New York, pp 7–55

Schegloff EA (2007) Sequence organization in interaction: a primer in conversation analysis, vol 1. Cambridge University Press, New York, NY

Stempfle J, Badke-Schaub P (2002) Thinking in design teams: an analysis of team communication. Des Stud 23(5):473–496. doi:10.1016/S0142-694X(02)00004-2

Stivers T, Enfield NJ, Brown P, Englert C, Hayashi M, Heinemann T et al (2009) Universals and cultural variation in turn-taking in conversation. Proc Natl Acad Sci United States Am (PNAS) 106(26):10587–10592. doi:10.1073/pnas.0903616106

Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Des Stud 18(4):385–403

Wooffitt R (2005) Conversation analysis and discourse analysis: a comparative and critical introduction. Sage, New York

Individual Differences in Tendency for Design Fixation

Song Liang Lai and L.H. Shu

Abstract Not all individuals may be equally susceptible to design fixation. We sought to identify characteristics that could predict individual tendency for design fixation, and explored the use of Kruglanski's Need for Closure Scale for this purpose. We devised an experiment to determine whether correlations exist between participants' score on this scale and the degree of fixation in concepts elicited. Specifically, engineering-student participants were asked to complete the Need for Closure Scale as well as develop concepts for which an example solution was provided. Two statistical techniques, the Mann-Whitney U test and ordinal logistic regression, showed that participants' Need for Closure scores correlated significantly with degree of fixation in generated concepts.

Introduction

Jansson and Smith (1991) describe *design fixation* as "a blind adherence to a set of ideas or concepts limiting the output of conceptual design". Despite potential benefits of fixation, e.g., for experts who know when to pursue certain ideas, many researchers are interested in reducing fixation.

We are interested in possible individual differences in tendency towards design fixation, with implications for corresponding interventions. To predict tendency for fixation, we explore the use of the Need for Closure Scale, an individual difference variable developed by Arie Kruglanski, a social psychologist. Kruglanski (1990), describes Need for Closure as a desire for "*an* answer on a given topic, *any* answer", which we believe to be relevant to design fixation. Below, we highlight research on (1) design fixation and (2) the Need for Closure Scale, the intersection of which comprises the current work.

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

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_18

Design Fixation

Regarding what constitutes design fixation, Youmans and Arciszewski (2012, 2014) highlighted the challenge to researchers due to inconsistent definitions of design fixation. They classified design fixation phenomena as: unconscious adherence, conscious blocking, and intentional resistance. Furthermore, they proposed an updated definition of design fixation 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”.

Many researchers have worked to identify factors that contribute to design fixation. Purcell and Gero (1996) conducted a study using Jansson and Smith’s (1991) car bike-rack problem and found no evidence of fixation, except for one feature of a pictorial fixating example, noting that fixation may depend on familiarity with an example. Viswanathan and Linsey discussed the roles in fixation of physical modeling as a sunk cost (Viswanathan and Linsey 2011) and of expertise (Viswanathan and Linsey 2013). Other researchers have examined the effects of representation of fixation as well as the fixating example. For example, Zahner et al. (2010) and Dong and Sarkar (2011) studied the differing levels of representation/abstraction involved in fixation. Atilola et al. (2016) studied the effects of different representations of the fixating example, i.e., as a sketch, function tree, and the two combined.

Efforts to measure and quantify design fixation include the following. Linsey et al. (2010) and Atilola et al. (2016) measured fixation by counting the number of times features from an example were repeated in generated concepts. Genco et al. (2012) used a similar “conformity score”, counting the number of basic and additional features that were translated from examples to generated concepts for alarm clocks.

Efforts towards overcoming design fixation have been reported by several researchers including: Chrysikou and Weisberg (2005) instructing participants to avoid problematic elements in an example; Kohn and Smith (2009) using incubation and distraction; Youmans (2011) and Kershaw et al. (2011) using prototyping and critical feedback; Smith and Linsey (2011) adding hints and clues; Moreno et al. (2014) using analogies; and Toh et al. (2012) using product dissection. Specifically, Toh et al. (2012) found that undergraduate students exposed to more parts in a product-dissection activity generated less fixated and more ideas when subsequently redesigning the product.

The notion that tendency for design fixation is not equal among all individuals is supported. Bellows et al. (2013) noted how individual differences in working memory capacity affects design behavior. Toh et al. (2012) noted that extroverts examined more parts during product dissection. Toh et al. (2015) reported the effect of ownership bias during concept selection, and specifically a gender difference where more males than females tended to select their own ideas during concept selection.

As for our laboratory’s past work, we conducted several unpublished design fixation studies with varying results, where we had noted potential individual differences in susceptibility for design fixation. Therefore, we sought to identify metrics that could predict individual tendency for design fixation. We had tried the Myers-Briggs Type Indicator (MBTI), commonly used to form design teams, but did not detect significant differences in fixation. In our other work, Hallihan et al. (2012, 2013) found that confirmation bias can lead designers to "ignore or discount factual contradictory information" and "fixate on initial (confirmed) ideas". Confirmation bias is related to Kruglanski and Webster's (1996) "seizing" and "freezing" in high Need for Closure, further discussed below.

Need for Closure

Plaks (2011) describes those with high Need for Closure as more likely to make snap judgments, while those with low Need for Closure (or high need to avoid closure) may have enormous difficulty deciding what to order from a restaurant menu. Need for Closure is an individual difference variable, as well as a situational/environmental variable that can be manipulated in the laboratory, e.g., through time pressure.

Kruglanski and Webster (1996) described a desire to attain closure quickly that may manifest as frantic "seizing" upon relevant cues to come to a quick evaluation, followed by "freezing" or protecting that evaluation. In a state of high Need for Closure, less information may be processed in the seizing stage before a judgment is committed in the freezing stage. Paradoxically, with fewer competing hypotheses considered prior to evaluation, a potential result is an inflated sense of confidence in the evaluation. Those with heightened Need for Closure tended to rely more on early and readily accessible cues, an observation we found closely related to design fixation, where participants adhere to initial fixating ideas and examples. Such findings motivated us to pursue the current work.

Kruglanski and Fishman (2009) reported that Need for Closure may be situationally manipulated, at least temporarily. For example, Need for Closure can be heightened in time-pressure situations that require immediate decisions. Mayseless and Kruglanski (1987) studied situational or environmental Need for Closure. They asked participants to list as many as possible hypotheses for the identity of everyday objects presented in photographs. These photographs were enlarged to a point where the identity of the objects was unclear. The group in the high Need for Closure condition was told that reaching firm decisions is an indication of general intelligence. The group in the low Need for Closure condition was told that correct visual identification is an indication of general intelligence. The low Need for Closure condition group produced the most hypotheses, followed by a control group, while the high Need for Closure condition group produced the fewest hypotheses.

Kruglanski et al. (1993) devised a formal measure of dispositional Need for Closure. Known as the Need for Closure Scale, it includes five different subscales: (1) order and structure, (2) ambiguity, (3) decisiveness, (4) predictability, and

(5) close-mindedness. The Need for Closure Scale consists of 42 questions scored on a 6-point scale, for a total Need for Closure score range of 42 (low)–252 (high). Those with high Need for Closure generally have: preference for order and structure, discomfort with ambiguity, high decisiveness of judgments, desire for predictability, and high close-mindedness. Conversely, those with low Need for Closure have the opposite preferences. The Need for Closure Scale was validated by Webster and Kruglanski (1994), receiving high internal consistency in groups of n = 281 (Cronbach's $\alpha = 0.8405$) and n = 172 (Cronbach's $\alpha = 0.8413$). Test-retest reliability was also high ($r = 0.8611$). The Need for Closure Scale has been translated into different languages including Arabic, French, and Mandarin Chinese. According to Kruglanski and Fishman (2009), numerous studies support that Need for Closure Scale ratings can be meaningfully compared across countries and cultures.

We saw strong parallels to design fixation in Kruglanski et al.'s (1993) Need for Closure Scale statements such as: "When trying to solve a problem I often see so many possible options that it's confusing", "I do not usually consult many different options before forming my own view", and "When faced with a problem I usually see the one best solution very quickly". We therefore pursued our research question below.

Research Question

As Need for Closure is an individual characteristic, we argue that a designer carries out his or her work in accordance with this dispositional characteristic. Thus, we sought to answer our research question:

> Is there any correlation between an individual's dispositional Need for Closure score and his or her tendency towards design fixation?

Experiment

This section describes our experiment participants, method and execution.

Participants

Following approval by the University of Toronto Research Ethics Board, 29 participants were recruited from a fourth-year undergraduate engineering design course. This course covers topics, e.g., design fixation and design for resource conservation, which may affect the results of the study. Therefore, we took care to schedule this experiment before these topics were introduced in the course. Participants included students studying mechanical engineering, industrial engineering, and engineering science. Of the 29 participants, 22 were male and 7 were female.

Method

Given that we aimed to investigate the relationship between Need for Closure and tendency for design fixation, participants completed Kruglanski et al.'s (1993) 42-item Need for Closure Scale, and developed concepts intended to reduce the amount of time that users spent running water in a shower (Fig. 1). The Intatec shower (Fig. 2c) was provided as an example.

We selected the shower problem because all participants were likely to be familiar with the problem domain. In addition, we were already familiar with potential pitfalls of using this as the experimental problem due to our laboratory's other research on resource conservation (e.g., Srivastava and Shu 2013). To address these known pitfalls, we imposed restrictions on acceptable concepts, by prohibiting the use of water aeration, as well as the use of information, feedback, or automation as behavior-change strategies.

Problem: Reducing Water Consumption While Showering

Background: California has declared a state of emergency due to severe drought conditions. Governor Jerry Brown issued Executive Order B-29-15 on April 1, 2015, imposing a statewide 25% reduction in urban water usage.

Description: Your task is to design products that reduce the amount of time spent running water while showering. Your solutions must not incorporate information, feedback or automation. Water aeration is also not an acceptable solution. Information and feedback, such as the information sign shown (in Figure 2a), can easily be ignored. Automated showers, such as the motion sensor activated model shown (in Figure 2b) below, take away user control and are susceptible to sensor failures, malfunctions, and unintended activation.

A product that does not use information, feedback or automation is the Intatec shower panel shown (in Figure 2c) below. The Intatec shower panel is operated with a push button that activates a timed water flow of approximately 15s.

Definitions:
Information: Tell the user about the problem and the benefits of change.
Feedback: The effect of the actions of the user is conveyed to the user.
Automation: Device performs actions for the user.

Fig. 1 Design problem description provided to participants

Information sign

Motion sensor activated shower

Intatec shower panel

Fig. 2 Examples provided to participants

Execution Details

The experiment was completed during a regularly scheduled 3-h laboratory session, entitled "Creativity exercises" on the course syllabus. Instead of the regular laboratory room, the experiment took place in a computer classroom, with computer stations fixed in five rows, all facing the front of the room. Participants were seated in the rows with at least one empty computer station between them to reduce their ability to see others' computer monitors. Each participant was given a confidential note with an access code to the experiment, which was conducted online. The access code allowed participants to anonymously complete the two experimental tasks: the Need for Closure Scale and the concept generation activity.

The Need for Closure Scale was implemented as an online questionnaire, where participants answered each question by clicking on radio buttons that correspond to a Likert scale. Written instructions included, "Read each of the following statements and decide how much you agree with each according to your beliefs and experiences." Upon completion of the questionnaire, participants clicked a submit button at the bottom of the page to move on to the concept generation task. If they neglected to answer any questions, they were instructed to return to those questions.

The concept generation activity was also implemented online using canvases of 1000 by 600 pixels. Written instructions included, "Please draw your concepts on the canvas below." Verbal instructions were provided to clarify that participants were to click and hold the left button of their workstation mouse to draw on the canvas, as well as to only draw one concept per canvas. In addition, clicking the right mouse button opened a text box into which they could type and save annotations for their concepts. A choice of six colors (black, blue, red, green, yellow, gray) was available for both drawing and text input. Participants could also undo and redo drawing and text-input actions before submitting their concepts.

While no explicit time limit was given for specific parts of the experiment, participants had up to 1 h 15 min to complete both tasks (the usual duration until a mid-laboratory break). Participants spent 50–75 min on the activities, during which they were instructed to not communicate with other participants, and to raise their hands if they had any questions, which were answered individually.

Participants were debriefed in a 15-min session at the end of the 3-h laboratory, following activities unrelated to the experiment. (Specifically, related to product affordances, students were asked to identify shared vs. unique action opportunities for pairs of products that fulfill similar functions, e.g., paper clip vs. binder clip.) Consistent with our ethics protocol, the debriefing consisted of a short explanation of the NFC scale and our research question of whether it is related to design fixation. We also clarified that the activity did not affect their course grade, and then distributed information letters requesting permission to use their data. All participants agreed to our use of their data in this experiment.

Results

Fixation Evaluation

We measured fixation relative to the provided Intatec shower example through degree of similarity in (1) form and (2) function. This evaluation is similar to the work of Linsey et al. (2010, 2013) and Genco et al. (2012), which compared participant concepts to features of provided examples. Table 1 shows our fixation-rating scheme. Table 2 demonstrates how we applied this scheme to 3 example concepts, shown in Figs. 3, 4 and 5.

Figure 3 shows the 1st sample concept, which is the 2nd of 3 concepts, as submitted by participant D. For Form Similarity, the concept was rated 5/5 because it used an identical panel housing and multiple push buttons. Although the concept used multiple push buttons versus Intatec's single button, it was still highly similar in form. For Function Similarity, the concept was rated 5/5, as the buttons had to be functionally identical (timed) push buttons for the concept to work. We treated the absence of explicit notes to the contrary as the participant intending the same function for the same form on the Intatec. In this case, while the participant did not explicitly state that the push button activates a timed water flow, it was rated as such because of the absence of any alternative information.

Figure 4 shows the 3rd of 8 concepts, as submitted by participant I. The concept rated 3/5 on Form Similarity because of the similarity between the pull handle and push button, especially given the explicit transfer of the 15-s timed water flow from the Intatec. The concept rated 5/5 on Function Similarity as the timed-water flow uses the same strategy as the Intatec.

Table 1 Fixation score rating scheme

Score	Form similarity	Function similarity
	Similarity in appearance to Intatec example; fixating features include panel, push button, but not showerhead	*Similarity in function (strategy) to Intatec example, which discretizes (interrupts) water flow through timed button*
1	Concept borrows no features from Intatec and has novel form	Concept uses strategy other than discretized flow (e.g., require work)
2	Concept borrows no features from Intatec but looks like a conventional shower	Concept uses resource-based discretized water flow (e.g., packetization of water as in a camping shower)
3	Concept uses features similar, but not identical, to Intatec (e.g., lever or other timer instead of timed button)	Concept implements other time-based flow i.e. "only on when needed" (e.g., pressure-sensitive pad)
4	Concept uses at least one identical feature to that used in Intatec	Concept uses an explicit countdown timer (e.g., kitchen timer)
5	Concept solely uses (multiple) panel(s), push button(s) as in Intatec	Concept uses timed device to provide water flow for set time

Table 2 Evaluations of sample concepts

Participant ID Concept # of total	Form similarity score and reasoning	Function similarity score and reasoning
D Concept 2 of 3	5/5, identical panel and push button elements	5/5, water flow controlled by timed button
I Concept 3 of 8	3/5, pulling handle instead of pushing button	5/5, water flow controlled by timed device
J Concept 1 of 8	1/5, no features from Intatec, novel form	1/5, used neither timed nor discretized flow as strategy

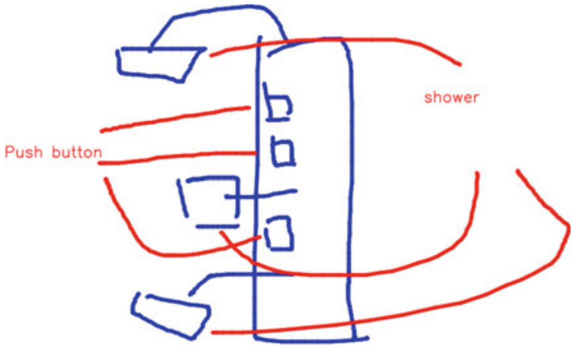


Fig. 3 Participant D, concept 2 of 3, similarity ratings: form 5/5, function 5/5, combined 10/10 (high)

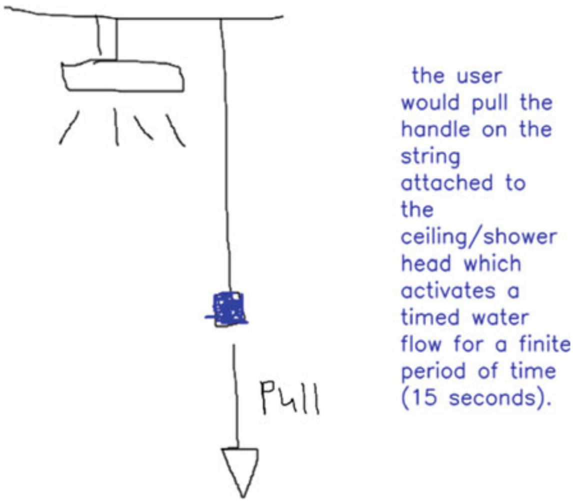


Fig. 4 Participant I, concept 3 of 8, similarity ratings: form 3/5, function 5/5, combined 8/10

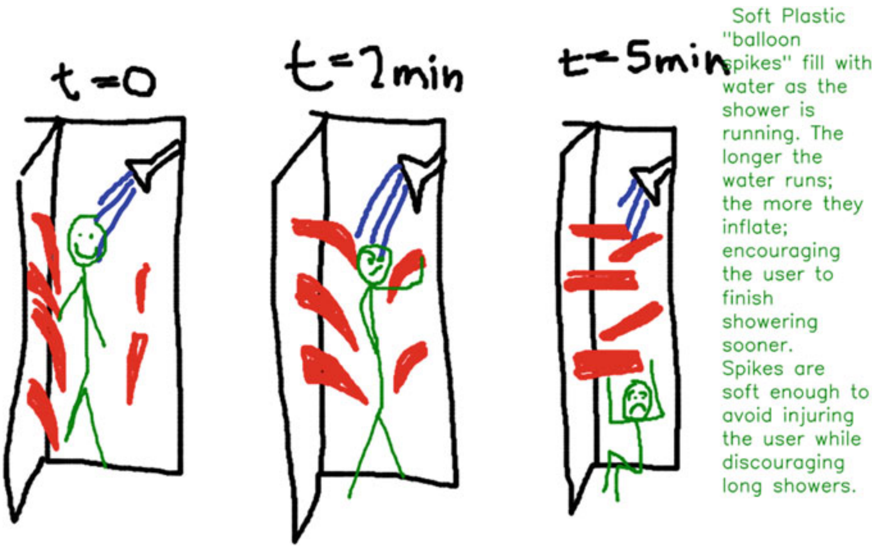


Fig. 5 Participant J, concept 1 of 8, similarity ratings: form 1/5, function 1/5, combined 2/10 (low)

Figure 5 shows the 1st of 8 concepts, as submitted by participant J. The concept rated 1/5 on Form Similarity because it does not borrow any features from the Intatec, and the "balloon spikes" constitute a novel form. The repeated use of a showerhead was not counted in Form Similarity. The concept rated 1/5 on Function Similarity as it does not use discretized (interrupted) water flow as a resource-conserving strategy. Rather, it aimed to decrease the comfort of the user through the use of "balloon spikes" which increasingly inflate as the shower runs.

In total, the 29 participants generated 110 concepts. Two raters, the first author and a second rater (not one of the authors) both rated all 110 concepts. Concepts were rejected if they did not follow the instructions stipulated in the problem description. Examples of rejected concepts include showers using aeration, and only information and/or automation as resource-conserving strategies. The first author accepted 88 concepts while the second rater accepted 84 concepts, which were a subset of the 88 concepts accepted by the first author.

The two components of rating for each concept, form similarity and function similarity, were summed into a combined similarity, or fixation score. The inter-rater reliability of the fixation score on the 84 concepts accepted by both raters was calculated as Krippendorff's α (ordinal) = 0.798, marginally below $\alpha \geq 0.8$, which corresponds to perfect agreement (Krippendorf 2004). Given this high level of agreement, we conducted further statistical analysis using the first author's ratings for the 88 accepted concepts.

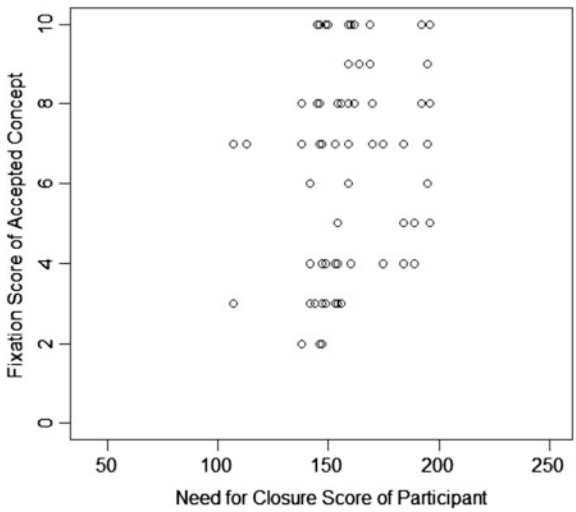


Fig. 6 Fixation scores of accepted concepts versus need for closure

Observations

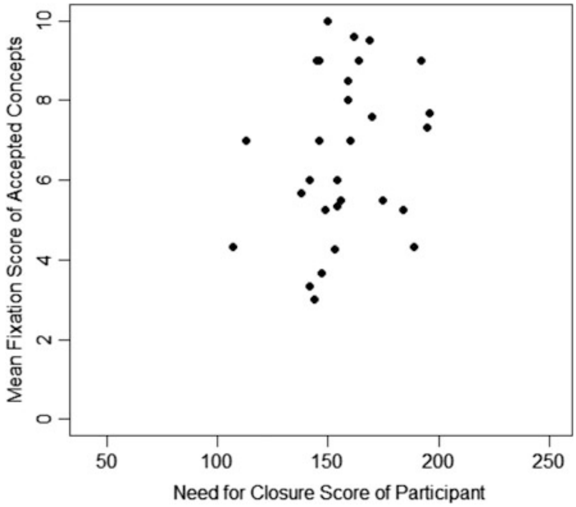
Figures 6 and 7 plot as functions of participants' Need for Closure scores, (1) fixation scores for each accepted concept, and (2) mean fixation score of accepted concepts per participant. A preliminary assessment of this data was performed using the Goodman-Kruskal gamma test, which assesses the similarity of ranking order between pairs of observations, not adjusting for ties. Goodman-Kruskal γ ranges from -1 (perfect negative association) to $+1$ (perfect positive association). We obtained Goodman-Kruskal $\gamma = 0.28$ for Fig. 6, which indicates a weak positive correlation between participants' Need for Closure and their accepted-concepts' fixation scores. Goodman-Kruskal $\gamma = 0.22$ for Fig. 7 indicates a weak positive association between participants' Need for Closure scores and their accepted-concepts' mean fixation scores.

In subsequent sections, we use the term "participant" to refer to actual participants of our experiment, and the term "individual" to refer to any given individual of a population, e.g., when interpreting the results of our ordinal logistic regression model.

Statistical Analysis

Our statistical analysis included two techniques. First, the Mann-Whitney U non-parametric statistical test was used to compare (ordinal) fixation scores of concepts by participants of low versus high Need for Closure (NFC). Second, an

Fig. 7 Mean fixation scores of accepted concepts per participant



ordinal logistic regression was used to model how participants' NFC scores affected their concepts' fixation scores. Both statistical techniques yielded significant results.

Mann-Whitney U Test

Table 3 shows participants placed in low versus high NFC groups based on their score, using a mean split. Participants who scored below the mean of 157.24 were placed in the low-NFC group, and the remaining were placed in the high-NFC group. We used the mean fixation scores of participants' concepts to meet the Mann-Whitney U test assumption of independent samples. In addition, the ordinal nature of fixation scores did not violate the Mann-Whitney U test assumption of non-nominal values. The mean fixation scores of participants' concepts in the low-NFC and high-NFC groups were 5.90 and 7.56, respectively.

We used a mean instead of a median split as we had 29 participants/data points and wanted to avoid removing 2 data points. As a non-parametric technique, the Mann-Whitney U test is also inherently more conservative compared to parametric techniques. Previously published analyses involving the NFC Scale used both mean (Holbert and Hansen 2006) and median splits (Kardes et al. 2007).

Table 3 Mann-Whitney U test groups

	Low NFC	High NFC
Number of participants, n	16	13
Mean NFC score	142.88	174.92
Mean participants' concepts fixation scores	5.90	7.56

The Mann-Whitney U test revealed significant differences in the mean fixation scores of concepts between participants in the low and high NFC groups (Mann-Whitney $U = 55.5$, $Z = -2.10$, $m_{\text{fixationscore1(high)}} = 7.56$, $m_{\text{fixationscore2 (low)}} = 5.90$, $n_1 = 13$, $n_2 = 16$, $p < 0.05$, $r = 0.39$, two-tailed). The effect size was $r = 0.39$, corresponding to a medium effect.

We also tested for, but did not find, significant differences in either NFC scores or fixation scores by gender in our 29 participants (7 female and 22 male). The female group had a mean NFC score of 160.29 and a mean fixation score of 6.13/10. The male group had a mean NFC score of 156.27 and a mean fixation score of 6.80/10. A Mann-Whitney U test did not find any differences in NFC scores between females and males (Mann-Whitney U = 86.5, Z = 0.48, $m_{1\text{(female)}} = 160.29$, $m_{2\text{(male)}} = 156.27$, $n_1 = 7$, $n_2 = 22$, $p = 0.63$, $r = 0.09$, two-tailed). The effect size $r = 0.09$ was below the $r = 0.10$ cutoff for a small effect. Another Mann-Whitney U test found no significant difference in the mean fixation scores of participants' concepts between males and females (Mann-Whitney $U = 59$, $Z = -0.92$, $m_{\text{fixationscore1(females)}} = 6.13$, $m_{\text{fixationscore2(males)}} = 6.80$, $n_1 = 7$, $n_2 = 22$, $p = 0.36$, $r = 0.17$, two-tailed). The effect size of $r = 0.17$ corresponds to a small effect.

Ordinal Logistic Regression

As our fixation metric is an ordinal variable, i.e., has two or more ordered/ranked categories, we used ordinal logistic regression. This regression is used to predict an ordinal dependent variable given one or more independent variables, and can also use interactions between independent variables to predict the dependent variable (McCullagh 1980). We selected participants' NFC score as the predictor (independent) variable and fixation score of participants' concepts as the outcome (dependent) variable. Fixation scores ranged from 2 (least fixated) to 10 (most fixated).

Because participants produced multiple concepts, our data consisted of repeated measures. We therefore used generalized estimating equations (GEE) for our model, which was fitted using PROC GENMOD in SAS 9.1, with further specifications of cumlogit link function and multinomial distribution. Table 4 shows significant effects for the predictor variable NFC score (log odds ratio = 0.0235, $\chi^2(1) = 4.32$, $p < 0.05$), which is equivalent to an odds ratio of 1.024 (exp (0.0235)). This means that if an individual's NFC score increases by 1 unit or point, there is a 2.4% increased likelihood that a concept by that individual will score a higher fixation score relative to the contiguous lower fixation score, further

Table 4 Log odds ratio for increase by one in NFC score on fixation score

Log odds ratio	95% confidence interval	Wald test
0.0235	0.0035–0.0435	$\chi^2(1) = 4.32$, $p = 0.038$*

*$p < 0.05$

Table 5 Probability distributions of a concept from an individual with various NFC scores of achieving a particular fixation score

NFC	Probability that concept from individual with NFC score corresponding to row label would have fixation score corresponding to column label									Sum
	2	3	4	5	6	7	8	9	10	
42	0.327	0.466	0.091	0.023	0.014	0.037	0.023	0.005	0.011	1
43	0.322	0.468	0.093	0.024	0.015	0.038	0.024	0.005	0.012	1
77	0.176	0.452	0.143	0.042	0.027	0.074	0.049	0.011	0.025	1
112	0.086	0.340	0.171	0.059	0.041	0.126	0.097	0.024	0.056	1
147	0.040	0.206	0.149	0.062	0.047	0.169	0.163	0.046	0.119	1
182	0.018	0.108	0.097	0.047	0.038	0.166	0.216	0.075	0.234	1
217	0.008	0.051	0.053	0.028	0.024	0.120	0.211	0.095	0.411	1
251	0.004	0.024	0.026	0.014	0.013	0.070	0.155	0.087	0.608	1
252	0.003	0.023	0.026	0.014	0.013	0.069	0.153	0.086	0.613	1

explained below. We assessed the validity of our model using a likelihood-ratio test and obtained significance at the 0.05 significance level ($\chi^2(1) = 6.24$, $p = 0.0125$).

To further illustrate the above, Table 5 shows the probabilities, obtained from our model, of a concept from individuals with NFC scores shown as row labels, achieving fixation scores shown as column labels. For example, the probabilities that a concept by an individual with NFC score of 42, will be scored fixation scores of 2 or 3 is 0.327 or 0.466, respectively. If an individual has an increase of one unit in NFC score (i.e., 42 + 1 = 43), the probabilities of a concept by that individual achieving fixation scores of 2 or 3 become 0.322 or 0.468 respectively. The odds for an individual with NFC score of 42 to develop a concept with a fixation score of 3 rather than 2 is 0.466/0.327 = 1.425. Similarly, with NFC score 43, these odds are 0.468/0.322 = 1.453. Thus, if an individual's NFC score increases from 42 to 43, the odds of their concept having a fixation score of 3 rather 2 is increased from 1.425 to 1.453, approximately a 2.4% increase. This 2.4% increase is consistent in different contiguous NFC score comparisons and different contiguous fixation score comparisons. For example, if an individual's NFC score increases from 251 to 252, the odds of his/her concept obtaining a fixation score of 10 rather than 9 increases by approximately 2.4% or ([0.613/0.086]/[0.608/0.087]).

Table 5 is a subset of the data shown in Fig. 8, a probability distribution plot of our model, which estimates the probability of a concept by an individual with a specific NFC score of achieving a particular fixation score. The 9 fixation score probability curves represent the probabilities of a concept achieving each respective fixation score outcome. By definition, for each NFC score, the total of each fixation score probability sums to 1.

We note that curves corresponding to the lowest (2) and highest (10) fixation scores make intuitive sense. The curve corresponding to Fixation Score 2 (Least Fixated) show that it is more probable for a lower NFC individual's concept to obtain a fixation score of 2, where probabilities range from 0.327–0.003 for NFC scores of 42–252. Conversely, the curve corresponding to Fixation Score 10 (Most

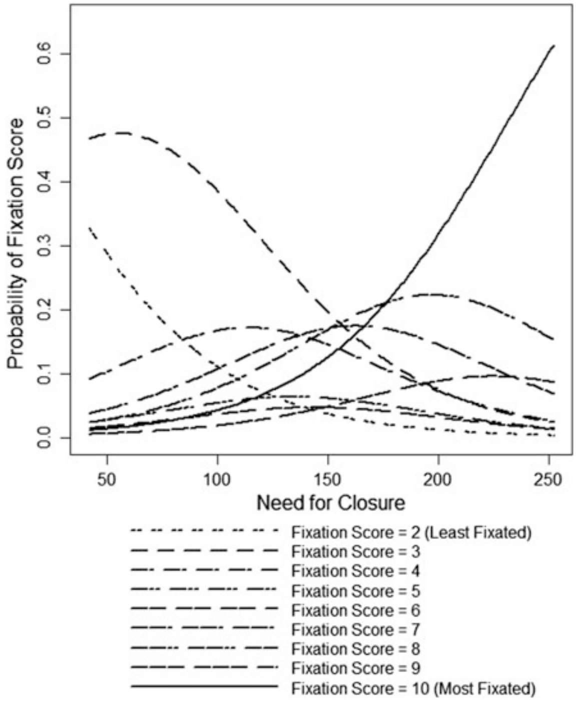


Fig. 8 Probability distribution plots for fixation scores

Fixated) shows that it is more probable for a higher NFC individual's concept to attain a fixation score of 10, where probabilities range from 0.011–0.613 for NFC scores of 42–252.

Discussion

We investigated the relationship between engineering-student designers' Need for Closure (NFC) score with the degree of fixation to a provided example, on concepts generated that aimed to reduce shower water-run time.

In the first statistical analysis, participants were divided into two groups, low NFC and high NFC based on a mean split of their NFC scores. A Mann-Whitney U test confirmed significant differences between the mean fixation scores of participants' concepts in the low versus high NFC groups. Specifically, the high NFC group had a higher mean fixation score than the low NFC group. Another Mann-Whitney U test was used to investigate gender effects, but found no significant differences between mean fixation scores of females' versus males' concepts or NFC scores.

In the second statistical analysis, ordinal logistic regression was used to create a model of participants' NFC scores and the fixation scores of their concepts. This model showed that an individual's tendency to fixate on the provided example for our concept generation problem increased with increased NFC score. Specifically, an increase by 1 in an individual's NFC score is associated with a 2.4% increased chance of a concept by that individual to be rated the next higher fixation score.

While our sample size of 29 participants is admittedly small, our results support our hypothesis of a relationship between NFC and design fixation, and relate to differences in design fixation observed by other researchers. For example, Purcell and Gero (1996) asked mechanical engineering and industrial design students to develop concepts that assist elderly people in and out of a bathtub, and provided an Autofit fixating example. Of the Autofit's 12 detail features, the mechanical engineering students fixated on 8 features (significant difference between experimental and control groups) while the industrial design group fixated on 3. Purcell and Gero (1996) note that a potential explanation is "differences in educational processes in the two disciplines" where industrial design "emphasizes creativity and difference". This difference may correspond to educations that attract and/or encourage higher NFC in mechanical engineers and lower NFC in industrial designers.

Bellows et al. (2013) also found that individual differences in working memory capacity correspond to different behavior with respect to design fixation. Toh et al. (2012) inversely related fixation with the number of parts examined during product dissection, and noted that extraverts examined more parts. Combined with our results, there may also be a relationship between NFC and introversion versus extraversion. Different from Toh et al.'s (2015) results of a gender difference in ownership bias during concept selection, we did not observe statistically significant difference by gender in either NFC or fixation scores.

We also did not observe a correlation between NFC score and the number of concepts that participants submitted. Others, e.g., Mayseless and Kruglanski (1987), showed that high NFC leads to fewer generated hypotheses. We attribute the absence of this correlation in our work to the fact that past work manipulated high and low NFC conditions through situational or environmental means, whereas we measured dispositional (individual) NFC.

In hindsight, we would have had participants complete the concept generation activity prior to completing the NFC questionnaire, however, we believe that this would not have a significant effect on our results because of the few number of questions directly related to concept-generation behavior. Overall, despite our relatively small sample size, the preliminary results are sufficiently promising to warrant further studies in this direction.

Conclusions and Further Work

We have shown that individual differences in Need of Closure (NFC) may be potentially used to predict an individual's tendency for design fixation. In our present study, participants' NFC scores were significantly correlated with their concepts' design fixation scores in an intuitive manner; higher NFC scores corresponded to higher fixation scores, and vice versa.

We did not attempt to manipulate NFC environmentally in the present study, i.e., by providing different experimental conditions. However, others' past work, demonstrating that NFC can be situationally manipulated, points to potentially individualized interventions to overcome fixation. For example, Mayseless and Kruglanski (1987) informed participants that reaching firm decisions is an indication of general intelligence to induce a high NFC condition. Such manipulations could be explored to examine their effects on fixation.

Finally, other characteristics have been significantly linked to NFC. For example, Jost et al. (2003) demonstrated a robust association between NFC and political orientation, i.e., conservatives have higher NFC. This and other characteristics could be used as a shortcut to the NFC questionnaire, in estimating individuals' tendency to fixate, and implementing interventions as required.

Acknowledgements We thank our participants for consenting and contributing to our study, and Professor Birsen Donmez for her assistance in our statistical analyses. We appreciate the guidance and approval from the University of Toronto Research Ethics Board. Finally, we gratefully acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

Atilola O, Tomko M, Linsey JS (2016) The effects of representation on idea generation and design fixation: A Study comparing sketches and function trees. Des Stud 42(1):110–136

Bellows BG, Higgins JF, Youmans RJ (2013) An individual differences approach to design fixation: comparing laboratory and field research. In: Design, user experience, and usability. Proc 2nd int conf design, user experience, and usability: design philosophy, methods, and tools. Springer, Berlin, pp 13–21

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

Dong A, Sarkar S (2011) Unfixing design fixation: from cause to computer simulation. J Creat Behav 45:147–159

Genco N, Hölttä-Otto K, Seepersad CC (2012) An experimental investigation of the innovation capabilities of undergraduate engineering students. J Eng Educ 101(1):60–81

Hallihan GM, Shu LH (2013) Considering confirmation bias in design and design research. J Integr Des Process Sci 17(4):19–35

Hallihan GM, Cheong H, Shu LH (2012) Confirmation and cognitive bias in design cognition. ASME IDETC/CIE, Chicago, IL

Holbert RL, Hansen GJ (2006) Fahrenheit 9–11, Need for Closure and the priming of affective ambivalence: an assessment of intra-affective structures by party identification. Hum Commun Res 32(2):109–129

Jansson DG, Smith SM (1991) Design fixation. Des Stud 12(1):3–11

Jost JT, Kruglanski AW, Glaser J, Sulloway FJ (2003) Political conservatism as motivated social cognition. Psychol Bull 129(3):339–375

Kardes FR, Fennis BM, Hirt ER, Tormala ZL, Bullington B (2007) The role of the need for cognitive closure in the effectiveness of the disrupt-then-reframe influence technique. J Consum Res 34(3):377–385

Kershaw T, Hölttä-Otto K, Lee Y (2011) The effect of prototyping and critical feedback on fixation in engineering design. In: 33rd annual conference of the cognitive science society CogSci'11. Boston, MA

Kohn N, Smith S (2009) Partly vs. completely out of your mind: effects of incubation and distraction on resolving fixation. J Creat Behav 43(2):102–118

Krippendorf K (2004) Reliability in content analysis: some common misconceptions and recommendations. Hum Commun Res 30(3):411–433

Kruglanski AW (1990) Motivations for judging and knowing: implications for causal attribution. Found Soc Behav 2:333–368

Kruglanski AW, Fishman S (2009) The need for cognitive closure. handbook of individual differences in social behavior. Guilford Press, New York, pp 343–353

Kruglanski AW, Webster DM (1996) Motivated closing of the mind: "seizing" and "freezing". Psychol Rev 103:263–283

Kruglanski AW, Webster DM, Klem A (1993) Motivated resistance and openness to persuasion in the presence or absence of prior information. J Pers Soc Psychol 65(5):861–876

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(4):041003-1-12

Mayseless O, Kruglanski AW (1987) What makes you so sure? Effects of epistemic motivations on judgmental confidence. Organ Behav Hum Decis Process 39(2):162–183

McCullagh P (1980) Regression models for ordinal data. J R Stat Soc Ser B (Methodol) 42 (2):109–142

Moreno DP, Yang MC, Hernández AA, Linsey JS, Wood KL (2014) A step beyond to overcome design fixation: a design-by-analogy approach. In: Gero JS, Hanna S (eds) Design computing and cognition '14. Springer, pp 607–624

Plaks JP (2011) The social psychology of motivation. Oxford University Press, Oxford

Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17(4):363–383

Smith S, Linsey J (2011) A three-pronged approach for overcoming design fixation. J Creat Behav 45:83–91

Srivastava J, Shu LH (2013) Encouraging resource-conscious behavior through product design: the principle of discretization. ASME J Mech Des 135:061002

Toh C, Miller S, Kremer G (2012) Mitigating design fixation effects in engineering design through product dissection activities. In: Gero JS (ed) Design computing and cognition'12. Springer, pp 95–113

Toh C, Patel A, Strohmetz A, Miller S (2015) My idea is the best! Ownership bias and its influence on engineering concept selection. ASME IDETC/CIE, Boston MA USA

Viswanathan V, Linsey J (2011) Design fixation in physical modeling: an investigation on the role of sunk cost. ASME IDETC/CIE, Washington, DC

Viswanathan V, Linsey J (2013) Design fixation and its mitigation: a study on the role of expertise. J Mech Des 135(5):051008-1-15

Webster DM, Kruglanski AW (1994) Individual differences in need for cognitive closure. J Pers Soc Psychol 67(6):1049–1062

Youmans RJ (2011) The effects of physical prototyping and group work on the reduction of design fixation. Des Stud 32(2):115–138

Youmans R, Arciszewski T (2012) Design fixation: a cloak of many colors. In: Gero JS (ed) Design computing and cognition'12. Springer, pp 115–129

Youmans RJ, Arciszewski T (2014) Design fixation: classifications and modern methods of prevention. AIEDAM 28(2):129–137

Zahner D, Nickerson J, Tversky B, Corter J, Ma J (2010) A fix for fixation? Re-representing and abstracting as creative processes in the design of information systems. AIEDAM 24(2):231–244

Functional Thinking: A Protocol Study to Map Modeling Behavior of Designers

Ashwinraj Thiagarajan, Apurva Patel, Steven O'Shields and Joshua D. Summers

Abstract Function modeling is a tool used to map functional requirements of a design problem to the solution space. Research has been done on various representations and uses of function modeling, however, there is a lack of research on how designers think about function modeling. This paper presents a protocol study which was used to analyze modeling behaviors and further the understanding of how designers create function models and subsequently understand how designers cognitively process function models. The experimental setup used for the study, the data collection, and the analysis of participant generated models are provided in this paper. Preliminary results suggest that designers mostly use forward chaining when modeling function structures, with some nucleation and almost no backward chaining. Moving forward, this study will be expanded to include a larger participant pool in order to provide substantial evidence for the conclusions. Additionally, other patterns that exist in designer activity will be explored and investigated from a cognitive perspective.

Motivation for Research on Functional Thinking

The design process is the intentional transformation or mapping from a problem space to a solution space. This transformation is performed in stages, such as problem definition, conceptual design, embodiment, or detailed design (Pahl et al. 2007; Ullman 2010). Function modeling is a tool that can be used during the design process to relate the knowledge obtained during problem definition, such as requirements, to the information that is known in the solution phase, the end product's function. Functions can be used in the requirement stages to understand the problem (Albers et al. 2004; Schaafstal et al. 2000), and also in the solution phase to generate concepts (Ottosson 1998; Hey et al. 2008).

A. Thiagarajan · A. Patel · S. O'Shields · J.D. Summers (✉)
Clemson University, Clemson, SC, USA
e-mail: jsummer@clemson.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_19

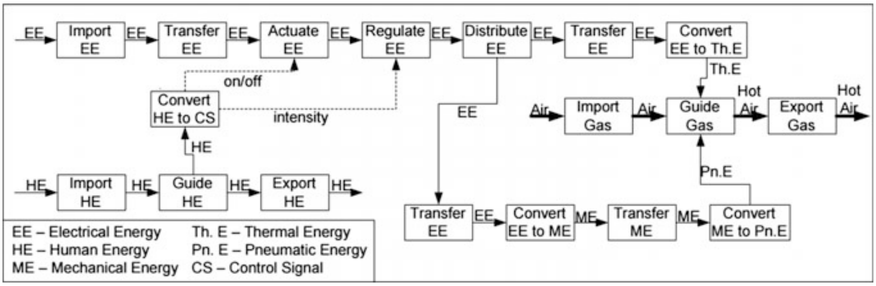


Fig. 1 Function model for a hair dryer from the design repository (http://repository.designengineeringlab.org/)

A typical function structure model is a compound flow chart of functions and flows, as shown in Fig. 1. A function is usually the anticipated effect of the component which helps in guiding, transforming, or regulating a flow of material, energy, or a signal in the part or system (Pahl et al. 2007). All of these transformations happen inside a system boundary within which energy and mass are conserved.

At present, there is experimental research to understand functional modeling and its outcomes (Thomas et al. 2009; Ramachandran et al. 2011; Nagel et al. 2013), but limited research has been conducted on understanding *how these models are being constructed* and used as a solution for design problems. Results from this research could enhance the use of function structures in academia, by helping improve student understanding about how designers model and, in turn, promote effective designs. This paper will use an existing protocol that has been tested and used to determine whether designers exhibit a certain pattern when using function structures as a modeling tool (Sen and Summers 2012). Specifically, the outcome of this experiment will help in understanding the behavior of designers, and how they construct function structures, in terms of modeling chaining.

Chaining of Function Models

Chaining of a model describes the directionality of the model as it was being constructed. Three different methods of model chaining can be identified: forward chaining, backward chaining, and nucleation. The model in Fig. 2 will be used as an example to demonstrate the three types of chaining discussed in this paper.

Forward Chaining

Forward chaining is a technique that generates the model from the known inputs to the desired outputs. For example, if the model in Fig. 2 was generated using

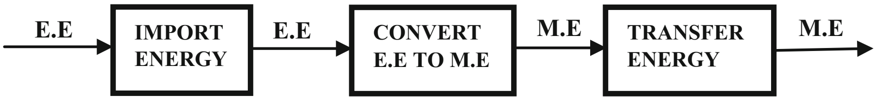


Fig. 2 Sample model used to generate general graphs for comparison

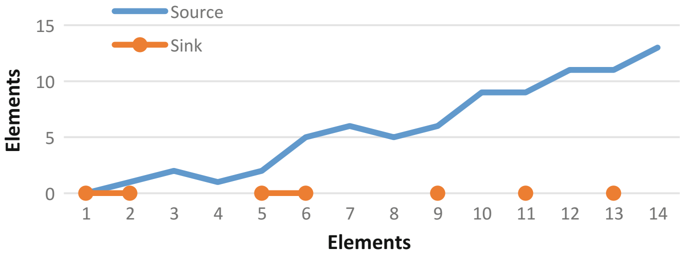


Fig. 3 General topology graph for forward chaining

forward chaining, the elements would be added such that the left most element (E.E.) would appear first, with the subsequent elements being added from left to right. As such, the source for a function is always added before the function itself, and the sink for the function always follows the function. A source is the element that serves as an input, whereas a sink is the element that acts as the output. This source-sink relationship can be graphically represented in a topology graph. An example of the topology graph for forward chaining is shown in Fig. 3.

In Fig. 3, the element, number chronologically, are shown on both axes, and the sources and sinks identified for each element are graphed. For a forward chaining model, the line connecting the sources shows a positive slope.

Backward Chaining

Backward chaining is done in the reverse order compared to forward chaining, with the final output being added first and the initial input being added last. In backward chaining, the sink for the function is always added before the function, and the source for the function is always added after the function. A general topology graph for backward chaining can be seen in Fig. 4.

As shown in Fig. 4, the line connecting the sources shows a wave form instead of a positive slope. This is indicative of backward chaining because it shows that the sinks to an element are added before the sources.

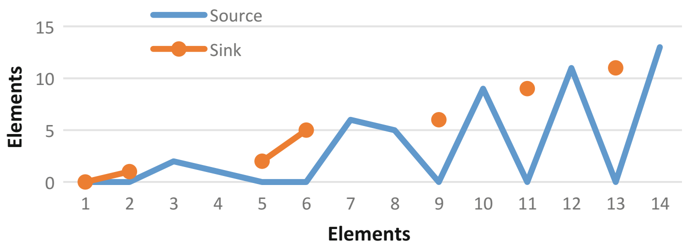


Fig. 4 General topology graph for backward chaining

Nucleation

Nucleation of a model occurs when most central, or the element carrying the most information, is added to the model first and the rest of the model is nucleated from there. In other words, nucleation uses both forward and backward chaining as needed. For example, if Fig. 2 was generated using nucleation, the "Convert EE to ME" function (block) would be added first, the input (E.E.) and output (M.E) flows for that function second, the functions associated with those flows next, and so on. A general topology graph for nucleation can be seen in Fig. 5.

As shown in Fig. 5, a topology graph for nucleation can described as a combination of the forward chaining and backward chaining topology, a wave form moving along a line with positive slope. Previous research showed that designers use forward chaining and nucleation for function modeling of novel design problems. Backward chaining and nucleation were generally used for reverse engineering (Sen and Summers 2012).

Pilot Protocol Study

A protocol study was conducted by Sen and Summers (2012) and Sen et al. (2010) to understand the modeling behaviors of designers using function modeling. The

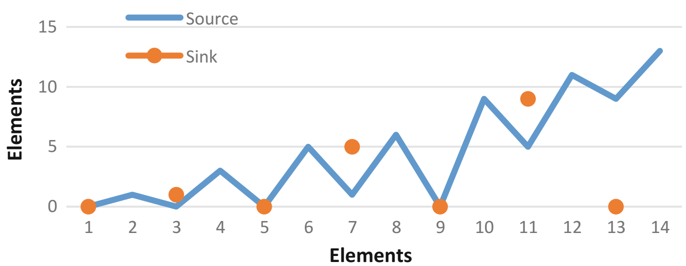


Fig. 5 General topology graph for nucleation chaining

pilot examined two participants, with varied background in design experience and with different years of design experience. The same design problem was given to both the designers, and their activities were recorded and coded. Findings from the pilot study suggested that a pattern may exist in how designers use function models. Observations from the study show that forward chaining and nucleation were primarily used for construction of the models. Additionally, the more experienced designer was recoded using nucleation more often than the novice designer.

As this study only used two participants, statistically significant conclusions cannot be drawn from the results. However, the data analysis methods developed for this study can be useful. An inter rater reliability test was conducted on the protocol and results suggest that the coding process was reliable.

The pilot study conducted by Sen and Summers (2012) and Sen et al. (2010) was the genesis of this research. The gaps identified in the pilot study lead to the following research questions:

1. "*Are there any patterns in designer's modeling behaviors when using function modeling for novel design problems?*"
2. "*Do designers use forward chaining predominantly while they model a novel design problem?*"

Experimental Setup

This research employs a protocol study approach to achieve the research deliverables. Four main factors were considered for this protocol study: the design problem, participant selection, mode of data collection, and the coding scheme. The following sections will discuss the study structure and coding scheme followed.

Design Problem

The design problem used for the study must be clear, concise, and feasible for artifact production and should be similar across cultures (Fraenkel and Wallen 1993). As this study is in the context of generative design, the problem focuses on a new product. However, the fundamental principles should be familiar to the engineer. This ensures that the participants are familiar with the domain and the models they generate are useful.

The problem given is a combination automatic ironing and folding machine, which has been adapted from prior research (Sen and Summers 2012). The problem satisfies the previously explained constraints of clarity, conciseness, significance and similarity over all cultures:

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.*

Participant Selection

The eight volunteer participants of this study were all graduate students from Clemson University with equivalent or comparable experiences and education. Each of the participants was required to answer a questionnaire that ensured they had sufficient design experience and expertise with function modeling to be observed for the research. All participants had taken the graduate level course in design methods and had design experiences ranging from two to five years.

All the participants were subjected to same environment and same time of the day to maintain consistency. The participants were not asked to complete the model in a set amount of time, however, it was anticipated that 45 min would be a sufficient amount of time for the actual modeling activity. The volunteers were told the following before the start of the experimental session:

1. They were to use function structures to address the design problem.
2. There would be no time limit to develop the model.
3. They were going to be videotaped by two different cameras.
4. They may read the problem statement and ask any clarification questions.
5. They may revisit the problem statement during modeling as needed.

Data Analysis Through Video Encoding

The selected method to capture the behavioral information without interfering with the designers while they model was through video recording. The experimental setup used for each participant included a dry-erase board and a tray, which contained dry-erase markers and erasers. Cameras were consistently placed in two different locations to capture all pertinent data whether the participant was right or left handed.

Coding and Steps to Code

Once the experiment was conducted, the aim was to record every activity of the designer and assign them with a code. Coding was carried out for the following three components of the activity.

1. Element coding—used to code the element first.
2. Activity coding—used to code the activity of the designer.
3. Topology coding—used in identifying how elements were connected while they were added.

Overall, there are six steps to encode designer behavior into element, activity, and topology coding. Detailed coding steps can be found in the "Appendix".

Element Coding

The previous study by Sen and Summers (2012) and Sen et al. (2010) identified the various elements that are used in modeling and tabulated them. An alphabet code was assigned to the most frequently used elements. The goal of element encoding is to capture the type of element being used by the participant. Table 1 gives all the codes used along with their definition.

Figure 6 gives a pictorial representation of all the elements used in a model along with how to identify the elements.

Activity Coding

Activity coding allows for the modeling activities of the designers to be more easily compared between designers. The pilot study in Sen and Summers (2012) and (Sen et al. (2010) helped show that there was a set of basic activities that were carried out by the designers while they modeled the product. These activities are given abbreviated codes, which can be used to identify the activity they carried out. The goal of activity coding was to gather all the information regarding which elements was added or deleted along with the time of action. All the codes and the definition associated with the code are given in the Table 2.

Table 1 Element coding scheme

Element	Code	Definition
Block	B	A rectangle typically used to represent a 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
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
Note	N	A textual or symbolic expression that is not an ET or a BT

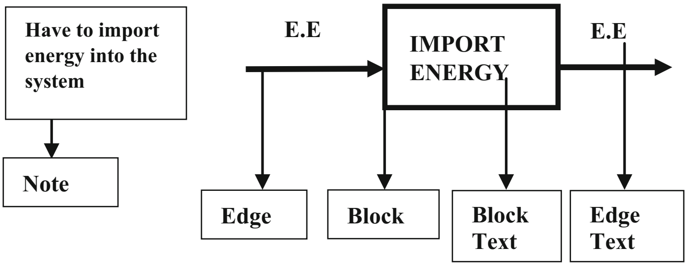


Fig. 6 Description of all the elements used in the model and coding scheme

Table 2 Activity coding scheme

Activity	Code	Definition
Add element	A	To add a new element into the model
Delete element	D	To delete an element from the model. Partial deletion will be considered a full deletion
Edit element	E	To make changes to an existing element. It could mean a simple change in route or arrow or the text on or in an element
Read problem stmt.	PS	To interact with the given design problem and instructions, while not doing A, D, or E
Pregnant pause	Blank	To take a pause for more than two second

Topology Coding

In this coding scheme, elements were numbered depending on how they were connected to other elements while being created. A source and a sink was identified for each element based on when the element was added in the model. Figure 7 shows a table of topology coding of P7 model. In Fig. 7, the first column under topology shows the source of the element, and the second column shown the sink. If the element is not connected to either a source or a sink at the time of addition, then the respective column under topology is left blank. If the environment is acting as the source or sink for the element, then a zero-value is used. Encoding the model in such a topology enables the future recreation of the model, in the sequence that the model was created by the designer.

Coding Scheme Inter-rater Reliability

An inter-rater reliability test was conducted for the element coding and activity coding processes using Cohen's kappa. If the kappa value is greater than 60%, agreement is considered substantial, whereas a kappa value of 80% or higher can be

Fig. 7 Example for topology coding

Elem ID	Elem Typ	Topology	
1	B	0	
2	BT	1	
3	E	1	0
4	B	3	0
5	BT	4	
6	E	4	0
7	B	6	0
8	BT	7	
9	B	0	
10	BT	9	
11	E	9	0

considered high agreement (Cohen 1960). For element coding, the average agreement between raters was found to be 96%, while the average agreement for activity coding was found to be 83%. Both kappa values suggest that the coding method is robust and yields a high agreement.

Protocol Study Results: Total Count of Variables

The activity of all the designers were coded according to the steps previously discussed. The data collected from the function modeling activity for each participant included activity coding, element coding (combined with topology coding), and time stamps. An activity graph was then generated to plot participant actions on a time sequence. The final model and the chronologically numbered element identifier (ID) are also stored for future use.

As shown in Table 3, the number of blocks and block texts are always equal each participant, while the number of edge and edge text are not. For P1, P2, P6, and P7, all four elements are equal which mean all the flows and the blocks were labeled. The number of edge text was always less than that for edges, except for P1 where the edge text was larger in number than edge. For P1, there are sixty edge text as compared to 43 edges. The model for P1 had two or more edge text for an edge in several instances, causing the increase in number of edge text when compared to edges.

The level of detail of the function structure can be associated with the total number of the functions and flows. The maximum number of block and block text was 15, whereas the maximum number of edges was 43. A review of the numerical results and each function model showed that P1 had the most detailed function structure of the participant pool in the protocol study. Conversely, the model for P6 was the least detailed with four block and block text, and twelve edges and edge text. The model in P8 has less edge text than P6, however, P8 drew color-coded edges with a legend.

Table 3 Number of block, block text, edge, and edge text for each participant

Participant	Element name			
	B	BT	E	ET
P1	15	15	43	60
P2	11	11	31	31
P3	7	7	20	6
P4	11	11	19	9
P5	5	5	21	16
P6	4	4	12	12
P7	8	8	13	13
P8	12	12	25	3
Total	73	73	184	150
Average	9.125	9.125	23	18.75
SD	3.758	3.758	10.128	18.683
Variance	14.125	14.125	102.571	349.071
Maximum	15	15	43	60
Minimum	4	4	12	3

The average difference between the edge and edge text is 15.25 and the occurrence of edge is 3.7 times more likely than edge text. This suggests that designers gave more importance in writing the function and block text instead of the flows as every block was labeled with a block text. This result contradicts the claims found in the previous pilot study and suggests that the designer gives more importance to labeling the functions when compared to flows.

A sign test was conducted to evaluate the differences between block and block text. The sign test showed no significant differences between block and block text as the number of blocks was always equal to the number of block text. For brevity, this test result is not presented here. A similar test was conducted to see the relation between the edge and edge text. The sign test hypothesizes that the median of differences between the edge and edge text will be zero. Table 4 shows the distribution of the signs across all the eight participants.

The sign test (Table 5) suggests there is no strong evidence to prove that the difference between edge and edge test is not equal to zero, as the total number of

Table 4 Sign test for E-ET

E	ET	E-ET	SIGN
43	60	−17	"–"
31	31	0	"+"
20	6	14	"+"
19	9	10	"+"
21	16	5	"+"
12	12	0	"+"
13	13	0	"+"
25	3	22	"+"

Table 5 Sign test results for E-ET

Frequencies		N
E-ET	Negative difference	1
	Positive difference	4
	Zero	3
	Total	8

Table 6 Signed rank test for E-ET

E	ET	E-ET	Sign	Abs value	Rank	Signed rank
43	60	−17	"–"	17	4	−4
31	31	0	N/A			
20	6	14	"+"	14	3	3
19	9	10	"+"	10	2	2
21	16	5	"+"	5	1	1
12	12	0	N/A			
13	13	0	N/A			
25	3	22	"+"	22	5	5
Positive						11
Negative						−4

positive difference is four. There should be a difference of at least 5 between edge and edge text to be considered significant.

Since the sign test neglects the number and takes into consideration only the sign, information could be lost. To avoid this, a signed rank test is conducted to check for the same difference between edge and edge text. Table 6 tabulates the result for the signed rank test.

The signed rank test shows that the test statistics value of four is less than the critical value required for this test of five. Therefore the test suggest that the null hypothesis be rejected which suggests that the difference is not zero and there is evidence to show that the edge occurs more than edge text.

Function Modeling Activity Coding Discussion

After examining the results, it can be concluded that the blocks are always labeled by a block text. However, only 81.5% of edges are being followed by edge text. This result suggests that the conclusion in the previous study (Sen and Summers 2012; Sen et al. 2010), that flows are perceived as more important than functions, may not be correct. To help understand the difference from the results of this research from that of the previous research, other patterns such as pauses and the activity around the pauses may need to be investigated before concluding if

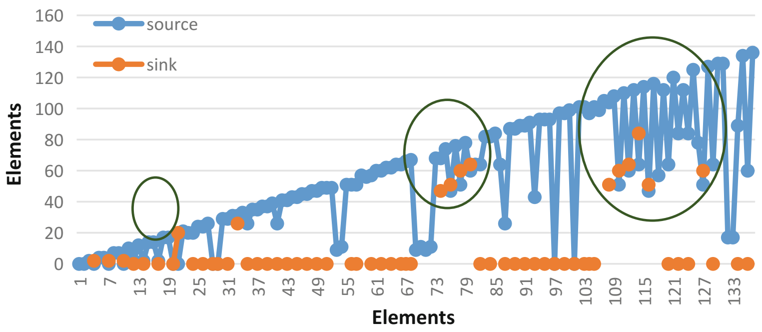


Fig. 8 Topology graph participant 1

functions are more important. Also, since quality of the solution was not expected from the designers, there is a possibility that not all of edges are labeled with an edge text.

Results for Chaining and Participants' Graphs

Each participants' activity and topology coding is shown. A chaining graph is then presented and can be compared with the general graphs. In comparing the general trend followed by P1 (Fig. 8) and the general graphs on forward, backward, and nucleation chaining, P1 relied on forward chaining. P1 also used nucleation chaining in 13 other instances. There was no evidence of backward chaining found in the model created by P1.

Similar topology graphs were created for the remaining seven participants. However they are not presented here for brevity. A summary of the model chaining results is presented in Table 7.

As shown in Table 7, most of the participants spend the bulk of their time using forward chaining to generate the model. P3 and P8 roughly used a third of their time doing nucleation, but the rest of the time was dedicated to forward chaining, and no backward chaining was seen. P7 had the highest amount of time spent in backward chaining at 10.4%. However, no nucleation was performed by P7 resulting in the majority of the time being spent forward chaining.

Activity Graph

The chaining of the function model showed that there is a pattern with designers using function models. Therefore, the similarity of the activities between designers

Table 7 Model chaining results for all participants

Participant	Forward chaining	Backward chaining	Nucleation
P1	89.6	0.0	10.4
P2	86.2	1.1	12.8
P3	72.5	0.0	27.5
P4	93.9	0.0	6.1
P5	85.7	0.0	14.3
P6	90.6	3.1	6.3
P7	89.6	10.4	0
P8	65.6	0.0	34.4
Average	84.2	1.8	14.0
Legend	Greater than 50%	Between 10 and 50%	Less than 10%

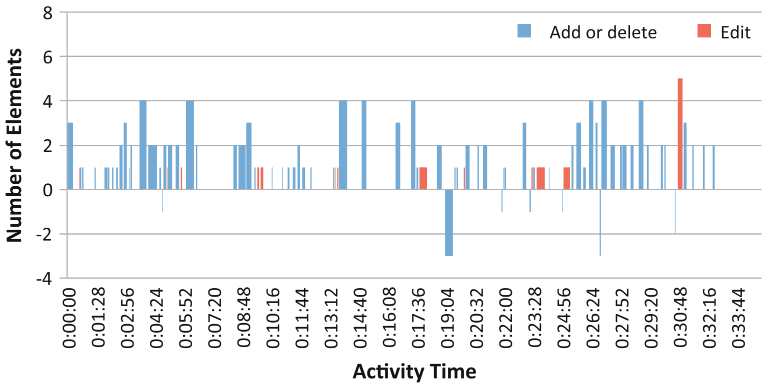


Fig. 9 Activity graphs for P1 plotted against the duration of the modeling activity

should also be investigated. In order to compare the participants, activity graphs were plotted for each participant. The activity graph for participant 1 can be seen in Fig. 9.

As seen in Fig. 9, the activity graphs were plotted with the duration of the modeling activity on the X-axis and the number of elements on the Y-axis. The variables plotted on the activity graph were the count of elements added or deleted, and the count of elements edited. The activity graphs for the remaining participants are not included for brevity. A summary of the observations from the activity graphs is provided in Table 8.

As shown in Table 8, besides participant 6, all participants generated the bulk of their model in the first two-thirds of the modeling time. The second and the final third mainly consisted of model refinements and some additions.

Table 8 Observations from the activity graphs

	Time	0–33% (time)	33–66% (time)	66–100% (time)
P1	33:00	11:00	22:00	33:00
		Larger addition	Fewer additions and larger pauses	Edits and addition
P2	18:00	06:00	12:00	18:00
		Larger addition	Larger addition and intermittent edits	Edits and addition
P3	07:00	02:20	04:40	07:00
		Larger addition	Larger pauses fewer addition	Larger pauses fewer addition
P4	07:00	02:20	04:40	07:00
		Larger addition	Larger addition and pauses	Edits and addition
P5	15:00	05:00	10:00	15:00
		Larger addition	Larger pauses fewer addition	Addition
P6	06:00	02:00	04:00	06:00
		Comparatively fewer addition	Comparatively fewer addition	Comparatively larger addition
P7	24:00	08:00	16:00	24:00
		Pauses and addition	Larger addition	Larger pauses fewer addition
P8	14:00	04:40	09:20	14:00
		Larger addition	Comparatively fewer addition	Comparatively fewer addition

Conclusions and Future Work

The figures from the element and activity analysis of the function model designers assisted in answering the two research questions posed in this paper. There are sufficient data to draw at least suggestive inferences. Results from these research questions are further discussed.

Are there patterns in designers using functional modeling as a design tool?

In reviewing the final models created by the participants, whenever a block was created in a function model, it also had an associated block text; however, not every edge had an associated edge text. With the exception of P1, the number of instances of edge text was always less than or equal to the number of edges. P1 had instances where multiple edge texts were associated with a single edge. This pattern also shows that the experience of the designer, though being similar, did have different results in the way the designer modeled the product.

Using the activity graphs, it can be pointed out that the designer adds much of their detail during the initial phases of modeling when compared to the later stages of modeling. The editing of elements also occurs predominantly towards the finishing stages of modeling.

Is forward chaining predominantly used?

Based on the topology observations, between eight participants, 84.2% of their time in the function modeling process was used for forward chaining, while only 1.8% of the duration was used for backward chaining. There were instances where the designers used nucleation in their modeling (an average of 14%), but it was not greater than forward chaining in any case. Therefore, it can be concluded that these participants used forward chaining predominantly.

Existing research on determining an ideal method to use function structures as a tool could help in directing further research towards the components of function structures (flows or functions). This future research could help explain whether designers perceive functions or flows as the major role in modeling. Moreover, if patterns exist with how designers perceive the components of function modeling, this can aid in improving existing tools for teaching function models to students as well as develop new tools that help students better apply their function modeling knowledge.

Appendix: Steps to Encode Video Data

There are six steps to be followed to code the designer's activity and they are explained in detail in the section below.

STEP 1: Run the video of the participant throughout to check which camera angle is best suited for the coding. Choose the best suited camera angle for coding. This could be different camera angle to avoid ambiguity while coding. The tools required for this step would be a video player that can be control the play back speed. The entire selection of video is subjective because the angle that one coder is comfortable is may not be easier for the other coder. So, the choice of the angle of the video could be according to each rater's comfort (Figs. 10 and 11).

STEP 2: Reproduce the final function structure or use the final picture of the functions structure to mark the elements in chronological order. In this stage there should be no ambiguity as the structure is exactly reproduced for the coding. The original function model is shown in Fig. 12 and the digitally reconstructed model is shown in Fig. 13.

STEP 3: Label all the elements using an element identifier in a chronological manner. In the Fig. 14, assuming the model is generated using forward chaining,

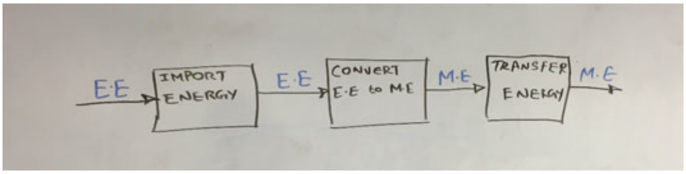


Fig. 10 Angle 1 from video

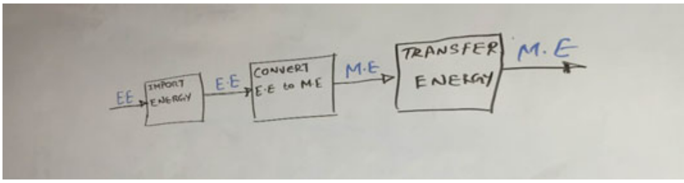


Fig. 11 Angle 2 from video

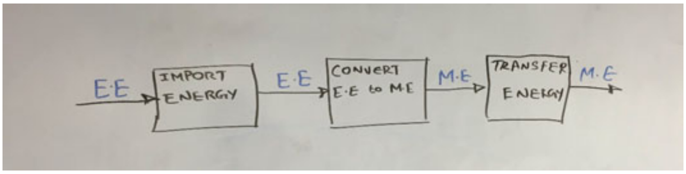


Fig. 12 Raw model

Fig. 13 Reconstructed model

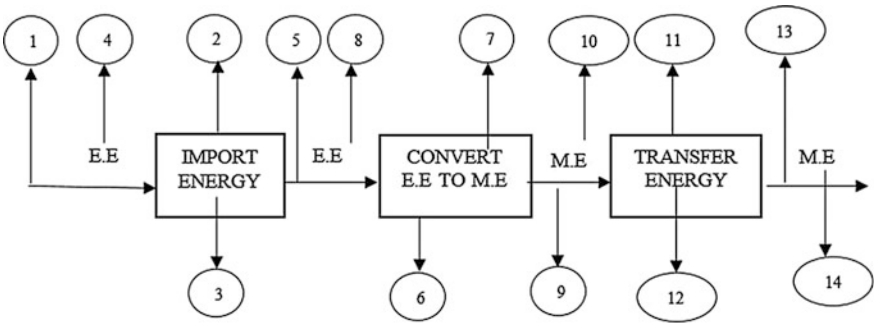


Fig. 14 Element Id numbering

the element ID is given to according to which elements are added first. If the edge is added first label is with element ID "1" and label the next element added as "2" and so forth. This information will be used to populate the element coding sheet in excel.

STEP 4: Use element coding and give the elements their respective element coding. Connect this information to the element ID (Table 9).

Table 9 Element coding for the model

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
CODE	E	B	BT	ET	E	B	BT	ET	E	ET	B	BT	E	ET

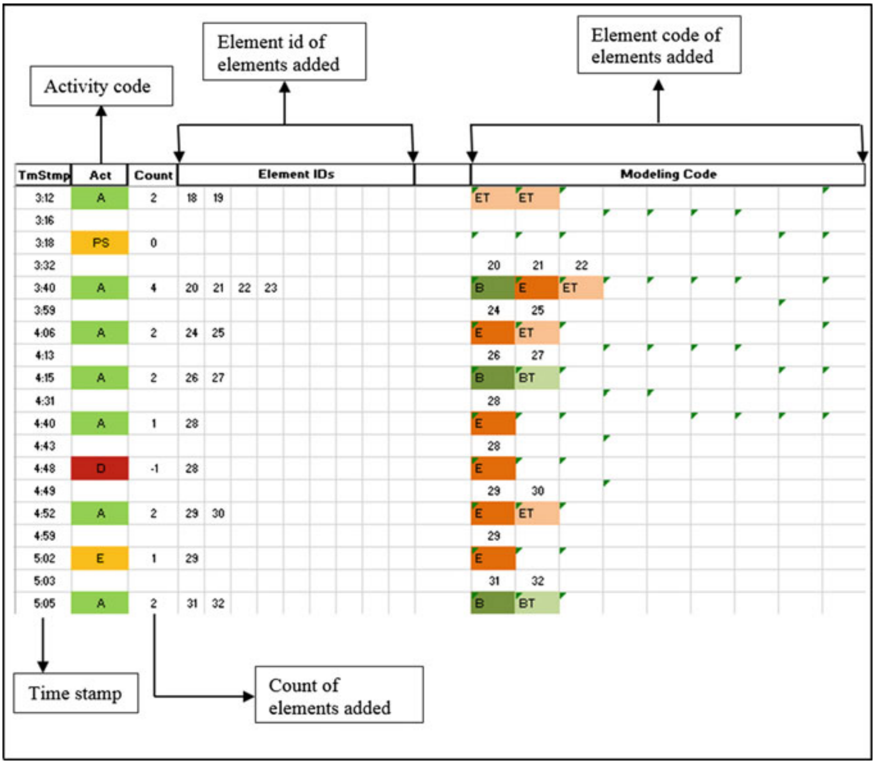


Fig. 15 Example of an activity coding sheet

STEP 5: Re run the video to focus on the activity of the designers. In this step, use the activity coding to code the activity of the designers. In addition, give time stamp to the activity too. Running the video at half speed has been found to have a good balance between efficiency and accuracy. This is also done to make sure that there is two or more seconds in pause in activity, which could be considered as a pause. Once the activity sheet is populated, it will look like Fig. 15.

STEP 6: Run the video for another time to take in details for topology of the function structure. Enter it into the topology chart in the element id sheet, which would say how that element was connected to the other element. This topology is going to be considered as soon as the element was added rather than in the end of the model. So the source or the sink of the element could be environment which is denoted by zero. A sample sheet is shown in the Fig. 16.

Elem ID	Elem Typ	Topology	
1	B	0	
2	BT	1	
3	E	1	0
4	B	3	0
5	BT	4	
6	E	4	0
7	B	6	0
8	BT	7	
9	B	0	
10	BT	9	
11	E	9	0
12	B	11	0
13	BT	12	
14	E	11	0

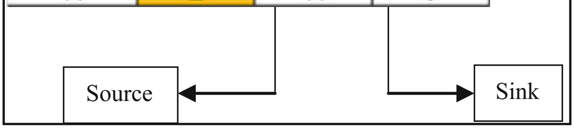


Fig. 16 Topology coding for participant 9

References

Albers A, Ohmer M, Eckert C (2004) Engineering design in a different way: cognitive perspective on the contact and channel model approach. In: Gero JS, Tversky B, Knight T (eds) Visual and spatial reasoning in design. Key Centre of Design Computing and Cognition, Cambridge, MA, USA

Cohen J (1960) A coefficient of agreement of nominal scales. Educ Psychol Meas 20(1):37–46

Fraenkel JR, Wallen NE (1993) How to design and evaluate research in education, 7th edn. McGraw-Hill, New York, NY

Hey J, Linsey J, Agogino AM, Wood KL (2008) Analogies and metaphors in creative design. Int J Eng Educ 24(2):283

Nagel RL, Bohm MR, Linsey JS (2013) An investigation into the effectiveness of an algorithmic approach to teaching functional modeling. In: Proceedings of the ASME 2013 international design engineering technical conferences & computers and information in engineering conference, 2013, pp 1–17

Ottosson S (1998) Qualified product concept design needs a proper combination of pencil-aided design and model-aided design before product data management. J Eng Des 9(2):107–119

Pahl G, Beitz W, Wallace K, Blessing L (2007) Engineering design: a systematic approach, 3rd edn. Springer, London

Ramachandran R, Caldwell BW, Mocko GM (2011) A user study to evaluate the functional model and function interaction model for concept generation. In: International design engineering technical conferences and computers and information in engineering conference, 2011, vol. 2011, no. 54860, pp DETC–47660

Schaafstal A, Schraagen JM, van Berl M (2000) Cognitive task analysis and innovation of training: the case of structured troubleshooting. Hum Factors J Hum Factors Ergon Soc 42(1):75–86

Sen C, Summers JD (2012) A pilot protocol study on how designers construct function structures in novel design. In: 5th International conference on design computing and cognition, 2012, No. 37

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

Thomas J, Sen C, Mocko GM, Summers JD, Fadel GM (2009) Investigation of the interpretability of three function structure representations: a user study. In: International design engineering technical conferences and computers and information in engineering conference, 2009, pp DETC2009–87381

Ullman DG (2010) The mechanical design process, 4th edn. McGraw-Hill, New York, NY

To Copy or Not to Copy: The Influence of Instructions in Design Fixation Experiments

Luis A. Vasconcelos, Chih-Chun Chen, Eloise Taysom and Nathan Crilly

Abstract Design fixation experiments often require participants to solve a design problem whilst being exposed to an example solution and instructions for how to treat that example. However, little is known about the influence of such instructions, leading to difficulties in interpreting results and understanding how the introduction of examples affects idea generation. In our experiment, participants were all provided with the same design problem and example solution, but were presented with different instructions, ranging from strongly encouraging copying the example to strongly discouraging copying. Analyses of participants' work indicated that only the instructions encouraging copying had an effect. When encouraged to copy, participants tended to only copy the structural features of the example rather than the underlying concept. By contrast, the number of features copied was not reduced when participants were discouraged from copying. These findings suggest that there are subtle interactions between instructions and stimuli that influence design fixation.

Introduction

Inspiration is vital to creative design. This has driven design researchers to conduct many studies into inspiration, for instance to find out what materials inspire designers (Gonçalves et al. 2014), how designers achieve inspiration (Zhao 2013), and how inspiration can improve designers' performance (Eckert 1997). Many of these studies have observed the use of external stimuli during idea generation, and have reported that whilst external stimuli can assist idea generation, they can also hinder it. This detrimental effect of inspiration is described in the design literature as

L.A. Vasconcelos (✉) · C.-C. Chen · E. Taysom · N. Crilly
University of Cambridge, Cambridge, UK
e-mail: lal40@cam.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_20

'design fixation' (Jansson and Smith 1991). Design fixation was originally studied in situations where the stimuli represented possible design solutions to the problem that was being addressed. It is often measured in the solutions that the designers propose according to the repetition of features from the stimuli that the designers see (Jansson and Smith 1991; Cardoso and Badke-Schaub 2011; Dahl and Moreau 2002; Linsey et al. 2010; Purcell and Gero 1996; Youmans 2011a, b). Although reproducing the features of good designs might be beneficial or efficient, the researchers have found that repetition is still present when the stimuli contain flawed inspiration sources, which they had thought designers would identify and avoid (Jansson and Smith 1991; Linsey et al. 2010; Youmans 2011b).

In general, fixation research suggests that blindly copying features from stimuli is harmful to idea generation, and therefore to the design process itself. As such, some studies have incorporated constraining textual instructions into the stimuli given to participants, to prevent them from copying features. However, the efficacy of the instructions seemed to vary across different accounts: they were effective in some cases (Chrysikou and Weisberg 2005; Yilmaz et al. 2010) and ineffective in others (Jansson and Smith 1991; Perttula and Sipilä 2007). Thus, it is still uncertain how textual instructions can influence the copying behaviour, and this uncertainty might be attributed to different factors. For instance, people may tend to overlook instructions presented along with example material (LeFevre and Dixon 1986), they can interpret the instructions in different ways if instructions are not strict, or wonder why they are being exposed to the stimuli and change their idea generation process accordingly. Whatever the reason might be, it is important to understand the relevance of the instructions provided to designers as part of the inspiration material. Methodologically, this would help in determining how fixation studies should be conducted and the results interpreted. More practically, it would also help to better understand how stimuli should be presented in software tools that aim to help idea generation by providing external stimuli to designers, and in other contexts in which designers are stimulated with examples (for such tools, see Chakrabarti et al. 2005; Linsey and Wood 2008; Vattam et al. 2011).

In order to better understand the influence of instructions on idea generation, we conducted an experiment in which participants in different treatment conditions were provided with the same design problem and the same stimuli, but the instructions they were given with respect to that solution differed between the groups. In reporting on that experiment we offer insights that are useful for interpreting the existing inspiration and fixation studies, and for designing new ones. These insights are also relevant for considering how inspirational material might best be presented to designers outside of experimental contexts, for example where computer tools are used to index and retrieve inspirational material.

Inspiration, Fixation and the Introduction of Inspiration Sources

Design researchers have been studying many different aspects of the inspiration process, including properties of the inspiration sources (e.g. quantity of sources) and aspects of the design process (e.g. total time available) (Sio et al. 2015). One of the stimuli characteristics that has been studied is whether good and bad examples would be copied indiscriminately. Researchers wanted to know whether designers would fixate on the examples they see and would copy them to some extent, irrespective of their quality. Indeed, a series of studies has found that participants still replicated features from previous examples even though they were flawed (Jansson and Smith 1991; Linsey et al. 2010; Youmans 2011b; Chrysikou and Weisberg 2005; Perttula and Liikkanen 2006; Viswanathan et al. 2014). To counteract this indiscriminate replication, some studies have tried warning participants about the flaws in the examples (Jansson and Smith 1991; Chrysikou and Weisberg 2005; Viswanathan et al. 2014), while others have tried instructing participants not to copy the examples (Perttula and Sipilä 2007; Smith et al. 1993; Yilmaz et al. 2010). Considering these two approaches, Chrysikou and Weisberg (2005) found that warning participants of flaws in the examples was not enough; they had to be told to avoid repeating those flaws. Yilmaz et al. (2010) also reported telling participants not to reproduce the examples and found that feature repetition was reduced. Conversely, Jansson and Smith (1991) and Perttula and Sipilä (2007) found fixation effects even when participants were instructed to avoid using features from the examples provided. As such, surveying the published literature reveals conflicting results relating to the use of instructions when providing stimuli to designers.

Whilst many variables have been manipulated in fixation experiments, and some studies have already tested the effectiveness of using textual instructions to some extent, the way the stimuli are introduced in such experiments has not yet been studied systematically. Such stimuli introductions can typically be divided into two components: a descriptive statement on what the stimulus is (e.g. “here is an example solution”) and a prescriptive instruction for how the stimulus should be used (e.g. “don’t copy its features”). Currently, the stimuli introductions (i.e. descriptions and instructions) given to participants vary from study to study. For instance, participants have been told that the example should be considered a solution for the given problem (Linsey et al. 2010), that it was provided to help them get started (Dahl and Moreau 2002), that it was there to raise thoughts (Liikkanen and Perttula 2010), and that it should be used to awaken thoughts but should not be reproduced (Perttula and Sipilä 2007) (also, sometimes the studies do not report how the stimuli were introduced). Such variation in the way the stimuli are provided can be attributed to a lack of agreement across studies about which ‘real world’ situation is being simulated (e.g. contexts in which examples are accidentally seen, intentionally searched for, or already known). Regardless of the reason, the variation in the way copying is encouraged or discouraged makes it difficult to know to what extent the design work is being influenced by such instructions. It could even be possible that designers will incorporate features from the examples provided irrespective of how constraining the instructions provided to them are.

To investigate the influence of stimuli instructions on idea generation, Smith et al. (1993) did an experiment in which participants were asked to either conform to or diverge from the example solutions provided in a creative task. Some participants (group 1) were told that the examples were great ideas previously created for that task, and that participants should create ideas like those whilst not copying them exactly. Other participants (group 2) were told that the examples restricted people's creativity, and that participants should create ideas that were different to the examples. Finally, other participants (control group) were given the same examples but without any instructions. It was found that when compared to the ideas generated by participants in the control group, those in group 1 *and* in group 2 generated more ideas containing features from the examples and that groups 1 and 2 were not different in this respect. Based on these results, the researchers proposed that the participants did not conform to examples because they had assumed that they should; they conformed because they could not forget the examples that they had seen. However, the instructions used in the experiment were not strict (i.e. they suggested how ideas should be created, but did not forbid or require the use of features from the examples, allowing participants to interpret the information in different ways) and both the description of the stimuli and the instruction for their use varied between groups, making it difficult to infer the influence of each piece of information in isolation. We believe that a different experimental setup could provide research with more data, complementing the results previously found and helping the field to clarify the influence of the textual instructions used in experimental research and professional design practice. To investigate this, we designed and conducted an experiment to find out the effects of instructions in design inspiration and fixation, be the instructions neutral or either—slightly or very—encouraging or discouraging.

Methodology

Objective and Hypothesis

This experiment investigates how textual instructions accompanying external inspiration sources can shape the design work of the participants. In particular, we hypothesise that instructions provided along with external sources will not influence the ideas that participants generate.

Experimental Method

Participants were randomly allocated to different experimental conditions. They were verbally asked to be creative and to design, individually, as many ideas as possible to a given problem. They were also asked to sketch and describe in writing their ideas on sheets of paper. Except for a control group that designed without any stimuli, all other experimental groups received a sketch of one example solution and a description of what the sketch represented. These stimulated groups received instructions for the use of features from the example, and these instructions varied with respect to how constraining or encouraging they were (see the materials section for the complete instructions). Finally, the authors assessed the participants' ideas to evaluate the influence of the instructions on the level of fixation demonstrated.

Participants

One hundred and sixty-eight first-year students in Engineering from the University of Cambridge, UK, were assigned to six experimental groups ($n = 28$). Participation in the experiment was part of the students' education, and was aimed at collecting data that could later be used to introduce them to the concept of design fixation. No demographic data was collected from the participants, but as first year undergraduate students they were broadly similar in age and design experience, drawn from a cohort with a male-female ratio of 3:1.

Task and Problem

The participants were told to solve the following problem. "Bicycles are a popular mode of transportation and recreation for many people. While growing up, a person might go through a series of ever-larger bikes, sometimes having several models, one after the other. However, having several bikes can be a problem for many reasons. Your task is to generate as many ideas as possible to eliminate the need to have multiple bikes as people grow up."

This problem was selected because it was expected to satisfy the following three criteria. First, it was unlikely that the participants had designed solutions to it before, although they were likely to have experienced the situation described in the problem previously (i.e. while growing up, they probably had multiple bikes), therefore helping their understanding of it. Second, the problem could be solved in many different ways, with many different underlying principles being applied, thus leaving enough room for creativity. Finally, both the design brief and the potential ideas held a low level of complexity, thus being suitable for a quick experiment fitting with the constraints of the course.

Procedure (Overview)

The experiment took place in a large lecture theatre with all the participants present. During the first 5 min, the participants listened to an oral explanation about the activities to follow and received all the material they needed. Participants in the five stimulated groups (SGs) received the design brief, the inspiration source, and blank sheets of paper, whilst participants in the control group (CG) received the brief without any inspiration source. Then, the participants were asked to think of ideas for 3 min without actually committing any designs to paper (because different participants had different materials and content, this ensured they all had enough time to read all the materials and start developing some ideas). Finally, for the remaining 10 min, all participants individually generated as many ideas as possible in silence, ideally including both a sketch and a written description for each idea.

Materials

All participants received the same design problem written on an A4 sheet, as well as blank A4 sheets to sketch and annotate their own ideas. Except for the participants in the control group, all participants received one additional sheet with an example solution, i.e. an annotated sketch of a bike (Fig. 1).

The example solution was preceded with the description: “Below is an example of how you should present your ideas (i.e. annotated sketches)”. This description was either immediately followed by an instruction regarding the use of features from the examples (constraining or encouraging) or by no instruction whatsoever. The instructions for the different experimental groups are listed below against a code for each experimental group.

- SG−2 (strictly forbidding): “*make sure you do not use features from this example in your own work*”.
- SG−1 (constraining): “*avoid using features from this example in your own work*”.
- SG0 (neutral): no instruction was given.
- SG+1 (encouraging): “*consider using features from this example in your own work*”.
- SG+2 (strictly requiring): “*make sure you use features from this example in your own work*”.

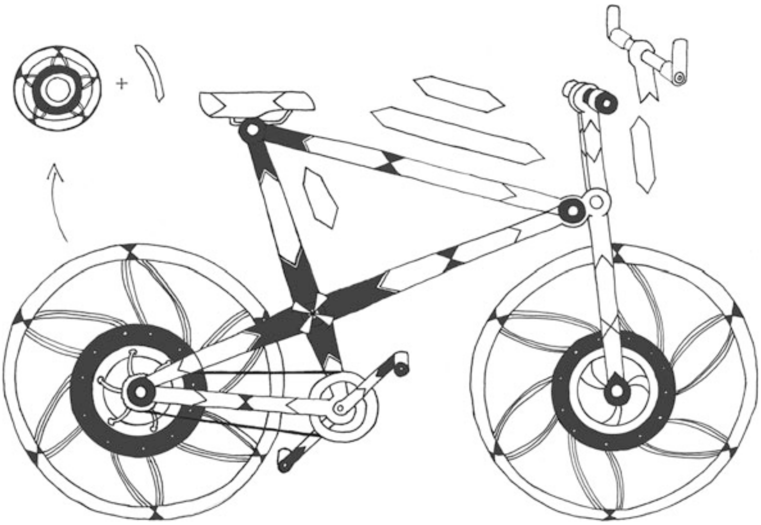


Fig. 1 Example solution provided to participants along with the following description. "A modular bike with parts of various sizes that can be connected and swapped to fit people with very different heights. Apart from the socketing parts and expansible/contractible wheels, the angles between tubes can also be modified in specific joints". The sketch used is a modification of the ECO 07 Compactable Urban Bicycle (ECO 2015)

Analysis

The assessment of the participant's ideas was conducted by the first three authors of this study, with backgrounds in design, computer science, and mechanical engineering respectively. Initially, the three evaluators agreed on the metrics to be included in the analysis. After that, the three evaluators analysed the design work of a random experimental group together in order to agree on the assessment method, ultimately reaching a consensus with respect to how to interpret and assess the ideas. Finally, each of the evaluators individually judged a subset of the remaining ideas. If any evaluator had trouble judging an idea, this idea was then discussed collectively. This interactive assessment method was intended to offer a reliable analysis of the ideas; but because there was no redundancy in the assessment, no inter-rater agreement could be calculated. We considered 'one idea' either to be a sketch or a written description (usually both) that presented an understandable way to solve the problem. Participants often generated more than one idea, but in some cases all ideas could be considered as one, particularly when the idea was a bike. For instance, if the ideas could all be incorporated onto the same bike without interference, then they were considered as a single idea. Conversely, if there were two or more ideas for the same bike component (e.g. frame, wheel, handle bar), then they were considered to be distinct ideas.

The metrics used in the assessment were 'idea fluency' and idea 'repetition'. Idea fluency is the total number of ideas generated, also called 'quantity' elsewhere (Shah et al. 2003). Idea repetition might happen in different levels, for instance, the repetition of idea types, conceptual features, or structural features. With respect to the idea type, we divided the ideas into two broad categories: bike and non-bike ideas, thus by designing a bike the participant would be repeating the idea type. With respect to their conceptual features, we also divided the ideas into two categories: modular or non-modular ideas, thus by designing a modular idea the participant would be repeating the conceptual feature. Finally, we examined the incorporation of structural features in the participants' ideas. These features were intentionally included into the example design in order to permit a measure of fixation. There were five structural features: swappable components to change bike size; frame joints (lugs) that act as sockets for the tubes; wheels with bendable spokes; an hourglass-shaped frame; and a saddle that cannot be adjusted in height directly.

Eight participants (4.8%) either did not generate any idea or generated ideas that could not be interpreted by the authors; the results from such participants were not included in our analysis. The adjusted number of participants per experimental group is indicated in the following section.

Results and Discussion

Idea Fluency

Instructions had no effect on the number of ideas generated. The data did not satisfy the assumptions for a standard ANOVA, therefore a non-parametic Kruskal–Wallis test equivalent to a one-way ANOVA was implemented instead. The results show that the number of ideas generated did not vary significantly across the five stimulated groups (SGs) ($H(4) = 2.63$, $p = 0.62$). However, there is a significant difference in the idea fluency between participants in these groups and those in the control group (CG) ($H(5) = 18.27$, $p < 0.01$), with participants in the control group generating a greater number of ideas. Additionally, although the majority of participants in the stimulated groups had an idea fluency of 1, participants in the non-constraining groups (SG, SG+1, and SG+2) had a higher frequency of fluencies greater than 1 (i.e. more participants in those groups generated more than one idea). However, this difference was not shown to be significant ($X^2(4) = 19.85$, $p = 0.47$). A significant difference in the frequency of idea fluencies was found between the stimulated groups and the control group ($X^2(5) = 55.38$, $p < 0.001$), with participants in the control group having a higher frequency of fluencies greater than 1. Table 1 shows summary statistics for these results.

Table 1 Summary of ideas generated per participant and ideas frequencies across groups

Generated ideas	SG−2	SG−1	SG0	SG+1	SG+2	CG
Mean (and SD) for the number of ideas per participant	1.50 (1.27)	1.30 (0.61)	1.48 (0.49)	1.41 (0.56)	1.54 (1.10)	2.39 (1.31)
Range of ideas per participant	1–6	1–3	1–4	1–4	1–6	1–5
Total number of ideas	39	35	40	38	43	67
Participants with 1 idea (and %)	21 (80.8)	21 (77.8)	16 (59.3)	19 (70.4)	19 (67.9)	11 (39.3)
Participants with 2 ideas (and %)	2 (7.7)	4 (14.8)	10 (37)	6 (22.2)	7 (25)	3 (10.7)
Participants with 3 ideas (and %)	1 (3.9)	2 (7.4)	0 (0.0)	1 (3.7)	0 (0.0)	7 (25)
Participants with 4 ideas (and %)	0 (0.0)	0 (0.0)	1 (3.7)	1 (3.7)	1 (3.6)	6 (21.4)
Participants with 5 ideas (and %)	1 (3.9)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	1 (3.6)
Participants with 6 ideas (and %)	1 (3.9)	0 (0.0)	0 (0.0)	0 (0.0)	1 (3.6)	0 (0.0)
Total number of participants	26	27	27	28	28	28

These results reveal that the idea generation rate was not influenced by how encouraging or constraining the instructions were. However, the presence of an example design affected idea generation: designing without exposure to stimuli resulted in more ideas being generated, which we interpret as a benefit of isolation from examples. This effect is consistent with other studies in which seeing an example caused reduction in the idea fluency (Linsey et al. 2010), although studies have also reported an increase in the idea fluency as a result from external stimulation (Purcell and Gero 1992) or even no effect at all (Jansson and Smith 1991). Finally, we should mention that although all groups were asked to present their ideas with both sketches and textual descriptions, the control group produced many ideas that were presented only in text. Thus, it is possible that the control group created more ideas because they did not spend their time sketching every idea.

Repetition of the Idea Type

Instructions had no effect on the number of bike ideas. The data did not satisfy the assumptions for a standard ANOVA, therefore a Kruskal–Wallis test was implemented instead. The results show that the number of bike ideas generated did not vary significantly across the five stimulated groups (SGs) ($H(4) = 2.98$, $p = 0.56$),

Table 2 Summary of bike and non-bike ideas generated across groups

Bike ideas	SG−2	SG−1	SG0	SG+1	SG+2	CG
Mean (and SD) for the number of bike ideas per participant	1.27 (0.92)	1.15 (0.36)	1.33 (0.83)	1.33 (0.55)	1.29 (0.53)	1.43 (0.84)
Range of bike ideas per participant	0–5	1–2	0–4	1–3	0–2	0–3
Number of bike ideas (and %)	33 (85)	31 (89)	36 (90)	36 (95)	36 (84)	40 (60)
Number of non-bike ideas (and %)	6 (15)	4 (11)	4 (10)	2 (5)	7 (16)	27 (40)

nor between these groups and the control group (CG) ($H(5) = 3.55$, $p = 0.62$). Consistent with these results, there is also no significant difference in the proportion of bike ideas generated (compared to non-bike ideas) across the five stimulated groups ($X^2(4) = 2.53$, $p = 0.64$). However, there is a significant difference in the proportions between these groups and the control group ($X^2(5) = 32.73$, $p = 0.001$), with participants in the control group having a greater proportion of non-bike ideas, such as other transportation means or policies to discourage the use and acquisition of bikes. Table 2 shows summary statistics for these results.

If we adopt a bike and non-bike categorisation for the ideas generated, these results reveal that the type of idea generated was not influenced by the instructions. Additionally, the presence of the bike example did not increase the number of bike ideas either. However, whilst the number of bike ideas was roughly the same for all groups, there was a large difference in the proportion of bike ideas between the control and stimulated groups. Only 60% of all ideas generated by the control group were bikes, whereas the stimulated groups had a much greater proportion of bike ideas (89% on average). The results indicate that bike ideas were equally likely to be generated irrespective of the experimental condition, but not seeing the example allowed participants from the control group to explore different areas of the solution space, again confirming the beneficial isolation effect. This effect is broadly consistent with other studies in which seeing an example caused participants to conform to certain types of solutions, thus reducing the diversity of ideas (Jansson and Smith 1991; Cardoso and Badke-Schaub 2011; Linsey et al. 2010).

Repetition of the Conceptual Feature

Instructions had no effect on the number of modular ideas. The data did not satisfy the assumptions for a standard ANOVA, therefore a Kruskal–Wallis test was implemented instead. The results show that the number of modular ideas generated did not vary significantly across the five stimulated groups (SGs) ($H(4) = 4.21$, $p = 0.38$). However, there appears to be a significant difference in the repetition of modularity between the stimulated groups and the control group (CG) ($H(5) = 11.40$, $p < 0.05$), with participants in the control group creating a greater number of modular ideas. When looking at the frequencies, there is no significant difference in the proportion of modular ideas generated (compared to non-modular ideas) across the five stimulated groups ($X^2(4) = 5.12$, $p = 0.27$), nor even when the control group is included in the comparison ($X^2(5) = 5.68$, $p = 0.34$). Table 3 shows summary statistics for these results.

If we adopt a modular and non-modular categorisation for the ideas generated, these results reveal that the type of idea generated was not influenced by the instructions. In fact, modular ideas were extremely rare across all groups. However, the results suggest that the presence of an example design affected idea generation and the control group created on average more modular ideas than the stimulated groups. This apparently surprising result can be attributed to a higher overall number of ideas generated by the control group, which is supported by there being no difference in the proportion of modular ideas generated among all groups. Still, the fixation literature would suggest that the stimulated groups would generate more modular ideas, an effect similar to the repetition of the idea type (i.e. bikes). In particular, participants in SG+1 and SG+2 were encouraged to use features from the example in their own work (and modularity was a visible feature on the example provided) but the results from those groups do not indicate that they acted accordingly. Similarly, participants in SG−2 and SG−1 were discouraged from using features from the example but generated as many modular ideas as the other groups. One possible explanation for this is that instructions have no influence on idea generation, as we had previously hypothesised. However, as there was no difference in the proportion of modular ideas between control and stimulated groups, we believe that the general principle of modularity (included in the example as a conceptual feature) was less obvious than the structural features and thus did not induce fixation effects.

Table 3 Summary of modular and non-modular ideas generated across groups

Modular ideas	SG−2	SG−1	SG0	SG+1	SG+2	CG
Mean (and SD) for the number of modular ideas per participant	0.08 (0.27)	0.3 (0.47)	0.19 (0.40)	0.19 (0.40)	0.18 (0.39)	0.43 (0.50)
Range of modular ideas per participant	0–1	0–1	0–1	0–1	0–1	0–1
Number of modular ideas (and %)	2 (5)	8 (23)	5 (13)	5 (13)	5 (12)	12 (18)
Number of non-modular ideas (and %)	37 (95)	27 (77)	35 (87)	33 (87)	38 (88)	55 (82)

Repetition of Structural Features

Instructions had a significant effect on the number of ideas that contained the structural features of the example provided. The data did not satisfy the assumptions for a standard ANOVA, therefore a Kruskal–Wallis test was implemented instead. The results show that the number of structural features incorporated into the participants' ideas varied significantly across the five stimulated groups (SG) ($H(4) = 41.62$, $p < 0.001$) and between these groups and the control group (CG) ($H(5) = 54.63$; $p < 0.001$), with participants in the encouraged groups (SG+1 and SG+2) incorporating a greater number of structural features. Consistent with these results, there is also a significant difference in the relative number of ideas with structural features across the stimulated groups ($X^2(4) = 37.32$, $p < 0.001$), with a higher number of features being associated with positive instructions to copy. There is also a significant difference between these groups and the control group ($X^2(5) = 49.74$, $p < 0.001$), with the control group's repetition rate being close to the neutrally stimulated group (SG0). Table 4 shows summary statistics for these results.

These results reveal that encouraging instructions influenced the repetition or incorporation of structural features, an outcome that does not support our hypothesis. On average, participants in SG+1 and SG+2 incorporated more features per idea and generated more ideas that incorporated any feature. Additionally, participants in the strictly forbidding (SG−2) and constraining (SG−1) groups produced results similar to the neutral (SG0) and control groups (CG). This result is partially consistent with research from Smith et al. (1993) but inconsistent with Chrysikou

Table 4 Summary of features incorporated into the participants' ideas and frequencies of ideas with features across groups

Feature	SG−2	SG−1	SG0	SG+1	SG+2	CG
Mean (and SD) for the number features incorporated per idea	0.08 (0.27)	0.17 (0.38)	0.10 (0.30)	0.71 (0.98)	0.93 (0.99)	0.13 (0.42)
Range of features incorporated	0–1	0–1	0–1	0–3	0–3	0–2
Ideas with 0 features (and %)	36 (92)	29 (83)	36 (90)	22 (58)	19 (44)	60 (90)
Ideas with 1 features (and %)	3 (8)	6 (17)	4 (10)	8 (21)	11 (26)	5 (7)
Ideas with 2 features (and %)	0 (0)	0 (0)	0 (0)	5 (13)	10 (23)	2 (3)
Ideas with 3 features (and %)	0 (0)	0 (0)	0 (0)	3 (8)	3 (7)	0 (0)
Ideas with 4 features (and %)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Ideas with 5 features (and %)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Ideas with features incorporated (and %)	3 (8)	6 (17)	4 (10)	16 (42)	23 (56)	7 (10)

and Weisberg (2005), as encouraging instructions increased fixation, whereas constraining instructions did not decrease it. However, contrary to the results from Smith et al. (1993), our constrained groups did not replicate more features than a control group, contradicting the idea that participants could not forget the example they had seen. It seems that a few structural features were naturally likely to be incorporated into the ideas generated (supported by the similar results from SG−1, SG0 and CG), with SG+1 and SG+2 incorporating more features because instructed to do so. When comparing the results from the repetition of structural features to the repetition of the conceptual feature used in this study, we infer that concrete structural features can be more easily copied or can fixate more than abstract conceptual features from examples. This is similar to what has been suggested in previous studies (Zahner et al. 2010; Cheong et al. 2014; Feng et al. 2014). Again, we should highlight that the control group produced proportionally less sketches than the other groups. This puts the other groups in an unfavourable position with respect to the count of structural features incorporated, since these groups had to represent a shape of the bike in which repetition could be more easily recognised—it is difficult to identify structural repetition when the idea is represented only by text—thus possibly biasing the results.

Study Limitations

The main limitations of this study involve the duration of the generation session, the pool of participants chosen, the design problem used, and the assumptions for inter-rater agreement. These limitations are discussed next.

The idea generation session in this study was 10 min long, which can be considered short when compared to other fixation studies in which generation sessions lasted for 30 or 60 min (Vasconcelos and Crilly 2016), and shorter still compared to professional practice (Crilly 2015). Also, past research suggests (Kudrowitz and Dippo 2013), novel ideas tend to occur later in the idea generation session. As a result, the short session adopted for this study might have contributed to inflated fixation scores.

The participants in this study were undergraduate students and the generation session was part of an ongoing engineering course. This might have resulted in a more diligent participant behaviour when compared to other studies in which participants and experimenters did not have a student-lecturer relationship. As a result, the setup adopted for this study might have contributed to an increased participant adherence to the instructions. The design problem used in this study was chosen because it was unlikely that the participants had designed solutions to it before. As such, it is possible that using familiar problems will produce different results, although research has demonstrated that fixation effects can be observed with both familiar and unfamiliar problems (Jansson and Smith 1991).

Finally, in this study we did not measure the inter-rater agreement for the assessment, a test which is often performed for similar studies. Such measurement is important to demonstrate that the reliability of the evaluators' assessment when working individually. Whereas the assessment performed in this study involved many interactions between evaluators (thus we expected a high agreement between evaluators), we cannot quantify how good this agreement might have been, or even if there were varying levels of agreement for the different ideation metrics used in this study.

Conclusion and Future Work

In this study we have tested the influence of instructions on idea generation. In particular, we analysed how instructions may affect the number of ideas generated and the repetition or incorporation of the example or its parts into the participants' ideas. The instructions used were provided along with an external stimulus and its description. It is important to differentiate the descriptions from instructions because in this study we have controlled the former but manipulated the latter. We found that instructions had some influence on the idea generation of our participants. When asked to use features from the example, participants copied structural features but failed to copy a more abstract conceptual feature. When asked not to use features from the example, however, most participants did not reduce the number of features copied. This result allows us to infer that more concrete features are easier to recognise—and thus reproduce—than more abstract features, such as modularity. Also, it might indicate that positive instructions are more effective than negative ones, which can tell us how to frame future instructions, whether that is with respect to experimental stimuli in research or inspirational stimuli in design practice.

Irrespective of how constraining the instructions were, participants exhibited fixation effects due to their exposure to the example design (in comparison to the control group, all stimulated groups created more ideas of the same type as the example provided). This result is in line with many other design fixation experiments in which participants become stuck on a particular idea type. However, it is important to emphasise that the description of the stimulus itself could also be causing the fixation effect as the stimulus was presented to the participants as "*an example of how they should present their ideas*". Thus, perhaps there is an implicit suggestion for the participants to produce ideas similar to the example, i.e. a bike. Future studies could investigate such possibility and complement our understanding about stimuli introduction with the influence of the descriptions provided to designers.

In this study we observed that some results might have been influenced by participants communicating their ideas in different ways (some of them drawing, some writing, and some doing both), an issue that cannot be confirmed until the analysis of the results. In inspiration and fixation studies, participants are asked to provide their ideas according to a content template, but that is rarely compulsory. It is fundamental that all ideas contain the same elements (e.g. text and sketch) in order to be analysed and compared, so studies should make sure every idea has a similar content. Additionally, future studies could investigate the difference between providing written, pictorial or spoken instructions to designers, but also how they assimilate instructions. For instance, it is possible that people need instructions about the instructions (i.e. “read the instructions carefully”), as one way to make sure that they will read and fully understand what is required from them.

Finally, when considering design practice, the results reported here are relevant to how inspirational stimuli should be framed when presented to designers. This is particularly important for the development and implementation of computer-aided design tools that provide designers with external stimuli. Much has been researched on how such software tools might be structured and interacted with, and what form the inspirational stimuli should take (Töre Yargin and Crilly 2015; Shneiderman 2000). However, it is also important to understand how those stimuli should be introduced, whether by description, instruction, or both. Should designers be steered towards or away from the repetition of structural features, directed to identify conceptual features, or simply left alone to interpret and respond as they see fit? By developing a better understanding of how stimuli instructions influence idea generation, we will move closer to answering such questions and thereby be more capable of supporting design activities.

Acknowledgements The authors would like to thank Carlos Coimbra Cardoso for commenting on earlier drafts of this paper, to all the students for taking part in the experiment, and to the three DCC reviewers for improving this work with their feedback. This work was supported by the CAPES Foundation, Ministry of Education of Brazil, under grant/process: BEX 11468/13-0 and by the UK’s Engineering and Physical Sciences Research Council (EP/K008196/1). Research data supporting this publication is available from the DSpace@Cambridge repository at the web address given below. It contains data from the participants and from our analysis. The data from the participants consists of all the annotated sketches that participants generated in the experiment. The data from the analysis consists of the evaluation for all ideas generated by the participants in the experiment. https://www.repository.cam.ac.uk/handle/1810/254702

References

Cardoso C, Badke-Schaub P (2011) The influence of different pictorial representations during idea generation. J Creat Behav 45:130–146

Chakrabarti A, Sarkar P, Leelavathamma B, Nataraju BS (2005) A functional representation for aiding biomimetic and artificial inspiration of new ideas. Artif Intell Eng Des Anal Manuf 19:113–132

Cheong H, Hallihan G, Shu LH (2014) Understanding analogical reasoning in biomimetic design: an inductive approach. In: Gero JS (ed) Design computing and cognition’12. Springer, Dordrecht, Netherlands, pp 21–39

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 31:1134–1148

Crilly N (2015) Fixation and creativity in concept development: the attitudes and practices of expert designers. Des Stud 38:54–91

Dahl DW, Moreau P (2002) The influence and value of analogical thinking during new product ideation. J Mark Res 39:47–60

Eckert CM (1997) Design inspiration and design performance. In: Proceedings of the 78th world conference of the textile institute, pp 369–387

ECO 07-Compactable Urban Bicycle. In: Behance. https://www.behance.net/gallery/293563/ECO-07-Compactable-Urban-Bicycle. Accessed 10 Nov 2015

Feng T, Cheong H, Shu LH (2014) Effects of abstraction on selecting relevant biological phenomena for biomimetic design. J Mech Des 136:111111

Gonçalves M, Cardoso C, Badke-Schaub P (2014) What inspires designers? Preferences on inspirational approaches during idea generation. Des Stud 35:29–53

Jansson DG, Smith SM (1991) Design fixation. Des Stud 12:3–11

Kudrowitz B, Dippo C (2013) Getting to the novel ideas: exploring the alternative uses test of divergent thinking. In: ASME 2013 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers, pp V005T06A013–V005T06A013

LeFevre J-A, Dixon P (1986) Do written instructions need examples? Cogn Instr 3:1–30

Liikkanen LA, Perttula M (2010) Inspiring design idea generation: insights from a memory-search perspective. J Eng Des 21:545–560

Linsey J, Wood K (2008) Wordtrees: a method for design-by-analogy. In: ASEE annual conference

Linsey JS, Tseng I, Fu K et al (2010) A study of design fixation, its mitigation and perception in engineering design faculty. J Mech Des 132:041003

Perttula MK, Liikkanen LA (2006) Exposure effects in design idea generation: unconscious conformity or a product of sampling probability. In: Proc Nord, pp 42–55

Perttula M, Sipilä P (2007) The idea exposure paradigm in design idea generation. J Eng Des 18:93–102

Purcell AT, Gero JS (1992) Artificial intelligence in design conference 1991 special issue effects of examples on the results of a design activity. Knowl Based Syst 5:82–91

Purcell AT, Gero JS (1996) Design and other types of fixation. Des Stud 17:363–383

Shah JJ, Smith SM, Vargas-Hernandez N (2003) Metrics for measuring ideation effectiveness. Des Stud 24:111–134

Shneiderman B (2000) Creating creativity: user interfaces for supporting innovation. ACM Trans Comput Hum Interact 7:114–138

Sio UN, Kotovsky K, Cagan J (2015) Fixation or inspiration? A meta-analytic review of the role of examples on design processes. Des Stud 39:70–99

Smith SM, Ward TB, Schumacher JS (1993) Constraining effects of examples in a creative generation task. Mem Cogn 21:837–845

Töre Yargin G, Crilly N (2015) Information and interaction requirements for software tools supporting analogical design. Artif Intell Eng Des Anal Manuf 29:203–214

Vasconcelos LA, Crilly N (2016) Inspiration and fixation: questions, methods, findings, and challenges. Des Stud 42:1–32

Vattam S, Wiltgen B, Helms M et al (2011) DANE: fostering creativity in and through biologically inspired design. In: Taura T, Nagai Y (eds) Design creativity 2010. Springer, London, pp 115–122

Viswanathan V, Atilola O, Esposito N, Linsey J (2014) A study on the role of physical models in the mitigation of design fixation. J Eng Des 25:25–43

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 24:335–355

Youmans RJ (2011a) Design fixation in the wild: design environments and their influence on fixation. J Creat Behav 45:101–107

Youmans RJ (2011b) The effects of physical prototyping and group work on the reduction of design fixation. Des Stud 32:115–138

Zahner D, Nickerson JV, Tversky B et al (2010) A fix for fixation? Rerepresenting and abstracting as creative processes in the design of information systems. Artif Intell Eng Des Anal Manuf 24:231–244

Zhao M (2013) Seek it or let it come: how designers achieve inspirations. In: CHI13 extended abstracts on human factors in computing systems. ACM, pp 2779–2784

Part VI
Design Processes

A Self-Organizing Map Based Approach to Adaptive System Formation

Dizhou Lu and Yan Jin

Abstract Multi-agent systems are considered to be potential solutions to complex tasks. Cellular self-organizing (CSO) multi-agent systems have been proposed that take a field-based approach to regulate agent behaviors. One difficulty in designing CSO systems is to generate rules to map given tasks to agent behaviors. This paper proposes an approach for adaptive system formation based on a field analysis and self-organizing map (SOM) algorithm. The tasks are captured as multiple task fields. The relationship among the agents is translated into a social field. Each agent has multiple function modes corresponding to the task fields. SOM and a function mode selection algorithm are devised to match the social field of the system with the task fields. Computer simulations have demonstrated the effectiveness of this approach and its potential in designing CSO systems for solving system formation tasks.

Introduction

When changes in the task requirement and operation environment occur, a system often needs to change its formation (e.g., form/shape, size, or structure) in order to stay functional. For example, a sophisticated rescue robotic system must be able to change its shape when the space on the path varies with the harsh and unpredictable environment. Space and deep sea explorations are also examples of such variable environments. In most, if not all, engineered systems, the physical components are designed for only limited and predetermined purposes and operation ranges, beyond which the system behaviors are not predictable.

Multi-agent systems are potential solutions to complex tasks. The flexible relationships among the agents provide the system adaptability to deal with changing functional requirements and operation conditions. In order to guide a multi-agent system to fulfill a task, the global task information needs to be encoded

D. Lu · Y. Jin (✉)
University of Southern California, Los Angeles, USA
e-mail: yjin@usc.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_21

into local rules for agents to follow. The adaptability of the system depends on the encoding process and the range of applicability of the local rules. Most of current approaches encode the task information into agent behaviors. In order to generate suitable rules for given tasks, designers must either specify rules based on their knowledge, or provide algorithms like genetic algorithm to optimize the parameterized rules. In both cases, external supervision is needed, limiting the system adaptability.

Our previous field based behavior regulation approach to cellular self-organizing (CSO) systems (Chen and Jin 2011; Humann et al. 2014) separates the task encoding process from the system model design process by translating the tasks and environment information into a task field, and the agents operate in the task field by moving from the “higher place” to the “lower place” in the field. The current approach, however, has several limitations, making it difficult to deal with system formation problems. First, in the earlier field based approach (Chen and Jin 2011), each agent operates independently based on its own sensed field information. The system formation is largely an emergent result. Although the emergence offers adaptability, the lack of explicit relations between the agents makes it hard to generate a specific formation, such as a tube or a ring, when the task demands it. Second, in the later dynamic social structuring approach (Khani and Jin 2015) agents form local social structures based on the predefined social rules. Although the introduction of social rules has made it possible to generate local forms and increase the global productivity, the coding of rules can be domain and designer dependent. Third, the approaches to CSO systems developed thus far generally lack learning capability, making it difficult to deal with complex system formation problems where changes need to be learned by the agents during the process of operation.

In this paper, a self-organizing map (SOM) based approach is proposed for CSO system design. The goal here is to apply unsupervised learning and build a system model which is compatible with the task fields generated by the field based behavior regulation. Each agent in the system is treated as a neuron in a SOM, with the ability to sense the environment and communicate with other agents. Tasks are represented as multiple task fields external to the system, and the cooperation among the agents is represented as the social field of the system. The system has the ability to organize itself to fulfill system formation tasks via matching its social field with the external task fields. In the rest of this paper, recent research related to multi-agent systems, field-based regulation, and shape formation is reviewed in second section together with a short introduction to self-organizing map. The adaptive system formation problem and the framework of our approach are introduced in third and fourth sections, respectively. Three case studies are described in fifth section to demonstrate the effectiveness of our approach. Final section draws concluding remarks and points to future research directions.

Related Work

In the past decade, design for adaptability has been popular in the field of engineering design and more efforts have been made to investigate multi-agent system design. A multi-agent system consists of multiple robotic agents that are supposed to work with each other to fulfill global level tasks based on their local information. The design of multi-agent system is usually a bottom-up process, with a focus on developing a suitable interaction model of multiple agents. Two types of approaches are commonly used in building such models.

The structural design approach is initially inspired by social structures and bio structures in nature. Fukuda and Ueyama (1992) compared robotic system with social system, and pointed out the importance of system structure for the intelligence of the system. Kawauchi et al. (1992) further proposed the "CEBOT" with genetic knowledge production algorithm that achieved self-organization and self-evolution with a "distributed intelligence system".

Also inspired by social structures, we introduced the concept of field based behavior regulation, in which agent behaviors are regulated by both social field and task field (Chen and Jin 2011; Humann et al. 2014). The task field is used to represent the task and the environment for the system. The social field is formed by introducing social rules in multi-agent systems (Khani and Jin 2015). This dynamic structure can enhance the self-organizing functionality for multi-agent systems using both general and context-based social rules among the agents in the system.

Behavior modeling and optimization is another approach to designing agent interaction model. Our previous work proposed a COARM behavioral model (Humann et al. 2014) that specifies agent behavior as the composition of five primitive behaviors: cohesion, avoidance, alignment, randomness and momentum. A genetic algorithm is used to search for the optimal composition for different tasks such as foraging and habitat construction.

Researchers have applied multi-agent systems to solve shape and structure formation problems. Nagpal et al. (2006) developed an algorithm for multi-agent systems to form arbitrary predefined shapes. A compiler is used to translate global shape information to local agent rules. Bai and Breen (2008) proposed a cell aggregation model to achieve self-organizing shape formation. In this model, an artificial scalar field is introduced for cells to move in the direction of the field gradient. A genetic Programming method is used to discover the relationship between different field functions and the shapes formed by the cells. Doursat (2012) presented a model using non-random genetic rules to achieve self-assembly and pattern formation. In his approach, self-assembly and pattern formation are integrated in loops to form complex shapes or structures. Werfel (2010) proposed a 3-dimensional model to implement low-level primitives shown in biological system. In his model, the gradient of morphogen field is used to guide modular robots to grow into structures of desired sizes, and provide them with position information.

De Rosa et al. (2006) proposed a shape formation algorithm for lattice-arrayed modular robots. In their algorithm, the desired shape is compiled into a plan which guide the movement, creation and deletion of the void space in the lattice, transforming the lattice into the desired shape. Tolley and Lipson (2010) presented an approach for stochastic assembly of modular robots. The modular robots are put in a fluidic tank, and they can self-reconfigure into different 3-dimensional shapes depending on the fluidic environment.

The current multi-agent system formation approaches mostly involve supervised learning such as genetic programming or compiling that requires pre-knowledge of the target forms. To deal with the uncertainty in the real world, unsupervised learning is more preferable since the information about the environment and the needed formation can be unknown and human interventions to the system are not possible. Self-organizing map is an unsupervised artificial neural network developed by Kohoen (1990). It has been successfully applied in the areas including image processing and speech recognition (Galda et al. 2004; Zhang et al. 2014).

With a high degree of topological order, SOM can be used in multi-agent systems for structure design through organizing the relationships among the agents. Although there is little direct research to apply SOM to multi-agent systems, some work showed how SOM could be used to solve inner-system relationships and physical pattern matching problems. Kiang et al. (1995) applied SOM as a clustering tool in group technology, where SOM was used to address the part machine relationships for grouping the parts. Kit et al. (2014) developed the location aware self-organizing map to discover visual features of geographical locations.

One major barrier for applying SOM in multi-agent systems is to map the codebooks (related to the social field discussed below) to the data space density (related to the task field discussed below). This problem is represented as magnification control in vector quantization. Villmann and Claussen (2006) explored different methods to control the magnification in SOM with regard to the typical SOM learning rules, and summarized them into three types. Localized learning algorithms introduce local learning rates and can achieve arbitrary magnification. The other two types of magnification learning algorithms are winner-relaxing learning and concave–convex learning (Claussen and Villmann 2005; Zheng and Greenleaf 1996). Compared with localized learning, winner-relaxing learning and concave–convex learning have the advantage of independence on the data distribution (Villmann and Claussen 2006).

The Problem: Adaptive System Formation

In this research, the problem of adaptive system formation is considered as a kind of design problem, in which a set of task properties or requirements is given and a system is formed to accomplish the task. The difference between this problem and the traditional design problems is that the system to be designed must be able to sense the changing task requirements and environments and adaptively form and configure itself in response to the changes. Therefore the system formation problem in this sense is a meta-design problem: it is about designing a self-design (i.e., formation or configuration) mechanism.

Examples of system formation include shape formation (Bai et al. 2008), structure and topology self-configuration (Ferguson and Lewis 2006), and more recently programmable matter (Derakhshandeh et al. 2015). Depending on the tasks, various approaches can be applied to solve system formation problems. In many robotic applications, sophisticated *mechanical mechanisms* are developed to facilitate dynamic formation of systems. In an *evolutionary approach*, a system evolves its formation overtime through generations of computational evolution. When system performance measures are difficult to obtain, *unsupervised learning approaches* can be effective for systems to attain their needed formations.

The long term goal of our research is to devise mechanisms that can allow systems to adapt to changing tasks and environments by changing and evolving system formation of shapes, structures, and functional components autonomously without human intervention. As the first step toward this goal, in this paper, we apply an unsupervised learning approach, self-organizing map (SOM), to shape formation problems.

The adaptive system formation framework is developed based on our previous work on cellular self-organizing (CSO) systems (Chen and Jin 2011; Humann et al. 2014; Khani and Jin 2015). As shown in Fig. 1, in this framework, task requirements together with environmental constraints are transformed into one or multiple task fields by the agents. At the same time, the agents sense each other and generate a system formation that is supposed to match the task fields. A SOM algorithm is introduced to do the mapping and update the system formation based on the mismatch. The following section provides details of this approach.

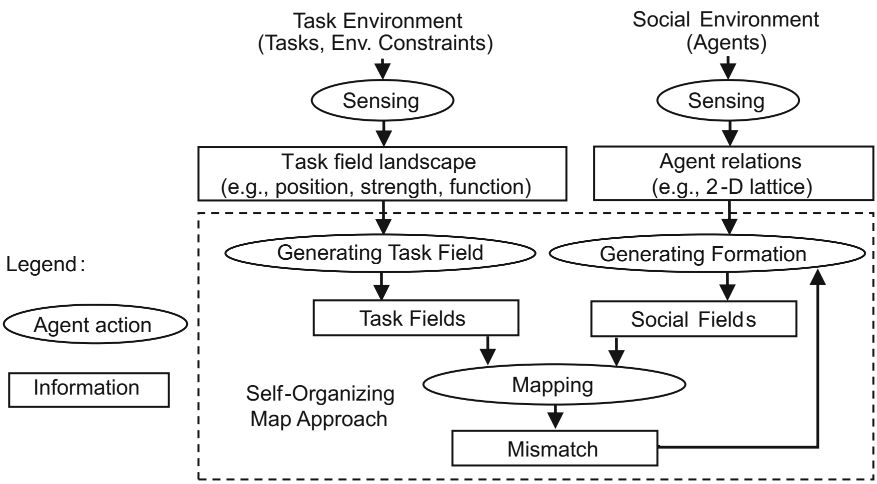


Fig. 1 A proposed system formation approach

A SOM Based System Model

System and Agents

Following the two dimensional SOM modeling convention, a system *Sys* is defined by $k \times k$ *agents* that form a lattice network topology.

Definition 1 (*System*)

$$Sys = \begin{bmatrix} Agent(1,1) & \cdots & Agent(1,k) \\ \vdots & \ddots & \vdots \\ Agent(k,1) & \cdots & Agent(k,k) \end{bmatrix}$$

where $Agent(a,b)$: *Agent* positioned in (*a, b*) in the network.

In the two-dimension model, the task fields are represented in a standard 2-D space (x,y), $0 \le x \le 1$ and $0 \le y \le 1$, which is also called *working space*. The system formation in this case is against this space. Therefore, each agent, in addition to the network position (*a, b*), holds an attribute of *position in the task field*: (x,y). Furthermore, in order to differentiate between different task fields that demand different agent functions, we introduce another agent attribute called *function-mode: f,* which can take an integer value to indicate a specific function mode of the agent. Therefore, we have three more agent attributes in addition to (*a, b*): $Agent(a,b;x,y;f)$.

Task Field

In the CSO framework, tasks together with environmental situations are represented as "task fields" in which agents seek and move to attractors (Chen and Jin 2011). In this research, the distribution of "attractors" represents the tasks requirements. The attractors can be singular or can be evenly distributed over an area. We introduce the concept of "task field strength" to capture the attractors. Since there can be multiple tasks involved in a single application, each task should have its own "task field strength distribution." In the special case of two-dimensional task field space, the field strength distributions of two tasks can be "parallel" meaning that the two tasks can be performed concurrently or "sequential" meaning only one can be performed at a time. We have the following task definitions.

Definition 2 (*Task field strength distribution*)

$$TF_i(x,y) = tFLD(Env, task_i)$$

where *Env* is the environment for the system to perform $task_i$, and *tFLD* is a task field formation operator for generating the field strength distribution of the task.

In case of multiple task fields, the overall task field strength at a given time and position in the field depends on whether the tasks are parallel or sequential. For example, an agent can have the tasks of moving towards a target, and moving away from an obstacle at the same time. Then the overall task field strength distribution can be calculated as:

$$TF_{overall}(x, y) = \sum \omega_i(x, y) * TF_i(x, y)$$

where ω_i is the weight function for task field i.

On the other hand, an agent can have the tasks of searching for food, and carrying food back to the nest. However, the agent can address only one of the two tasks at a time. In such case, the overall task field can be represented by the following equation:

$$TF_{overall}(x, y) = \begin{cases} TF_1(x, y), & Function\,Mode1 \\ TF_2(x, y), & Function\,Mode2 \\ & \vdots \end{cases}$$

In most cases, both of the above two methods are needed to add all task fields together. Thus in a more general form, the overall task field can be represented combining Definitions 2 and 3.

Definition 3 (*Overall task field strength distribution*)

$$TF_{overall}(x, y) = \begin{bmatrix} \sum \omega_{1i}(x, y) * TF_i(x, y) \\ \sum \omega_{2i}(x, y) * TF_i(x, y) \\ \vdots \\ \sum \omega_{mi}(x, y) * TF_i(x, y) \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \dots & \omega_{1n} \\ \omega_{21} & \omega_{22} & \dots & \omega_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{m1} & \omega_{m2} & \dots & \omega_{mn} \end{bmatrix} \begin{bmatrix} TF_1 \\ TF_2 \\ \vdots \\ TF_n \end{bmatrix}$$

where n is the number of all the tasks for the system, and m is the number of the function modes for each agent.

Social Field

Solving the adaptive system formation problem with a CSO systems approach requires the agents in the system to work together to update their formation in response to the needs and changes of the tasks. As shown in Fig. 1, a specific system formation in our approach is manifested by its corresponding social field. Similar to the task field, a social field can be characterized by its “social field strength” and “social field strength distribution” over the task field space (x, y), $0 \leq x \leq 1$ and $0 \leq y \leq 1$. As Fig. 1 indicates, the mapping between the social field and the task fields provides guidance for the agents of the system to adjust their formation in order to finally achieve minimum mismatch with the task fields.

Our previous work has shown that social structuring among individual agents plays a very important role for self-organizing systems to achieve desired performance (Khani and Jin 2015). In this research, the social structuring is achieved in two ways (see Fig. 1). First, the "Agent Relations" are predefined between the agents. As shown in Definition 1, a 2-dimension lattice network topology is employed in this paper for modeling agent relations. Each agent (*a, b*) has 2–4 connections depending on whether (*a, b*) is located in the corner, edge or center part of the network. These connections form the neighborhood of agents. During the process of system formation and update, neighbors influence each other to maintain a desirable formation for given tasks. Second, a *social field strength operator* is introduced for agents to assign and evaluate the social field strength at any given position (*x, y*) in the task field space. Based on the SOM algorithm, an ideal system formation is the one that has the compatible "social field strength distribution" with the given "task field strength distribution." Mismatch between the two leads to agents in the system working with their neighbors to adjust their system formation toward better ones. We introduce the following definitions.

Definition 4 (*Social Field Strength Distribution for function model j*)

$$SF_j(x, y, t) = sFLD(Sys, Function\,Mode\,j)$$

where *Sys* represents the system network, *sFLD* is a field formation operator which transforms the information from *Sys* to social field strength under *Function Mode j* at point (*x, y*).

Definition 5 (*Overall Social Field Strength Distribution*)

$$SF_{overall}(x, y) = \begin{bmatrix} SF_1(x, y) \\ SF_2(x, y) \\ \vdots \\ SF_m(x, y) \end{bmatrix}$$

where *m* is the number of all the possible *Function Modes* of each *agent* in the system.

Matching Social and Task Fields

The ideal system formation for a given *task j* is determined by assessing the *level of match* between task field strength and the social field strength. The ideal match is that the two strengths are equal. Otherwise, the system formation can be "under-matched" or "over-matched" for the given task. We introduce the following definition:

Definition 6 (*Level of Match for Task j*)

$$Match_j(x,y) = SF_j(x,y)/TF_j(x,y)$$

where SF_j and TF_j are the social field strength and task field strength for task j at point (x, y).

Definition 7 (*Overall Level of Match*)

$$Match_{Overall}(x,y) = \begin{bmatrix} Match_1(x,y) \\ Match_2(x,y) \\ \vdots \\ Match_m(x,y) \end{bmatrix}$$

The overall level of match for the system is represented by the vector called matching vector in the above definition. From the definition, it can be seen that the best match happens when Match $(x, y) = 1$. When it is "<1", the system is called "under-matched" (indicating the system is not capable enough to fulfill the task) and when it is ">1" the system is called "over-matched" (the system is overly capable for the task, indicating waste of agent resources).

SOM Based System Formation Algorithm

The approach proposed in this paper aims to employ an unsupervised learning to generate system formations based on the social field and task fields, so that the system does not need to go through iterations to find the parameters in structural or behavioral models for convergence. Self-organizing map is one of the most popular unsupervised machine learning algorithms, and the topology of SOM network shows great potential for achieving adaptive system formation. The learning process of our approach includes the following two steps.

Step 1: Distribution

This step is to update the position in the task space (x, y) of each agent. In this step, a variant of SOM algorithms is applied, as shown below:

1. *Initialization*

The system is initialized by deploying the agents into the working space (i.e., task space). In this paper, it is a 2-D space D^2. Each $Agent(a,b)$ has a vector (x,y,SF). (a,b) represents the topological position of the agent in the lattice of the SOM. (x,y) represents the position of the agent in the 2D working space. (x,y) is randomly picked during initialization.

SF is the overall social field strength at (x, y) and is calculated by the number of agents within the neighborhood of distance c around (a, b).

2. *Competition*

A random point (x_p, y_p) is picked up from the working space. The picked point has a vector $\left(x_p, y_p, TF\right)$. TF is the overall task field strength at $\left(x_p, y_p\right)$, which is a vector. All the agents in the system compete against each other to find the one which is closest to $\left(x_p, y_p\right)$. The winner is selected as the best matching agent (BMA), denoted by *Agent* (u, v).

3. *Cooperation*

The BMA then communicates with their neighbors and addresses the task field along with the neighbors. The "amount" of cooperation depends on the distance between their positions in the lattice network.

Definition 8 (*Neighborhood Function for Agent* (*a,b*))

$$k_{(a,b)} = exp\left(-\frac{(u-a)^2 + (v-b)^2}{2 * p_1^2}\right)$$

where (u, v) is the position of the BMA in the network, p_1 is a parameter which can be adjusted to control the strength of cooperation inside the system.

4. *Adaptation*

All agents update their vectors according to their neighborhood function and the level of task-social field matching at the picked point $\left(x_p, y_p\right)$, indicated as learning rate.

Definition 9 (*Learning Rate at* $\left(x_p, y_p\right)$)

$$\sigma\left(x_p, y_p\right) = exp\left(-\frac{\frac{\left|SF\left(x_p, y_p\right)\right|}{\left|TF\left(x_p, y_p\right)\right|}}{2 * p_2^2}\right)$$

where $\left|SF\left(x_p, y_p\right)\right|$ is the norm of $SF\left(x_p, y_p\right)$, $\left|TF\left(x_p, y_p\right)\right|$ is the norm of $TF\left(x_p, y_p\right)$, and p_2 is a parameter which can be adjusted to influence of level of match has on the distribution of the agents in the system.

Finally, the new position of *Agent (a, b)* in the working space is determined by the following equation:

$$\left(x_{(a,b)}, y_{(a,b)}\right) = \left(x_{(a,b)}, y_{(a,b)}\right) + p_3 * \kappa_{(a,b)} * \sigma\left(x_p, y_p\right) * \left(\left(x_p, y_p\right) - \left(x_{(a,b)}, y_{(a,b)}\right)\right)$$

where p_3 is a parameter which can be changed to control the rate of learning for the system.

Step 2: Differentiation

Noticing that the function mode of each agent is not changed in the distribution step, it is updated in the second step, the differentiation step. In this step, the function modes of agents are updated based on their level of match as defined in Definitions 6 and 7 according to the Function Mode Selection Diagram of Fig. 2. In general, an agent chooses the function mode corresponding to the task field that the agent has the best match.

For example, as shown in Fig. 2, each agent has three different function modes, with Mode 0 being the default standby mode. When an agent updates its function mode, the position of its matching vector determined its behavior. If the matching vector goes across the blue boundary, the agent switches to Mode 1, so as the green boundary for Mode 2 and red boundary for Mode 0. If the match vector does not intersect any boundary, the agent stays at its old mode. $s0$ is a parameter which can be adjusted to control the sensitivity for agents to switch function modes. Randomness maybe added to the function mode changing rule to increase the adaptability of the system.

The process introduced above provides an approach to apply SOM to solving system formation problems. Using this approach, in a real application, the task fields can be manipulated by changing the weight matrix ω_{m*n}, and a detailed agent behavior model should be designed by specifying the parameter p_1, p_2, p_3, the neighborhood distance c and the Function Mode Selection Diagram. The Following section provides specific case studies to show how the process described above can be implemented.

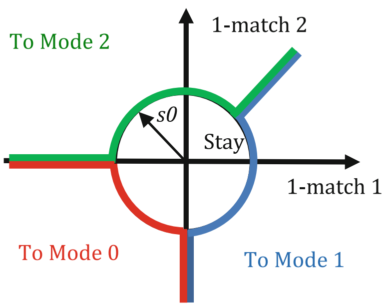


Fig. 2 Function mode selection diagram example

Case Studies

The self-organizing map algorithm is commonly used to organize high-dimensional data and visualize them in low-dimensional maps (Kit et al. 2014). The great performance of self-organizing map in vector quantization shows the possibility of building unsupervised algorithms using SOM for adaptive system formation. Several cases simulating shape formation applications are developed to validate this possibility. Matlab was used as the platform for the simulation to take advantage of its capability in calculating vectors and matrices. Figure 3 shows an example of the simulation result.

Figure 3 represents the simulation result for shape formation in a 2D space. The horizontal axis represents the x axis; the vertical axis represents the y axis. Task fields (not shown in Fig. 3) distributes in the area of $\{(x, y)|\ 0 < x < 1, 0 < y < 1\}$. Nodes with different color represent agents working under different function modes. The links between agents represent the physical connections between agents. The length of the physical connection can be changed, while the topology of the agents is predefined. Depends on different cases, the nodes and links have different specific meaning, which will be shown in the case studies below.

Case 1: Self-Organizing Tube Formation

Tubes and pipes are used to transfer fluids like air or water. While most tubes are pre-made and then installed in their applications, a self-organizing formation system will be more flexible for installation and repair. In this case study, we demonstrate the capability of our approach in using multiple non-overlapping uniform discrete

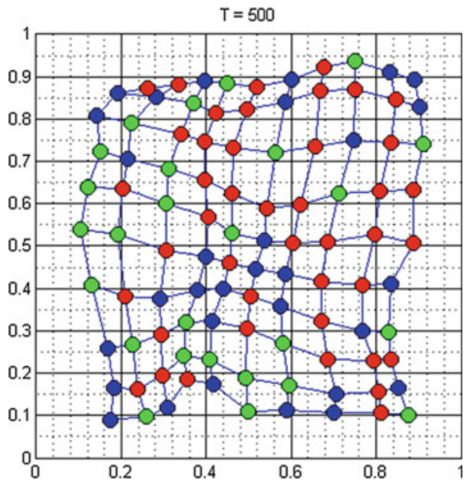
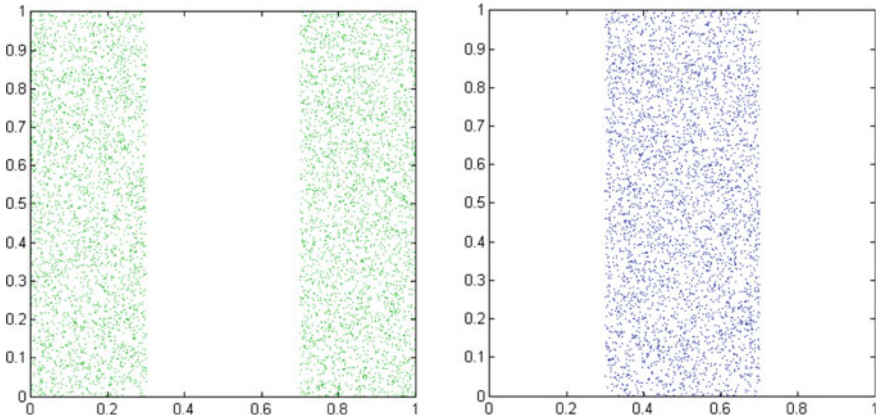


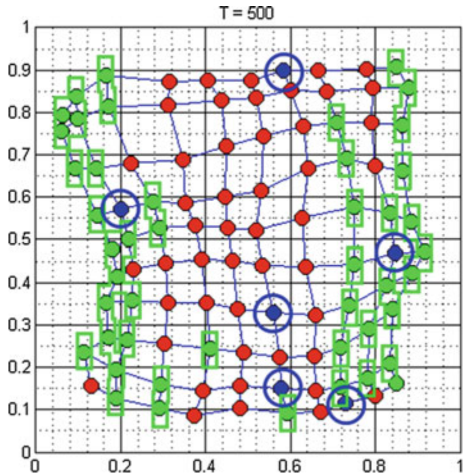
Fig. 3 Example of simulation results



(a) TF1 Distribution (b) TF2 Distribution

Fig. 4 Task field distribution

Fig. 5 Function modes



task fields. As shown in Fig. 4, the working space is a 2-D space with boundaries from 0 to 1.

Task Field 1: Tube Wall Construction Field TF1. TF1 indicates the positions of the boundary of a tube in 2-D. At the boundary, tube wall is needed to isolate the fluid from the environment. TF1 is distributed over the work space with the task field strength equals to 1 at green areas in Fig. 4a.

Task Field 2: Flow Resistance Demand Field. TF2 indicates the resistance needed to control the flow rate at each point. TF2 is distributed over the work space with the task field strength of 1 at blue areas in Fig. 4b.

The specific parameters used in the simulation are shown in Table 1.

Table 1 System parameters

Size of network	Simulation steps T	Parameter $p1$	Parameter $p2$	Parameter $p3$	Neighborhood distance c	Switch Threshold $r0$
10 × 10	500	3–0.05	0	1–0.1	0.2	Rand(1,1)/2

Table 2 Function modes

Function mode	Color	Size and shape	Function	Related task field
0 (default)	Red	Small, circle	Cause negligible resistance for the flow	None
1	Green	Long, block	Seal the flow with other agents in this mode	TF1
2	Blue	Large, circle	Cause certain resistance for the flow	TF2

Each agent in the system is programmed to have three function modes with regard to the two task fields as in Table 2. Figure 5 shows an visual example of the simulation result.

Results

The simulation results with varying strength of each task field are plotted in Fig. 6a–f. Here we make the initial task field strength fixed as 1, and adjust the weight operator w for TF1 and TF2 to change the corresponding task field strength for each simulation. Figure 6 shows simulation results for different w_1, and w_2. Figure 6a shows an initial distribution of the system: all the agents are randomly distributed in Mode 0.

Figure 6b shows the result of a simulation with both task fields are strong ($w_1 = w_2 = 15$), which means a thick tube wall and high resistance to the flow are needed. The results shows 52 agents are in Mode 2, forming the wall, and 42 agents are in Mode 1, creating large resistance inside the tube.

Figure 6c shows the result of a simulation with a medium TF1 ($w_1 = 10$), and a large TF2 ($w_2 = 15$), meaning a medium tube wall and high resistance are needed. In this simulation, 38 agents are in mode 2, forming a thinner wall. 44 agents are in Mode 1 inside the tube, which is similar to Fig. 6b.

Figure 6b, d, f shows the simulation results of decreasing TF2 ($w_2 = 15, 10, 5, 0$ respectively) with the same strong TF1 ($w_1 = 15$). As can be seen from these figures, when TF2 decreases, fewer agents inside the tube is in Mode 1 (42, 34, 17,

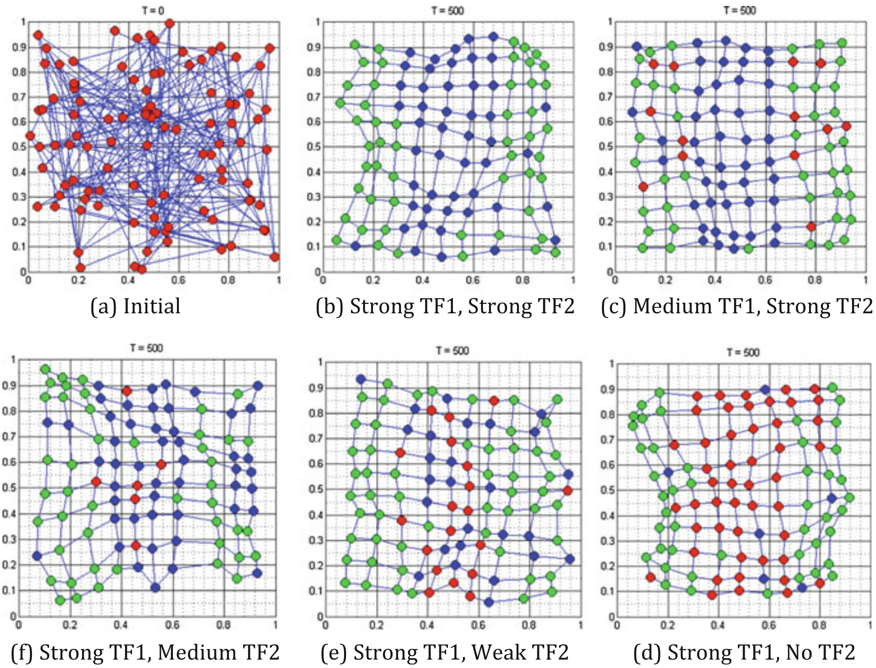


Fig. 6 Self-organizing tube simulation results

4 respectively), which means the resistance inside the tube is decreasing, correctly responding to the changing task demands.

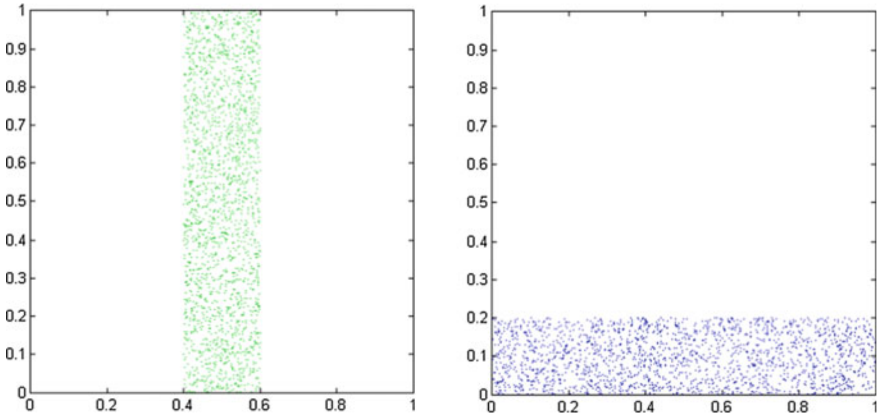
Case 2: Self-Organizing Structure Formation

The first case study validates that the proposed approach can be used in applications where task fields are "tiled" to each other. In addition, it showed that task field strength can be used to control the formation of the system. In this case study we show that the approach is also applicable for multiple intersecting task fields.

The task for the system is to form a multi-member structure, each member requiring different materials/components for achieving different functions. Two task fields are used to indicate required formation of the structure as shown in Fig. 7.

Task Field 1: Compression Field TF1 shows the expected path for the weight support of a structure. TF1 is distributed over the work space with the task field strength equals to 1 at every point.

Task Field 2: Tension Field TF2 shows the expected path for the tension force which will be loaded on the structure. TF2 is distributed over the work space with the task field strength equals to 1 at every point.



(a) Compress Field Distribution (b) Tension Field Distribution

Fig. 7 Example of task field distribution

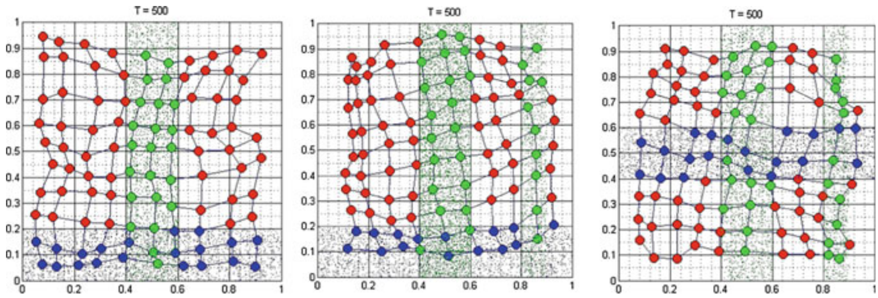
Table 3 Function modes

Function mode	Color	Connection	Function	Related task field
0 (default)	Red	None	Idle and ready to be applied	None
1	Green	Compression force	Support weight (e.g., a beam)	TF1
2	Blue	Tension force	Maintain leveling (e.g., a slab or beam)	TF2

The same system parameters are used as in Case 1, and each agent in the system is programmed to have three function modes regarding to the two task fields as in Table 3.

Results

By changing the task field distribution, we can control the formation of the structure. The simulation results are shown below in Fig. 8. From Fig. 8, we can see that the proposed approach can give accurate results to map the agent function modes to the task fields. The distribution of green and blue agents can adequately adapt to the changing arrangement of the task fields, forming needed structure to fulfill the tasks.



(a) TF1=15 (V), TF2=15 (H) (b) TF1= 15(V), TF2=15(H) (c) TF1=15(V), TF2=15(H)

Fig. 8 Simulation results (V = Vertical, H = Horizontal)

Case 3: Smart Material Formation

The two cases described above have evenly distributed task fields within the defined geometry boundary. The agents going through the self-organizing mapping process can mostly form proper shapes and structures to fulfill the designated tasks. In many other applications, however, the task distribution may not be even, and certain gradient of change of strength along a given direction may be needed for certain system formation purpose. Forming smart materials by depositing multiple kinds of materials in different densities and mixes along given directions and depth is an example of such applications. Two task fields are used in this case study.

Task Field 1: Thermal Field TF1 has its strength either fixed (1 or 8) or decreasing along the direction of x (15*(1-x) or 8*(1-x)), shown in Fig. 9.

Task Field 2: Flexibility Field TF2 always has its strength increasing along the direction of x (15x or 8*x), as shown in Fig. 9.

The same system parameters are used as in Case 1, and each agent in the system is programmed to have three function modes regarding to the two task fields as in Table 4.

Results

In each of the simulation, the two task fields TF1 and TF2 cover the entire 2D working space and are completely overlapped. Figure 9a–e show the system formation results of various combinations of the distributions of TF1 and TF2. From the figures it can be seen that the distribution of green "heat resistors" and blue "flexiblers" are mostly consistent with the demands of the two task fields TF1 and TF2, respectively. Because the working space is only 2-dimensional, it is hard to realize how such distribution can lead to smart materials. When the working space is 3-D or n-dimensional the physical realization of such distributions becomes imaginable.

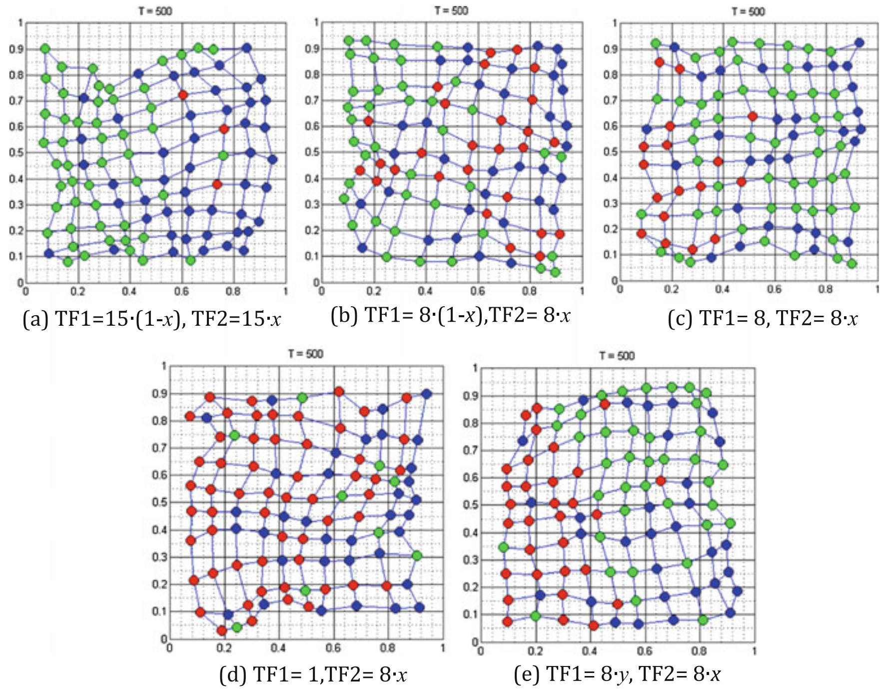


Fig. 9 Overlapping task fields simulation results

Table 4 Function modes

Function mode	Color	Connection	Function	Related task field
0 (default)	Red	None	Idle and ready to be applied	None
1	Green	Thermal effect	Resist heat	TF1
2	Blue	Flexibility	Allow bending and torsion	TF2

The comparison of the simulation results demonstrates two important features of our system:

- The distributions of agents in different functional modes are properly mapped to the distributions of multiple task fields.
- The number of agents activated in different functional modes is determined by the varying strengths of multiple task fields.

However, the results also show that even though the task field densities are varying linearly, the agent distributions are not linear. The distortion of the results is possibly because of the nonlinearity of SOM, the randomness of function mode selection, the border effect, and the limited number of agents. Our future research will address these issues.

Concluding Remarks and Future Work

Solving adaptive system formation problems is essential for developing self-organizing engineered systems. Many natural systems, including both physical and biological systems, possess the capability of dynamically evolve or develop forms, structures and functional components in response to the environmental changes and survival needs. Taking advantages of the natural field distributions, such as gravity and morphogen, natural systems can develop, maintain, and change their formations autonomously. The challenge for developing engineered systems in a similar way as nature is to devise general and yet powerful mechanisms (i.e., physical agents) and algorithms that can be applied to solve real engineering problems. Recent progress in micro robotics as well as drones has offered mechanism opportunities, whereas challenges remain with seeking proper self-formation algorithms.

In this paper, a SOM based approach is proposed to design CSO systems for solving adaptive system formation problems. As an unsupervised learning algorithm, SOM provides certain level of intelligence and makes systems more flexible for complex tasks. The core idea behind the proposed approach is to transform SOM from an artificial neuron network into an agent network by using field-based behavior regulation. The task fields are used to capture the task requirements, and the social field represents the system formation. The CSO system "redesigns" itself automatically by matching the social field with the task fields. The case studies demonstrated the ability of the SOM-based approach.

Compared with existing research, our approach shows uniqueness in the following aspects:

- The information of the tasks is embedded in task fields, external to the multi-agent system. The structural model and behavior model of our approach is independent from tasks, which means the same system with the same configuration can be used to perform different tasks in different environments. In case of shape formation, the change of the task fields will guide the agents self-organize into different shapes.
- Our approach has the capability of dealing with multiple task fields. The case study results show that agents in the system are able to differentiate into different "species" according to their tasks. Differentiation is the fundamental capability for biological systems to evolve sophisticated functional organs. Our approach aims toward that direction.
- In our approach, agents do not need to synchronize their coordinates for gaining global spatial information in order to choose their behaviors. Instead, the agents

are engaged in a parallel mapping process, which increases the speed of self-organization, but also makes it difficult for the system to perform formation tasks that requires precision.
- From a SOM algorithm perspective, in comparison with existing density tracking network algorithms, our approach is a bottom-up one. The implementation of the algorithm is distributed to individual agents rather than a single computer. By using the field regulation, the system captures multiple 3-dimensional dataset in 2-dimensional space at the same time, making the multiple mappings and calculations more efficient.

The long term goal of our research is to devise algorithms that can allow systems to adapt to changing tasks and environments by changing and evolving system formation of shapes, structures, and functional components autonomously without human intervention. To pursue this goal, we will conduct more case studies with more close-to-real example problems. In doing so, we will further enhance our task field modeling scheme and social field system presentation. Physical implementation is also a future direction.

Acknowledgements This paper is based on the work supported in part by the National Science Foundation under Grants No. CMMI-0943997 and No. CMMI-1201107. 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

Bai L, Eyiyurekli M, Breen DE (2008) Self-organizing primitives for automated shape composition. In: IEEE international conference on shape modeling and applications, Stony Brook, NY, pp 147–154

Chen C, Jin Y (2011) A behavior based approach to cellular self-organizing systems design. In: IDETC/CIE 2011, Aug 28–31, 2011, Washington, DC

Claussen JC, Villmann T (2005) Magnification control in winner relaxing neural gas. Neurocomputing 63:125–137

De Rosa M, Goldstein S, Lee P (2006) Scalable shape sculpting via hole motion: motion planning in lattice-constrained modular robots. In: 2006 IEEE international conference on robotics and automation, pp 1462–1468

Derakhshandeh Z, Gmyr R, Strothmann T (2015) Leader election and shape formation with self-organizing programmable matter. In: 21st international conference on DNA and molecular programming, pp 117–132

Doursat R (2012) The self-made puzzle: integrating self-assembly and pattern formation under nonrandom genetic regulation. In: Unifying themes in complex systems VII. Springer, Berlin, Heidelberg, pp 54–67

Ferguson SM, Lewis K (2006) Effective development of reconfigurable systems using linear state-feedback control. AIAA J 44(4):868–878

Fukuda T, Ueyama T (1992) Self-organizing cellular robotic system: comparison between social system and robotic system. In: 1992 IEEE international biomedical engineering days, pp 39–45

Galda H, Murao H, Tamaki H (2004) Dermoscopic image segmentation by a self-organizing map and fuzzy genetic clustering. IEICE Trans Inf Syst E87-D(9):2195–2203

Humann J, Khani N, Jin Y (2014) Evolutionary computational synthesis of self-organizing systems. AIEDAM 28(3):259–275

Kawauchi Y, Inaba M, Fukuda T (1992) Self-organizing intelligence for cellular robotic system 'CEBOT' with genetic knowledge production algorithm. In: 1992 IEEE international conference on robotics and automation, pp 813–818

Khani N, Jin Y (2015) Dynamic structuring in cellular self-organizing systems. In: Design computing and cognition'14. Springer, Berlin, pp 3–20

Kiang MY, Kulkarni UR, Kar YT (1995) Self-organizing map network as an interactive clustering tool—an application to group technology. In: 2nd workshop on information technology and systems (WITS'92), pp 351–374

Kit D, Kong Y, Fu Y (2014) LASOM: location aware self-organizing map for discovering similar and unique visual features of geographical locations. In: 2014 international joint conference on neural networks, pp 263–270

Kohonen T (1990) The self-organizing map. Proc IEEE 78(9):1464–1480

Nagpal R (2006) Self-organizing shape and pattern: from cells to robots. IEEE Intell Syst 21 (2):50–53

Tolley MT, Lipson H (2010) Fluidic manipulation for scalable stochastic 3D assembly of modular robots. In: 2010 IEEE international conference on robotics and automation, pp 2473–2478

Villmann T, Claussen JC (2006) Magnification control in self-organizing maps and neural gas. Neural Comput 18(2):446–469

Werfel J (2010) Biologically realistic primitives for engineered morphogenesis. In: Swarm intelligence, volume 6234 of the series Lecture Notes in Computer Science, pp 131–142

Zhang Y, Tang Y, Fang B (2014) Real-time object tracking in video pictures based on self-organizing map and image segmentation. In: 2014 7th IEEE joint international information technology and AI conference, pp 559–563

Zheng Y, Greenleaf JF (1996) Effect of concave and convex weight adjustments on self-organizing maps. IEEE Neural Netw 7(1):87–96

Utilizing Markov Chains to Understand Operation Sequencing in Design Tasks

Christopher McComb, Jonathan Cagan and Kenneth Kotovsky

Abstract Design often involves searching for a final design solution by iteratively modifying and adjusting a current design. Through this process designers are able to improve the quality of the current design and also learn what patterns of operations are most likely to lead to the quickest future improvements. Prior work in psychology has shown that humans can be adept at learning how to apply short sequences of operations for maximum effect while solving a problem. This work explores the sequencing of operations specifically within the domain of engineering design by examining the results of a human study in which participants designed trusses. A statistical analysis of the data from that study uses Markov Chains to show with high confidence that meaningful operation sequences exist. This work also uses an agent-based modeling framework in conjunction with Markov Chain concepts to simulate the performance of teams with and without the ability to learn sequences. These computational studies offer confirmation for the conclusion that sequence-learning abilities are helpful during design.

Introduction

Design often involves searching for a final design solution by iteratively modifying and adjusting a current design. Through this process designers search the design space, attempting to iteratively improve on their current solution. However, through the process of searching, designers also learn how to efficiently navigate the design space through the operations that they use to modify their solutions. This work studies the ability of designers to learn beneficial sequences of operations, and examines the performance implications of such behavior.

The ability to learn sequences of operations is essential to human performance across a variety of both everyday and specialized tasks (Clegg et al. 1998). A related behavior is the ability of humans to collect many pieces of related

C. McComb · J. Cagan (✉) · K. Kotovsky
Carnegie Mellon University, Pittsburgh, USA
e-mail: cagan@cmu.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_22

information within a single unit of memory. This process is commonly referred to as *chunking*, and has been observed across a variety of domains (Chase and Simon 1973; Egan and Schwartz 1979; Reitman 1976). Observations of chunking behavior in humans have even led to the enhancement of computational design algorithms (Moss et al. 2004). While chunking is an important aspect of design, this work focuses on an equally important aspect: sequencing of operations. Sequencing may be thought of as temporal chunking, but the two behaviors are functionally distinct.

In some domains, the presence of sequential behavior has been shown to be indicative of expertise (Pretz et al. 2003). However, in other studies of individual problem-solving that used the Tower of Hanoi puzzle (Kotovsky et al. 1985; Kotovsky and Simon 1990) or the Thurstone Letters Series Completion task (Simon and Kotovsky 1963) it was shown that participants were able to acquire and begin employing move sequences within a short period of time. In studies using the Tower of Hanoi puzzle, participants spent most of their time learning to sequence moves appropriately, and a relatively short amount of time in finally solving the puzzle (Kotovsky et al. 1985). When comparing easy and difficult isomorphs of that puzzle, it was discovered that participants solving hard isomorphs spent longer in the learning phase, but that both conditions took approximately the same amount of time in the final solving phase (Kotovsky et al. 1985). Studies using the Thurstone task identified that participants' problem-solving behavior consisted broadly of two steps. In the first step participants recognized some order in the letter series and codified it as a rule (i.e., generating a pattern), and in the second step they used that rule to extrapolate the letter series (i.e., generating a sequence) (Simon and Kotovsky 1963). This two-step process was validated with computer simulations (Simon and Kotovsky 1963), and has since been revisited and confirmed (Kotovsky and Simon 1973). Other work explicitly studied the order of operations used in a geometric analogy task (Novick and Tversky 1987). Although the task itself placed no constraints on the order in which operations could be performed, it was found that there was a strong preference amongst participants for a specific order (Novick and Tversky 1987). Further, participants made to use a non-preferred order performed with lower speed and accuracy, indicating that the preferred order was an important construct (Novick and Tversky 1987).

There is evidence that humans learn sequences implicitly, without the need for direct attention (Nissen and Bullemer 1987; Reed and Johnson 1994) but there is also evidence that explicitly learning sequences can boost effectiveness (Perruchet and Amorim 1992; Willingham et al. 1989). This evidential duality gives credence to the theory that sequence learning can occur through a variety of cognitive pathways (Clegg et al. 1998; Curran and Keele 1993).

Research on sequencing in design has examined both the ordering of design stages and the ordering of specific tasks. The sequencing of discrete design operations, however, has received little attention in engineering design. The following literature review is summarized in Table 1. These three sequencing levels (design stages, tasks, or operations) can be conceptualized along a continuum that describes the degree of abstraction of the sequenced object, from design stages (the most abstract and generalized of the three) to design operations (the least abstract and

Table 1 Summary of sequencing research in engineering design

Sequenced object	Timescale/abstraction	Engineering design literature summary
Stages	Long timescale; high abstraction	Teams tend to talk about a single design stage several times before transitioning to another (Stempfle and Badke-schaub 2002) Experts' patterns of design stage transitions are more linear and smooth than those of novices (Atman et al. 2007) Few designers follow a perfectly linear pattern (Goldschmidt and Rodgers 2013) Linear patterns can produce more effective solutions (Radcliffe and Lee 1989)
Tasks	Intermediate timescale; moderate abstraction	Appropriate task sequencing can: increase concurrency (Todd 1997) minimize the time and cost (Rogers 1996) increase information availability (Sen et al. 2010) Tasks may span design stages (Waldron and Waldron 1988) Task order can be optimized algorithmically (Meier et al. 2007)
Operations	Short timescale; low abstraction	To be addressed in this work

most detail-specific). The sequencing levels can also be differentiated in terms of the timescale at which the sequenced object is enacted, with design stages at the longest timescales and design operations at the shortest.

The sequencing of design stages is commonly examined through design studies with either teams or individuals. In a study that coded intra-team design communication as either content-focused or process-focused, it was noted that teams had a high probability of remaining on one focus for several communicative acts before switching to the other focus (Stempfle and Badke-schaub 2002). The same study also coded communication in terms of steps in the design process, and noted a similar pattern of repeated communicative acts within a step (Stempfle and Badke-schaub 2002). Another study tasked participants with the design of a playground, and activity was coded with respect to design stages (Atman et al. 2007). Expert sequences flowed smoothly from one stage to the next, while novice sequences were more choppy and unstable (Atman et al. 2007). In a comparison of undergraduate and PhD students, it was noted that there was significant variability in the sequence in which design stages were visited (Goldschmidt and Rodgers 2013). Interestingly, none of the participants followed a linear design process (Goldschmidt and Rodgers 2013). A similar study showed that participants who transitioned linearly through the design process produced more effective solutions (Radcliffe and Lee 1989).

The sequencing of individual design tasks however takes place at shorter timescales than the sequencing of design stages, with numerous design tasks usually performed within a single stage. Intelligent ordering of tasks can increase the concurrency with which tasks can be completed (Todd 1997), minimize the time and cost involved in developing a product (Rogers 1996), and increase the availability of information for important decisions (Sen et al. 2010). Waldron and Waldron observed the sequence of tasks involved in the design of a complex walking vehicle (Waldron and Waldron 1988). They noted that there was not a distinct separation between conceptual and detailed design and that tasks may span across design stages (Waldron and Waldron 1988). Theoretical work on task sequencing has shown that optimal strategies for ordering tasks may be tied to problem complexity (Sen et al. 2010). Other work has utilized genetic algorithms to optimize task sequences for a variety of different objectives (Meier et al. 2007).

The current work specifically stems from a study in which small teams of engineering students were instructed to design a truss (McComb et al. 2015a). Previous work hypothesized that the order in which operations were performed in that study may have had a large impact on the quality of solutions (McComb et al. 2015b). By focusing on the ordering of operations, the analysis in the current work is conducted at a more fine-grained resolution and at shorter timescales than other work on design stage or task sequencing. This degree of resolution is more akin in scope to studies conducted in the psychology literature. It is at this level that engineers and designers directly interact with potential solutions, so the selection and application of the best actions is particularly crucial for the creation of high-performing solutions. Two questions are specifically addressed in this work:

1. Do designers employ recognizable sequences of operations when solving a design problem?
2. Is the employment of sequences of operations beneficial to designers?

This work proposes that the Markov Chain model can be used as a tool for understanding and simulating the order in which humans perform operations in a design context, thus helping to address the two research questions. It should be noted that Markov Chain models do not directly extract fixed-length sequences of operations. Instead, such models implicitly represents fixed-length sequences using probabilistic chains.

After a brief background section, the main body of this paper is presented as two investigations, each of which is aligned with one of the research questions posed previously. The first explores whether or not designers employ recognizable operational sequences. To answer that question, a statistical analysis is conducted on the human data from a prior cognitive study. This analysis indicates with high confidence that participants of the study utilized sequences of operations when creating solutions. The second investigation analyzes whether or not sequences of operations are beneficial to the designer. This is accomplished by performing computational simulations of design team behavior within the Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework, an agent-based platform that approximates the process and performance of engineering design teams. These simulations demonstrate that the ability to learn sequences during design was extremely beneficial to teams in the study.

Background

This work draws from two elements of prior work. The first is the Markov Chain model, a statistical model that provides a means for modeling sequences of operations. The second is a corpus of data from a truss design study by McComb et al. (2015a). The current work applies Markov Chain models to study and analyze the operations used in that study.

Markov Chains

A Markov process is a stochastic process in which a system transitions between a finite number of discrete states. Markov processes are commonly modeled using Markov Chain models (Stroock 2005). In a Markov Chain model, the probability of transitioning to a future state depends only on the current state of the system, and not on previous states (Stroock 2005). The probability of transition to a future state from a current state is given by the transition matrix, $\boldsymbol{T}$, where the value of T_{ij} is the probability of transitioning from state i to state j. Markov Chains were first introduced more than a century ago (Seneta 2006), and have since been used in a number of applications including computer performance evaluation (Scherr 1962), web search (Page et al. 1999), and chemical processes (Tamir 1998).

Figure 1 depicts a Markov Chain with three states (S_1, S_2, and S_3). The entries of the transition matrix are shown next to arrows that indicate the relevant transition. Self-transitions are allowed in the model, meaning that the system remains in the same state across multiple time steps.

In this work, Markov Chains are used to describe the order in which modifying operations are applied to designs. Operations constitute the states of the Markov Chain model, and the transition probabilities in $\boldsymbol{T}$ describe the probability of transitioning between those operations.

It should be noted that this work considers only first-order Markov Chain models, meaning that the next state depends only on the current state. Higher-order Markov Chain models assume that the selection of the next state is dependent on

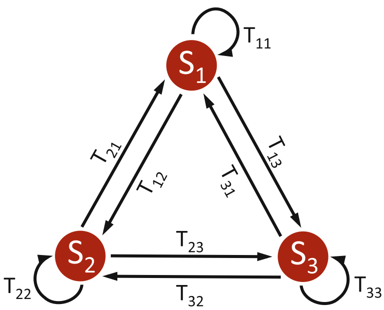


Fig. 1 Example Markov Chain with 3 states

the current state as well as any number of past states, thus encoding some degree of "memory" in the model (Raftery 1985). Within the domain of design, the inherent memory of higher-order models could provide a useful analog of expertise and memory of human designers. However, increasing the order of a Markov Chain model quickly increases the overall complexity of the model, which in turn increases the likelihood of over-fitting and decreases the accuracy with which the model parameters can be estimated. Therefore, in the interest of model parsimony and accurate parameter estimation, this work uses only first-order Markov Chain models. It will be shown that first-order Markov Chain models are sufficient to achieve the objectives of this work.

Cognitive Study Data

The operation sequences used in this work were extracted from a design study in which 16 small teams of 3 engineering students were tasked with the design of a truss structure (McComb et al. 2015a). Design was conducted over the course of six, 4-min design sessions.

During the study, the design problem was changed twice without warning through the introduction of new problem statements. The initial problem statement simply asked teams to design a truss with the two loading points and three supports shown in Fig. 2. The first change presented participants with the same layout, but also instructed them to account for the fact that any one of the supports could be removed at any time. The second change modified the problem so that teams had to build their truss around an obstacle. Teams were given a target mass and

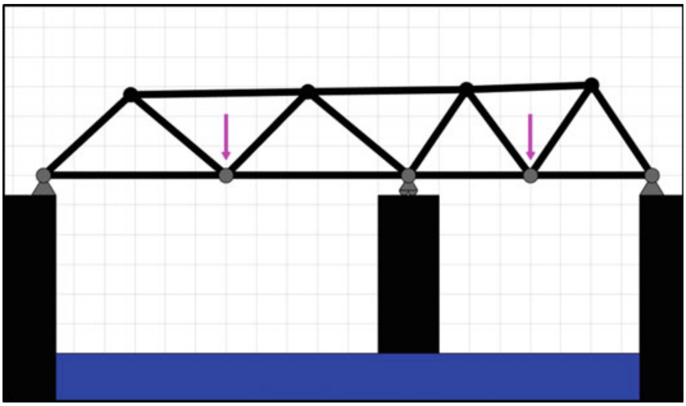


Fig. 2 Example solution to the truss design task. The locations of the supported and loaded joints (the *lowest row of grey circles*) could not be modified by participants

factor-of-safety for the initial problem statements and for each of the subsequent modified problem statements.

To facilitate design, every participant was given access to a computer-implemented truss design program that was created for the purpose of the study. This program allowed the participants to build, assess, and share truss designs within their teams. In addition to facilitating design, the truss design program was used to continuously record the operations that participants chose and applied in order to modify their designs, thus allowing a full account of design operations to be accumulated. The operations available to participants were:

1. Adding a joint.
2. Removing a joint.
3. Adding a member.
4. Removing a member.
5. Changing the size of a single member.
6. Changing the size of all members.
7. Moving a joint.

This information was post-processed to extract a string of move operators (denoted by the integers 1 through 7) for each of the 48 participants in the study. A typical solution sequence consists of 400–500 operations. A short example sequence is provided in Fig. 3.

When training multinomial models and Markov Chains in the current work, each operation is treated as a state observation. In addition, the sequence of moves for a single participant is treated as a single piece of data, and is not subdivided during cross-validation.

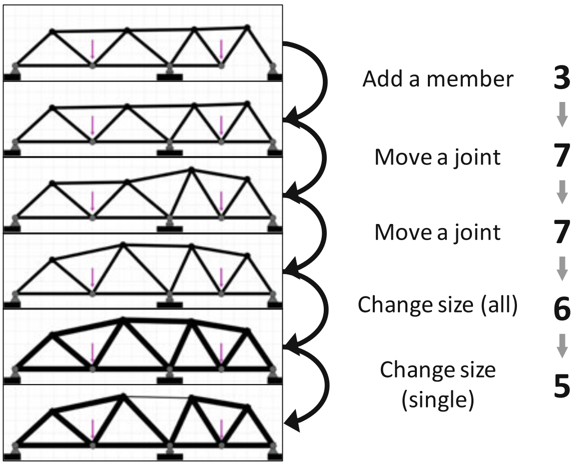


Fig. 3 Example operation sequence, with numbers corresponding to the list of truss operations introduced previously

Do Designers Use Operation Sequences?

This paper first analyzes human protocol data from the truss design study with the objective of determining whether or not any common operation sequences are apparent. This is accomplished by training and comparing sequential (Markov Chain) and no-sequential (multinomial) statistical models on the operation protocol data extracted from the truss design study.

Methodology

Markov Chain models are trained on the human data in order to provide a statistical representation of the sequence in which operations occurred in the cognitive study. The following discussion is based on material in Stroock (2005), but is presented in terms of design operations (rather than Markov process states).

A Markov Chain is defined by the values of its transition matrix, $\boldsymbol{T}$. Element T_{ij} in the matrix defines the probability that the next operation will be operation j, given that the previous operation was operation i. Therefore, the elements of the transition matrix for a Markov Chain can be estimated using

$$T_{ij} = \frac{N_{ij}}{N_i} \tag{1}$$

where N_{ij} is the number of times that operation j is observed to follow operation i, and N_i is the number of times that operation i is observed.

The log-likelihood is a quantity that indicates the probability that a model could have produced a given set of data, and can thus be used to compare different models. The log-likelihood for a Markov Chain model ($\mathcal{L}_{MC}$) is

$$\mathcal{L}_{MC} = \sum_{i=1}^{M} \sum_{j=1}^{M} N_{ij} \cdot \ln\left(T_{ij}\right) \tag{2}$$

where N_{ij} and T_{ij} are defined as above and M is the number of different operations.

Multinomial models are also trained on the same data in order to provide a non-sequential statistical representation. In this application, the multinomial model can be thought of a weighted die with M sides (one corresponding to every truss operation). Training the multinomial model on the truss design data encodes the frequency with which each operation occurred. This information is stored in a vector $\boldsymbol{p}$ of length M. The elements of $\boldsymbol{p}$ can be estimated by computing

$$p_i = \frac{N_i}{N_{TOTAL}} \tag{3}$$

where N_i is the number of times that operation i is observed in a given dataset, and N_{TOTAL} is the total number of operations performed in the dataset. The log-likelihood for a multinomial model ($\mathcal{L}_{mult}$) can be computed similarly to the multinomial model as

$$\mathcal{L}_{mult} = \sum_{i=1}^{M} N_i \cdot \ln(p_i) \tag{4}$$

where N_i, p_i, and M are defined as above.

These models can be used to provide non-sequential (multinomial) and sequential (Markov Chain) representations of the human operation data from the cognitive study. By comparing the log-likelihood of the two models, it becomes possible to determine which is a better representation of the data. Further inspection of the trained models can divulge details of the way in which human study participants operated on solution concepts.

Results

Statistical models are trained on the data using k-fold cross-validation (Stone 1974). Cross-validation first partitions the available data into k subsets of equal size. A model is then trained using $k-1$ subsets, and tested using the remaining subset. This is repeated until every subset has been used for testing (k times). Specifically, this work uses k-fold cross validation with $k = 8$.

The log-likelihood of each model (computed on the testing subsets) is shown with a boxplot in Fig. 4. The horizontal lines of the boxes indicate the 25th, 50th, and 75th percentiles, and the tails above and below the boxes indicate maxima and minima.

The difference between the two models is highly significant (ANOVA: $\rho < 10^{-6}$, $F = 70.4$). Because the sequential Markov Chain model has a significantly higher log-likelihood than the non-sequential multinomial model, the human data displays significant amounts of sequencing. This means that participants in the study identified and utilized chains of move operators. However, this analysis does not indicate whether or not this behavior was beneficial. An evaluation of the relationship between sequencing and performance is addressed in the next section.

The probabilities of the average multinomial model are shown in Fig. 5, and the transition matrix of the average Markov Chain model is shown in Fig. 6. The colors of the squares indicate the magnitude of the probability, which is also noted numerically within each square.

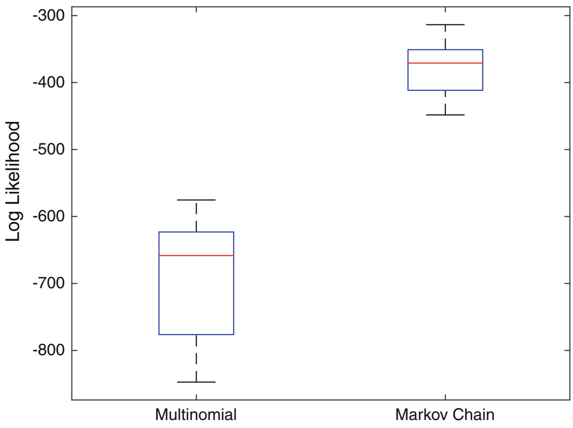


Fig. 4 Log-likelihood of multinomial and Markov Chain models

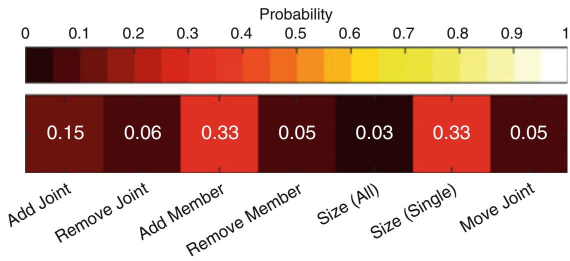


Fig. 5 Parameters of multinomial model based on human data

Examining the differences between Figs. 5 and 6 provides some insight as to why the Markov Chain model is more veridical than the multinomial model. The most striking aspect of the transition probability matrix is that it is largely diagonal. This indicates that most move operators were likely to be applied several times before another move operator was selected. This aspect of the participants' behavior simply cannot be captured in the multinomial model. For instance, the multinomial model indicates a 33% chance that the next operation will be to add a member, regardless of the previous operation. However, the column of the Markov Chain transition matrix that corresponds to adding a member shows that that operation is most likely chosen after adding a joint, removing a joint, or adding a member, and rarely selected if the last move involves changing the size of truss members. Thus, the selection of the subsequent operations is highly conditional on the last operation performed. Similarly, the multinomial model shows a 33% chance of changing the size of a single member. However, the Markov Chain transition matrix more incisively encodes the operational data by showing that that operation is rarely selected after adding a joint, removing a joint, adding a member, or removing a member.

In addition, a graphical representation of the average Markov Chain model is provided in Fig. 7. Arrows are used to indicate transitions between states and line thicknesses represent the relative likelihood of those transitions, with thicker lines

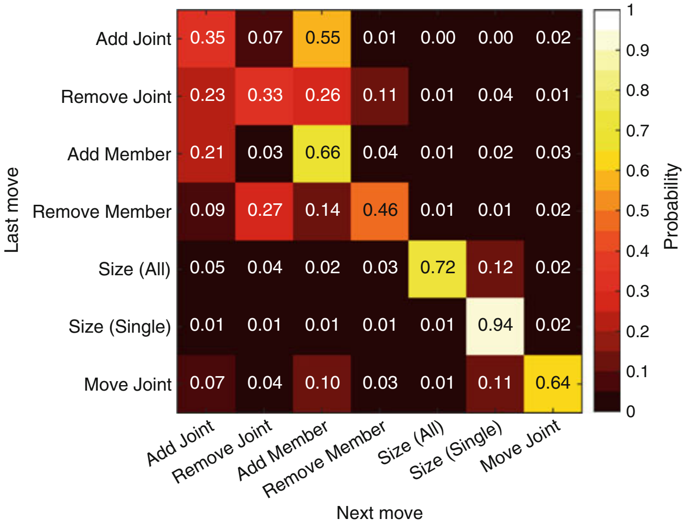
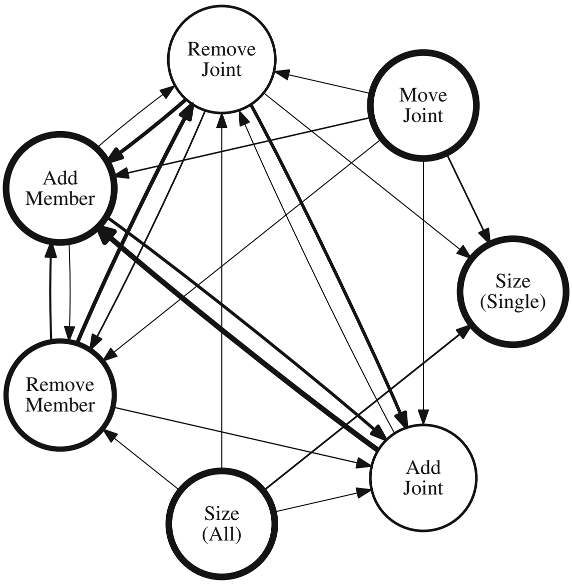


Fig. 6 Transition matrix for the Markov Chain model based on human data

Fig. 7 Graphical representation of the Markov Chain model showing the most likely transitions. Transition probability is indicated by the line thickness of the corresponding arrow (in the case of self-transitions, the thickness of the border of the corresponding operation)



indicating the more likely transitions. Only the strongest transitions were included in the representation (those with transition probabilities above the median, approximately 0.03). The probabilities pertaining to self-transitions are indicated by the thickness of the border of the corresponding operation.

Figure 7 provides further insight as to the pattern of operations in the study. Actions involving the topology of the truss (adding and removing joints and members) are connected by the thickest arrows, indicating that there is a strong probability of transition between these operations. This indicates that these topology operations were usually employed together during the design study.

In contrast, operations that only change the shape of the truss (changing the size of members, or moving joints) are connected by arrows that are much thinner, indicating that there is far less interaction between these operations. These operations are also much more likely to be applied multiple times in a row, as indicated by the probability of a self-transition. Finally, there is a very low probability of leaving the state corresponding to changing the size of a single member ("size (single)"). This indicates that participants were very likely to repeatedly apply this operation to fine-tune the size of the members in their designs.

Are Operation Sequences Beneficial to Designers?

The previous section established that there are recognizable operation sequences in the data from the truss design study. This section investigates whether or not the use of operation sequences was beneficial to the participants of the study. Because sequence learning can take place implicitly (Nissen and Bullemer 1987; Reed and Johnson 1994) it would be nearly impossible to control as an experimental variable in a study with human participants (i.e., implicit processes could take over even if a participant were successfully prevented from explicitly learning any operation sequences). For that reason, this work utilizes the the CISAT modeling framework (described in greater detail below) to test the effects of sequence learning (McComb et al. 2015b). Individuals and teams simulated in CISAT have clearly defined abilities, making it possible to perfectly control the way that these simulated individuals learn operational sequences. Therefore, a comparison between sequential and non-sequential learning patterns becomes feasible, and offers insight as to the benefit of sequence learning for real human designers.

Methodology

The Cognitively-Inspired Simulated Annealing Teams (CISAT) modeling framework is an agent-based computational platform that effectively simulates both the process and performance of engineering design teams (McComb et al. 2015b). At its core, the CISAT framework makes use of simulated annealing constructs. Eight additional cognitive characteristics are then layered on top of the core structure with the intention of supporting a more full description of how individuals proceed during design, and how they interact with one another as part of a team (McComb et al. 2015b).

The operational learning characteristics of the CISAT framework is of particular interest in this work. The intention behind this characteristic is to provide CISAT agents with the ability to learn which move operators should be applied in the future. The characteristic was originally implemented as follows. During every iteration, an agent selects which move operator to apply next by taking a random draw from a multinomial distribution defined by a vector of probabilities, $\boldsymbol{p}$. The chosen move operator, i, is then used to modify the current solution. If operator i improves the quality of the solution, then the probability that that operator will be chosen in the future is increased using the equation

$$p_i \leftarrow p_i \cdot (1 + k_{OL}), \tag{5}$$

where k_{OL} is a constant used to control how quickly learning occurs. If the move operator worsens the quality of the solution, the probability of selecting it in the future is decreased using

$$p_i \leftarrow p_i \cdot (1 - k_{OL}). \tag{6}$$

The probability vector is re-normalized following every update.

This work updates the operational learning characteristic in CISAT to accommodate sequence learning via Markov Chain concepts. The move operator selection process is modified so that agents select move operators to apply using the probabilities in a Markov Chain transition probability matrix, $\boldsymbol{T}$, and then iteratively tune the probabilities in $\boldsymbol{T}$ to bias selection towards the best move operators. This process is similar to the bipartite pattern/sequence generation process identified by Kotovsky et al. for human participants solving the Thurstone Letter Series Completion task (Simon and Kotovsky 1963). Here, the tuning of probabilities in $\boldsymbol{T}$ constitutes the pattern generation step, and the selection of a subsequent operation based on $\boldsymbol{T}$ is analogous to the sequence generation step.

To select which move operator to apply, a random draw is taken from the multinomial distribution defined by row i of the matrix $\boldsymbol{T}$, where operator i is the last move operator applied. The chosen operator, operator j, is then applied to the current solution. The values in $\boldsymbol{T}$ are updated depending on whether the quality of the current solution improves

$$T_{ij} \leftarrow T_{ij} \cdot (1 + k_{OL}), \tag{7}$$

or worsens

$$T_{ij} \leftarrow T_{ij} \cdot (1 - k_{OL}). \tag{8}$$

Through the Markov Chain transition matrix, the selection of the next move operator is made conditional upon the last move operator that was applied. This modification enables CISAT agents to learn and apply beneficial sequences of operations. Note that the Markov Chain model does not directly extract fixed-length sequences of operations from the experience of the agent. Rather, these sequences are implicitly coded within the transition matrix of the Markov Chain model by updating probabilities according to move operator performance (i.e., whether applying a move operator improved or worsened solution quality). This makes the enactment of previously beneficial sequences more likely, thus accounting for sequence learning within the CISAT modeling framework. However, the probabilistic nature of operator selection still allows for the exploration and discovery of new sequences. This is the only modification made to the framework, and all other constants and settings are as given in McComb et al. (2015b).

Results

The simulated teams that are capable of learning operation sequences are programmed to use Markov Chain learning models, whereas the teams that are not capable of learning sequences will use multinomial learning models. Comparing the solution quality obtained by these two groups will indicate whether or not sequence-learning can be beneficial within a design context.

In the interest of simplicity, performance is only simulated for the initial problem statement as shown in Fig. 2. For the multinomial learning condition, the simulation parameters are the same as those used in McComb et al. (2015b). For the Markov Chain learning condition, the only modification made is to the operational learning characteristic to enable sequence learning via a Markov Chain model. A total of 100 teams are simulated for each condition. The results of the simulations are then post-processed to track each team's best solution over time. The mean Normalized Strength-to-Weight Ratio of each team's current best solution (abbreviated SWR, and indicative of solution quality) is then computed, and is shown in Fig. 8.

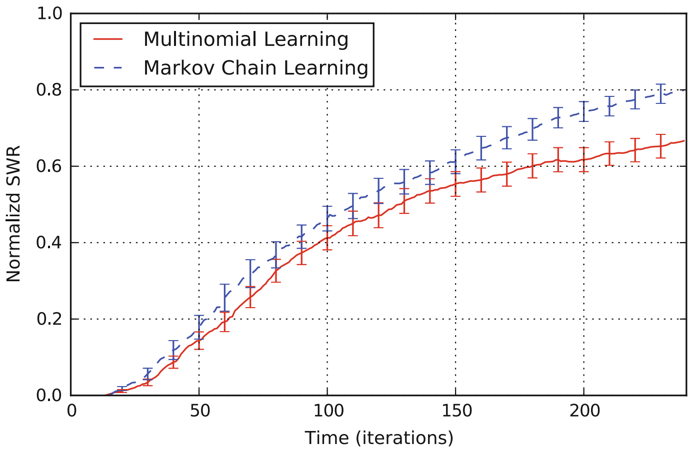


Fig. 8 Comparison of normalized SWR for multinomial and Markov Chain learning approaches (error bars show ±1 SE)

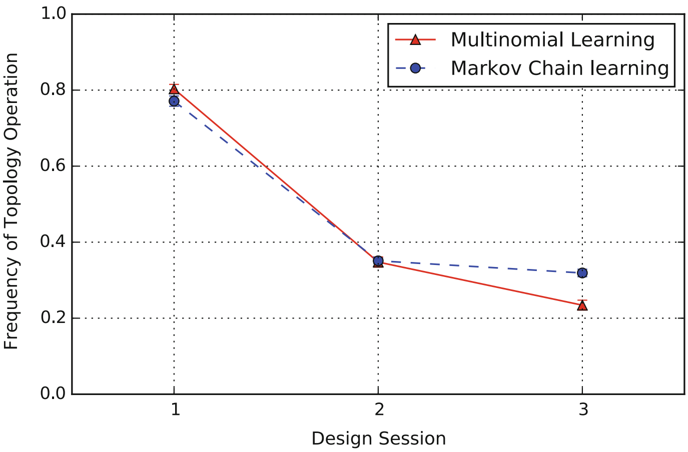


Fig. 9 Comparison of frequency of topology operations for multinomial and Markov Chain learning approaches (error bars show ±1 SE)

The difference in final design quality between the two conditions is highly significant (ANOVA: $\rho < 10^{-3}$, $F = 11.2$) and the Markov Chain condition achieves a higher final design quality. The framework used to conduct these simulations has been shown to accurately model engineering design teams solving this design problem (McComb et al. 2015b). Therefore, these simulations indicate that the ability to learn operation sequences while solving a design problem is beneficial to human designers.

Figure 9 compares the frequency with which topology operations are applied during the simulations. Topology operations are those that modify how the truss designs are connected: adding members, removing members, adding joints, and removing joints. All other operations (moving joints and changing the size of members) are shape operations. This metric can be used to track the progressive transition from layout design to detailed design. In the original cognitive study, the initial problem statement was solved over the course of three design sessions, of 4 min each, in which participants conducted design activities with short breaks in between. To calculate the frequency of topology operations metric, the data is split according to the three design sessions in the original study.

Comparing the two conditions through Fig. 9 indicates that the overall pattern of topology versus shape operations did not change substantially with the addition of sequence learning. Simulated teams in both conditions follow a similar global pattern in transitioning from layout design to detailed design. However, the intelligent sequencing of those operations in the Markov Chain condition leads to substantially improved final design quality.

The CISAT simulations indicate that operation sequencing is a beneficial aspect of human cognition during design. However, the results also indicate that applying the right set of operations to a design problem has little benefit unless those operations are applied in the correct order. For this reason, designers should emphasize the importance of learning beneficial operation sequences during iterative design problems.

Conclusions

This work examined operation sequencing in engineering design and demonstrated that it was both present and beneficial to designers in a human study. Two research questions were specifically addressed:

1. Do designers employ recognizable sequences of operations when solving a design problem?
2. Is the employment of sequences of operations beneficial to designers?

To address the first question, both sequential (Markov Chain) and non-sequential (multinomial) models were trained on the operation data from a human study. A comparison of the two models showed that the Markov Chain model was more likely. This indicated that participants in the truss design study employed sequences of move operators while designing. Further analysis of the Markov Chain model provided insights as to the pattern of topology versus shape operations used by participants.

To address the second question, computational simulations were performed to assess whether or not the ability to learn and employ operation sequences was beneficial to study participants. Simulated teams without the ability to learn sequences utilized a multinomial reinforcement learning strategy, while teams capable of sequence-learning were simulated with a Markov Chain reinforcement learning strategy. It was shown that the ability to learn operation sequences significantly increased the quality of solutions.

Together, these two findings indicate that designers can benefit from identifying and employing beneficial sequences of operations while solving iterative design problems. This would allow them to improve solutions with greater efficiency, thus achieving the delivery of final design solutions of higher quality. This work also demonstrated the utility of Markov Chains for modeling and extracting meaning from operation-based protocol data. This approach can be applied to any protocol data that has a finite and well-defined set of operations. For that reason, designer studies that entail the application of design grammars may be particularly well-suited to analysis via Markov Chains. This approach will be further utilized and extended in future work.

Acknowledgements This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE125252 and the United States Air Force Office of Scientific Research through Grant FA9550-16-1-0049. 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 sponsors.

References

Atman CJ, Adams RS, Cardella ME, Turns J, Mosborg S, Saleem J (2007) Engineering design processes: a comparison of students and expert practitioners. J Eng Educ 96(4):359–379. doi:10.1002/j.2168-9830.2007.tb00945.x

Chase WG, Simon HA (1973) Perception in chess. Cogn Psychol 4(1):55–81. doi:10.1016/0010-0285(73)90004-2

Clegg BA, Digirolamo GJ, Keele SW (1998) Sequence learning. Trends Cogn Sci 2(8):275–281. doi:10.1016/S1364-6613(98)01202-9

Curran T, Keele SW (1993) Attentional and nonattentional forms of sequence learning. J Exp Psychol Learn Mem Cogn. doi:10.1037/0278-7393.19.1.189

Egan DE, Schwartz BJ (1979) Chunking in recall of symbolic drawings. Mem Cogn 7(2):149–158. doi:10.3758/BF03197595

Goldschmidt G, Rodgers PA (2013) The design thinking approaches of three different groups of designers based on self-reports. Des Stud 34(4):454–471. doi:10.1016/j.destud.2013.01.004

Kotovsky K, Simon HA (1973) Empirical tests of a theory of human acquisition of concepts for sequential patterns. Cogn Psychol 4(3):399–424. doi:10.1016/0010-0285(73)90020-0

Kotovsky K, Simon HA (1990) What makes some problems really hard: Explorations in the problem space of difficulty. Cogn Psychol 22(2):143–183. doi:10.1016/0010-0285(90)90014-U

Kotovsky K, Hayes J, Simon HA (1985) Why are some problems hard? Evidence from tower of Hanoi. Cogn Psychol 17(2):248–294. doi:10.1016/0010-0285(85)90009-X

McComb C, Cagan J, Kotovsky K (2015a) Rolling with the punches: an examination of team performance in a design task subject to drastic changes. Des Stud 36(1):99–121. doi:10.1016/j.destud.2014.10.001

McComb C, Cagan J, Kotovsky K (2015b) Lifting the veil: drawing insights about design teams from a cognitively-inspired computational model. Des Stud 40:119–142. doi:10.1016/j.destud.2015.06.005

Meier C, Yassine AA, Browning TR (2007) Design process sequencing with competent genetic algorithms. J Mech Des 129(6):566–585. doi:10.1115/1.2717224

Moss J, Cagan J, Kotovsky K (2004) Learning from design experience in an agent-based design system. Res Eng Des 15:77–92. doi:10.1007/s00163-003-0042-4

Nissen MJ, Bullemer P (1987) Attentional requirements of learning: evidence from performance measures. Cogn Psychol 19(1):1–32. doi:10.1016/0010-0285(87)90002-8

Novick LR, Tversky B (1987) Cognitive constraints on ordering operations: the case of geometric analogies. J Exp Psychol Gen 116(1):50–67. doi:10.1037//0096-3445.116.1.50

Page L, Brin S, Motwani R, Winograd T (1999) The PageRank citation ranking: bringing order to the web. Stanford InfoLab

Perruchet P, Amorim M-A (1992) Conscious knowledge and changes in performance in sequence learning: evidence against dissociation. J Exp Psychol Learn Mem Cogn 18(4):785–800

Pretz JE, Naples AJ, Sternberg RJ (2003) Recognizing, defining, and representing problems. In: Davidson JE, Sternberg RJ (eds) The psychology of problem solving. Cambridge University Press, Cambridge

Radcliffe DF, Lee TY (1989) Design methods used by undergraduate engineering students. Des Stud 10(4):199–207. doi:10.1016/0142-694X(89)90002-1

Raftery AE (1985) A model for high-order Markov chains. J R Stat Soc Ser B (Methodol) 47 (3):528–539

Reed J, Johnson P (1994) Assessing implicit learning with indirect tests: determining what is learned about sequence structure. J Exp Psychol Learn Mem Cogn 20(3):585–594. doi:10.1037/0278-7393.20.3.585

Reitman JS (1976) Skilled perception in go: deducing memory structures from inter-response times. Cogn Psychol 8(3):336–356. doi:10.1016/0010-0285(76)90011-6

Rogers J (1996) DeMAID/GA—an enhanced design manager's aid for intelligent decomposition. In 6th symposium on multidisciplinary analysis and optimization, American Institute of Aeronautics and Astronautics. doi:10.2514/6.1996-4157

Scherr AL (1962) An analysis of time-shared computer systems. Massachusetts Institute of Technology

Sen C, Ameri F, Summers JD (2010) An entropic method for sequencing discrete design decisions. J Mech Des 132(10):101004. doi:10.1115/1.4002387

Seneta E (2006) Markov and the creation of Markov chains. In Markov Anniversary Meeting 2006, 1–20

Simon HA, Kotovsky K (1963) Human acquisition of concepts for sequential patterns. Psychol Rev 70(6):534–546. doi:10.1037/h0043901

Stempfle J, Badke-schaub P (2002) Thinking in design teams—an analysis of team communication. Des Stud 23(5):473–496. doi:10.1016/S0142-694X(02)00004-2

Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J R Stat Soc Ser B (Methodol) 36(2):111–147

Stroock DW (2005) An introduction to Markov processes. Springer, New York

Tamir A (1998) Applications of Markov chains in chemical engineering. Elsevier, Amsterdam

Todd D (1997) Multiple criteria genetic algorithms in engineering design and operation. University of Newcastle

Waldron MB, Waldron KJ (1988) A time sequence study of a complex mechanical system design. Des Stud 9(2):95–106. doi:10.1016/0142-694X(88)90035-X

Willingham DB, Nissen MJ, Bullemer P (1989) On the development of procedural knowledge. J Exp Psychol Learn Mem Cogn 15(6):1047–1060. doi:10.1037/0278-7393.15.6.1047

Designerly Pick and Place: Coding Physical Model Making to Inform Material-Based Robotic Interaction

Daniel Smithwick, David Kirsh and Larry Sass

Abstract To study how designers explore ideas when making physical models we ran an experiment in which architects and undergraduate students constructed a dream house made of blocks. We coded their interactions in terms of robotic pick and place actions: adding, subtracting, modifying and relocating blocks. Architects differed from students along three dimensions. First, architects were more controlled with the blocks; they used fewer blocks overall and fewer variations. Second, architects appear to think less about house features and more about spatial relationships and material constraints. Lastly, architects experiment with multiple block positions within the model more frequently, repeatedly testing block placements. Together these findings suggest that architects physically explore the design space more effectively than students by exploiting material interactions. This embodied know-how is something next generation robots will need to support. Implications for material-based robotic interaction are discussed.

Introduction

Much of what has been said of designerly ways of knowing is about manual sketching activity: it is the means by which designers have 'reflective conversations' with their design (Schon 1992); it is how designers 'see as' and 'see that' (Goldschmidt 1991); it situates designers and enables them to 'think on the fly' (Suwa et al. 1998); and it enhances a designer's ability to perceive visual-spatial features and conceive multiple design ideas (Bilda and Demirkan 2003). Indeed, much of design knowledge initially takes form as exploratory sketching activity.

D. Smithwick · D. Kirsh · L. Sass
MIT-SUTD International Design Center, Singapore, Singapore

D. Smithwick · D. Kirsh · L. Sass
University of California, San Diego, USA

D. Smithwick (✉) · D. Kirsh · L. Sass
MIT, Cambridge, USA
e-mail: djs2@mit.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_23

However, sketching is not always necessary in design activity (Bilda et al. 2006) and it is not the only means of exploring a design manually. Architects also make physical models, including: sketch models, diagram models, concept models, massing models, presentation models, and more (Mills 2011). Gursoy and Oskar (2015) distinguish two types of architectural models. Models *of* architecture represent ideas that have already been resolved; models *for* architecture explore and develop new ideas.

In this paper we are concerned with models for architecture. Unlike sketching, which involves marking a 2 dimensional paper surface, physical model making takes place in three dimensional space and involves different forms of material interaction. For instance, different modeling materials afford different actions: chipboard can be layered, paper can be folded, wood can be milled, concrete can be cast, and so forth. Schon (1992) says such complexity gives way to entirely different design worlds. Moreover because physical models have many sides what one sees depends on one's position in relation to the model and on the many different ways of physically manipulating the model.

If sketching is thought of as *visual* design thinking (Goldschmidt 1994), model making, with its emphasis on building, assembly and manipulation, ought to be considered *physical* design thinking—a more tangible, interactionist way of exploring designs. Yang (2005) calls 3-D prototyping a unique design language quite distinct from sketching. Yet very little has been said about model making as a kind of designerly knowing in action. Physical model making has been found to be beneficial by giving students hands-on opportunities to test and refine design concepts (Lemons 2010) and by reducing fixation (Youmans 2011). But there is more to say. Since "A designer's knowing in action involves sensory, bodily knowing" (Schon 1992, p. 5) it ought to be possible to go beyond visual thinking to explain how designers think with their bodies.

Framing certain forms of action as enactive thinking is central to theories of embodied and situated cognition in cognitive science (Anderson 2003). Gesture can facilitate thinking in calculation (Martin and Schwartz 2005) and problem solving (Goldin-Meadow and Beilock 2010). Body movement enables dancers to probe movement structures in ways inaccessible through observation alone (Kirsh 2011). Some have even argued that because of the role played by external resources during thinking the process of thought may actually extend beyond the brain into one's physical interactions with material objects beyond the body (Clark 2008).

Aims

Our objective here is to extend the notion of designerly thinking to physical model making. What sketching is to visual thinking model making is to physical thinking. To explore this idea we created a simple design world—a *blocks world*—where model making is abstracted to picking, manipulating, and placing blocks in a configuration on a site. We devise a simple coding scheme that tracks the key material interactions over time. By video recording the design sessions of designers and non-designers we are able to compare their interactions and by using the method of voice aloud protocol analysis we can relate these to what is said during activity. Assuming designers do behave differently than non-designers, our central concern is to elucidate a framework that will reveal these differences and explain them.

Significance

Our ultimate motivation is to provide a theoretical and empirical foundation for developing interactive robotic design tools that enhance the material-based nature of exploratory model making. A first step in creating new tools is to analyze and understand a particular design activity in specific terms that the tool could potentially enhance or automate. The tool we are interested in developing is an interactive robotic manipulator arm, therefore in this study we strive to understand what designers do in terms of picking and placing physical material.

We believe that formalizing how designers interact with blocks offers insights into how physical forms of design thinking can be enhanced. Many projects illustrate the potential for such tools (e.g. Willis et al. 2011; Zoran and Paradiso 2013). However they do not make claims for enhancing design thinking. Some researchers have begun exploring how interactive manipulatives can enhance creativity (Grace 2014). Other researchers have asked how making as a kind of design-in-action can be formalized in terms of generative processes (Knight and Stiny 2015) and evaluative processes (Bernal et al. 2015). Our work has implications for how material interaction and body movement can be enhanced with robot-aided design processes.

Background: Previous Protocol Analysis Methods

Very few protocol analysis studies (those that either utilize coding schemes or segmentation methods) have been conducted on designing with physical media (e.g. Lemons 2010; Kim and Maher 2008). In a small scale study (n = 6), Kim and Maher (2008) compare the use of TUIs versus GUIs to determine how hands-on interaction changes spatial cognition in a design task. Their coding scheme is based on those developed in Suwa and Tversky (1997) and Suwa et al. (1998) including the following categories of analysis: action, e.g. drawing and looking; perception, e.g. perceiving visual-spatial features; process, e.g. setting up goals; and lastly, collaborative, e.g. communicative gestures. They found that TUIs support exploratory activity in terms of epistemic actions and that the ‘naturalness’ of interacting with physical objects enhances spatial reasoning and visual-spatial perception by offloading cognitive tasks to the environment.

To explain the benefit of hands-on model making in engineering design, Lemons (2010) had participants construct models with Lego bricks. They focused on their subjects' accounts of what they were thinking, using verbal protocols exclusively and did not code the different physical actions performed such as joining, disjoining, rotating or sorting bricks.

Many more studies have been conducted on revealing the benefits of interaction with physical media although they do not utilize protocol analysis methods (e.g. Yang 2005; Youmans 2011; Fjeld and Barendregt 2009; Klemmer et al. 2008), In a large scale study (n = 120), Youmans (2011) compared the design performance outcome benefits in a tool design exercise of having access to physical prototyping materials versus having no access. He found that with access to materials designers would fixate less on previous tool design examples given in the exercise and produce more novel and better functioning tools designs.

Predominantly, sketching has been the activity that is analyzed in protocol studies (e.g. Suwa et al. 1998; Bilda et al. 2006; Suwa and Tversky 1997; Kavakli 2002). Typically, video recording is used to capture the design activity followed by retrospective reporting, where the participant is asked to watch himself in the video and verbally account for what he was thinking. Another approach, known as the concurrent thinking aloud method, which we report here, is to have participants describe what they are thinking while in the design session. Concurrent reporting is thought to provide a more accurate depiction of how thought unfolds over time (Suwa et al. 1998). With retrospective reporting the account may be more thorough and less disruptive of the design process (Kavakli 2002) but potentially less true to the moment.

Many protocol studies are based on Suwa and Tversky's coding scheme (1997) which identified four information categories that architects see and think about while sketching: emergent properties, spatial relationships, functional relationships and background knowledge. Suwa et al. (1998) adapted these categories to correspond to the flow of cognitive processes involved in human cognition: distinguishing physical, perceptual, functional and conceptual action. Physical actions include drawing, looking, and gesturing. Perceptual actions involve attending to features, relationships, and making comparisons. Functional actions include relating non-visual information with spatial features e.g. circulation of people through rooms. Conceptual actions involve setting up goals and making value judgments based on domain knowledge. Physical and perceptual action can be identified by watching the video recording. Functional and conceptual action are identified from a designer's verbal account.

By examining the flow of actions Suwa et al. (1998) found that physical actions, along with perceptual actions are drivers of design thinking in sketching just as much as background knowledge and predetermined goals. In other words they found that physical and perceptual action is able to drive thinking forward as much as conceptual and functional. They concluded that the physical action of sketching enables designers to 'think on the fly'.

In the current work, we are interested particularly in how physical action in model making drives design thinking and in what ways this may differ between experts and novices. Looking more closely at the kinds of physical actions defined by Suwa et al. (1998), however, we find their action types ill-suited for describing material interactions in model making. They distinguish these seven:

• Revise the shape, size or texture of a depiction • Create a new depiction • Trace over a depiction on same/new sheet	• Depict a symbol • Write sentences or words • Look at previous depiction • Move a pencil/depiction

In physical model making one may perform these actions in the course of making a model, e.g. to depict a shape to be cut out of paper, but this would be a sketching action *plus* some other physical action. Strictly speaking, model making means working with material in hand. The actions possible depend largely on the type of material, opening up a wide range of interactions: folding, twisting, laminating, stacking, sorting, cutting, milling, pouring, and so on. In addition, model making commonly involves the application of skill and technique with tools as diverse as knives, drill presses, laser cutters, and 3D printers.

Methodology

Experiment Setup

In our experiment participants were asked to build a physical model of their dream house by arranging blocks on a wooden site model (Fig. 1, left). The task was open-ended: no specific constraints were imposed, such as building a minimum number of rooms or meeting a square footage requirement. Nor was a particular scale given; a single block could represent the entire house or a piece of furniture. Participants were supplied with 44 3D printed parallelepiped-shaped blocks and free to use as many or as few as they wished. The parallelepiped shape enabled fairly complex assemblies with varied spatial relationships. No other tools or medium (e.g. no pencil or paper) was allowed. Participants were given 15 min to complete the task. Prior to that they had a minute or two familiarize themselves with

Fig. 1 The experiment site model and 3D printed blocks, *left*; a robotic arm picking and placing blocks, *right*. How can model making be supported by this tool?

Table 1 Participant groups

Architects	3	3M, 0F
Students in architecture	3	1M, 2F
Students in non-design fields	3	2M, 1F
Total	9	6M, 3F

the look and feel of the blocks. Once they began the experiment they were encouraged to voice aloud any thoughts they had concerning their design or their process as they manipulated blocks. Video/audio was captured looking down on the site model and photographs were taken throughout each session. The participants were not told how the dream house would be evaluated other than that once the 15 min was over they were to walk the researcher through their model.

Participants

Three participants were architects with 4–8 years of professional practice as well as teaching experience at the graduate level. The student participants were spilt into two groups. Three students were undergraduates majoring in architectural design, and three were undergraduates from non-design related departments. With the exception of one architect, all participants were from the same academic institution. See Table 1.

Hypothesis

Our hypothesis was that in an exploratory design task requiring physical manipulation, there will be significant differences between the way architects and students interact with materials. We expected student participants with design training also to show differences with novices. But we doubted those differences would do more than trend toward significance.

Why Make a Dream House by Picking and Placing Blocks?

Building a dream house is an architectural program accessible to both architects and non-architects. Most people at some point have imagined what their dream house might look or feel like and if they haven't developed evolved ideas they typically have likes and dislikes they can share. For experimental tractability we constrained the task, however, to arranging blocks. This limits the possible architectural forms but leveled the playing field by making irrelevant differences in technical skill or specialized training.

We also limited possible operations to those which may in principle be performed by a robotic manipulator arm: picking, transforming and placing blocks (Fig. 1, right). Given our goal of understanding how a robotic manipulator arm may support early stage conceptual design our work here sets the stage for informed speculation.

Coding Scheme

The goal of our coding scheme is to analyze the physical actions and the talk aloud protocols captured on video in order to understand the role physical action plays in design thinking. We do not code for perceptual, functional or conceptual actions as have previous studies. Instead we focus exclusively on physical action. Coding for functional and conceptual actions requires retrospective reporting by the participant (Suwa et al. 1998). Perceptual actions are inherently dependent on physical actions so we collapse these two categories together.

Pick-Manipulate-Place Framework

In our model making activity one works with discrete objects and arranges them in different relationships called configurations. The making of configurations can be characterized as a three-stage process: picking, manipulating (rotation, translation), and placing the objects. Picking and placing are defined by location in the world, either within a configuration or on the site. Manipulation is what a participant does after picking up a block and before placing it. We consider only rotations and translations, not such things as squeezing, spinning or other physical actions. Each manipulation has a duration, measured in seconds.

Primitive Interaction Types

Based on the pick-manipulate-place framework we construct four primitive interaction types: *adding* blocks to a configuration, *subtracting* blocks from a configuration, *modifying* blocks within a configuration, and *relocating* blocks across the site. Each of these four types is defined by the locations of the picking and placing actions as identified in Table 2. For each participant we counted each occurrence of an action type and measured its duration.

Table 2 Definition of interaction types by pick and place location

Interaction type	Pick location	Place location	Description
ADD	Site	Configuration	Participant adds block from the site to the configuration
SUBTRACT	Configuration	Site	Participant removes block from configuration and places on site
MODIFY	Configuration	Configuration	Participant adjusts block within configuration only
RELOCATE	Site	Site	Participant moves block across the site

Interaction Sequences

An interaction sequence is a pattern of interactions over time. We observed three types of sequencing:

- *Consecutive interactions* involve three or more identical actions in a row. For example, Add–Add–Add or Modify–Modify–Modify.
- *Compound interactions* occur when the Place location of one interaction becomes the Pick location of next action. For example, when a subject Adds a block to a configuration and without putting it down, does a Modify action, such as adjusting its orientation etc. We count the adjustment as a new action despite the participant never releasing the block.
- *Coordinated interactions* are two-handed actions. There are two types: Synchronous and Asynchronous. For example a subject may modify the orientation or shift its location using both hands. We call this a synchronous interaction because both hands are involved in the same action. Asynchronous interaction is doing two different interactions with two hands, such as picking up two blocks or moving two apart. See Table 3 for all interaction sequence definitions.

Table 3 Definition of interaction sequence types

Interaction sequence type	Name	Interactions
CONSECUTIVE	Manage	Relocate–relocate–relocate
	Disassemble	Subtract–subtract–subtract
	Assemble	Add–add–add
	Explore	Modify–modify–modify
COMPOUND	Test:reject	Add–modify–subtract
	Test:affirm	Add–modify
	Test:eject	Modify–subtract
COORDINATED	Synchronous	Modify/modify, e.g.
	Asynchronous	Add/modify, e.g.

Results

Despite our small sample size there were important differences between the three participant groups. At a structural level there were significant differences in the block count, block variation, and block arrangement of the final model. At a protocol level there were suggestive differences but variance in the quantity of talk aloud results prevented us from finding anything more than anecdotal differences. At an interaction level, however, there were clearer differences in the primitive interaction counts as well as trends in the interaction sequences across the three groups.

Dream House Models

Figure 2 shows renderings of the participants' final models. Each rendering is an orthographic projection of the model. It should be noted that the projection angle was chosen in order to provide the best overall visibility of the blocks in the model.

Block Count

Architects tended to use fewer blocks in their models than the other two groups. Their mean number of blocks used was 11.0, whereas the novices used 17.3 and the non-designers used 19.0. We found that the novices and non-designers would often keep adding blocks to their configuration until the supply ran out. The experts seemed to identify a limited set of preferred blocks, e.g. all red blocks, and then work with those. See Table 4.

Block Variations

Architects also used fewer variations of blocks. Ten variations of blocks were provided. Two experts only used 2 variations of the blocks—in one case both were red blocks (Fig. 2, E1) and in the other case the architect used only the blue and wire-frame blocks (Fig. 2, E3). The novice student designers used 6.3 (mean) block variations; the non-design students used 7.3, while the architects used 3.3 block variations.

Based on visual analysis of the models, the architects appear to be more selective and controlled providing them with further problem constraints. Why might this be the case? We seek evidence in the verbal protocols that may shed light on the architects' thought processes.

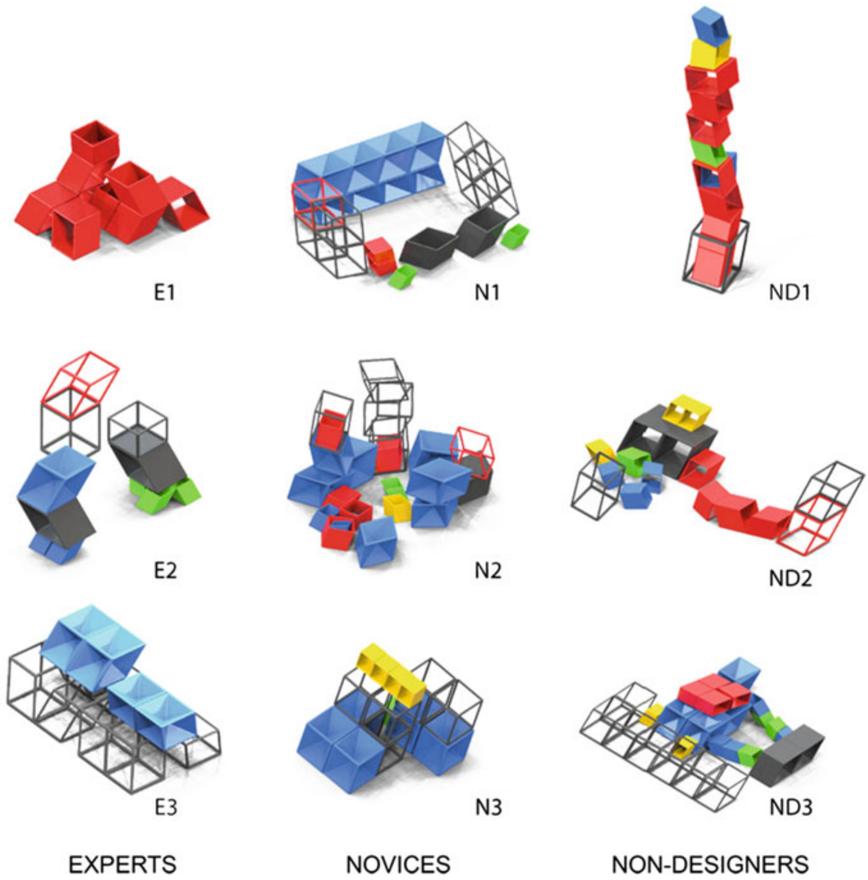


Fig. 2 Renderings of the final dream house models by Architects, *left*; novice design students, *middle*; and non-design students, *right*

Table 4 Block count and variation in the dream house models

	Architects			Novices			Non-designers		
	E1	E2	E3	N1	N3	N2	ND1	ND2	ND3
Block count	9	12	12	22	15	22	12	16	29
Block variations	2	6	2	6	4	9	6	8	8

Verbal Protocol Results

We do not code or segment the verbal protocols as have previous protocol studies (e.g. Lemons 2010); we admittedly only present anecdotal evidence from a few thoughts voiced by participants during the session. We have chosen the quotes below to convey general differences between the three groups. The key difference we find is the generative approach voiced by the architects and the more representational approach taken by the novice and non-design students.

For example, one novice design student described her process as one of finding blocks to represent typical household features:

> I'm using this [block] as the entrance…and I'm going to use these [blocks] as the grass and flowers…and these [wireframe blocks] will be windows (Fig. 2, N1).

A non-design student voiced a similar approach, expressing practical concerns about the house, commenting that:

> It'd be kind of cool to have a wall of windows along the side of the house…maybe the south facing side where there's a lot of sun (Fig. 2, ND3).

Furthermore, the participant was thinking about particular rooms and how to connect them:

> This is a wide open living room, here's an open hallway on the side of the building…the little blocks I'm treating as hallways and maybe stairs, this is the kitchen, here is a studio… there's bedrooms on the second floor.

Architects had a different approach. They commented on the effect of moving shapes around to explore the spatial relationships between blocks. One architect started off his design session by exploring the blocks' properties, saying:

> Using the blocks, I am creating face-matching walls.

He was giving himself a constraint—joining the blocks face-to-face—and seeing what relationships were possible. While exploring these relationships he developed design goals:

> I'm looking for…what I'm hoping this will be is some kind of dividing wall…or on this side I want to have an elevated mass…held up by three [blocks]…because I want the idea of danger to be present.

Another architect remarked very early on in the session about the design potential embedded within the blocks, saying:

> I like that these [the red blocks] create an 'outdoor/indoor'…a kind of exterior spatial definition and an interior spatial definition…it's super strong…and you don't have to do much to let them do that…which is nice (Fig. 2, E1).

With this realization he developed a design concept early on and was able to give himself a constraint by working with a limited set of blocks, saying:

> I didn't intermingle the blocks because I like the way the red ones fit together. It's more controlled for me.

Another notable difference was that the architect was not thinking about the blocks in terms of household features:

> I can't go through it and say what's my bedroom, what's my living room…but as a plain figure I like it and can imagine it occupied in many ways.

Material interaction for the architects was more like visual-kinesthetic experimentation rather than fitting blocks to a preconceived idea. We now look at the interaction protocols to seek evidence for how this experimentation is reflected in physical action.

Interaction Results

We first coded each of the participant's video session in terms of primitive interactions (add, subtract, modify, relocate). See Table 5. For example, subject E2 did 27 Add actions. This accounted for 24.5% of all of his interactions.

We did a Chi-Square analysis to test the null hypothesis that the physical action data represent a chance distribution. The result is $x^2(6, n = 9) = 32.22$, $p < .01$. This is a very low probability and we can therefore confidently reject the null hypothesis and claim there is a significant relationship between participant group and interaction count. But what is the relationship?

The charts in Fig. 3 reveal two differences between the participant groups: the percentage of Modify interactions and Relocate interactions. Differences in Add and Subtract interactions across the groups are minimal, which is what one would expect given that the task was to build a structure out of blocks. Comparing just the architects with the non-design students we see that the architects more often Modify blocks and less often Relocate blocks. By devoting more action to Modifying blocks within the model architects are able to cycle through and see more potential block relationships.

Table 5 Interaction primitive counts per participant organized by group

Interaction type	Architects			Novices			Non-designers		
	E1	E2	E3	N1	N2	N3	ND1	ND2	ND3
# of ADD actions	11	27	16	42	36	26	21	22	33
% of total	25.0	24.5	28.1	28.6	22.2	24.5	39.6	23.9	31.1
Time/action	1.0	1.0	1.0	1	1.1	1	1.2	1.1	2.5
# of SUBTRACT actions	4	12	4	9	13	9	6	5	7
% of total	9.1	10.9	7.0	6.1	8.0	8.5	11.3	5.4	6.6
Time/action	1.0	1.0	1.0	1.0	1.5	1	1	1	1
# of MODIFY actions	19	39	20	59	90	35	13	31	17
% of total	43.2	35.5	35.1	40.1	55.6	33.0	24.5	33.7	16.0
Time/action	5.5	6.6	3.5	5.0	2.6	4.5	4.9	4.2	2.8
# of RELOCATE actions	10	32	17	37	23	36	13	34	49
% of total	22.7	29.1	29.8	25.2	14.2	34.0	24.5	37.0	46.2
Time/action	1.1	3.5	1.4	1.9	2.0	1.4	2.4	4.7	1.8
Total action time	130	410	113	429	340	243	126	335	227
Total actions	44	110	57	147	162	106	53	92	106

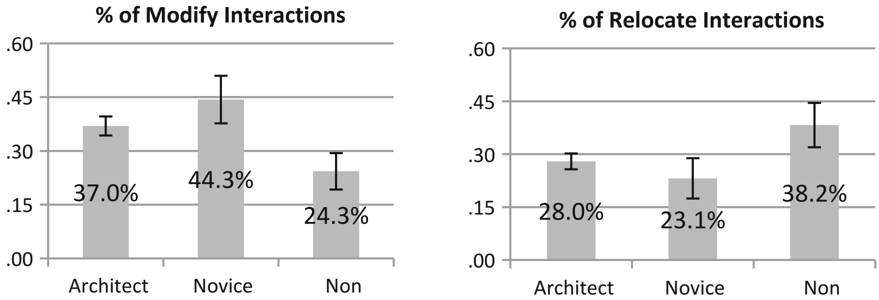


Fig. 3 Mean percentage of modify and relocate interactions by group. *Error bars* show standard error of the mean

We characterize this as experimental model-centric interaction: position a block this way in relation to its neighbors and see what it looks like; reposition it another way and see what that looks like, and so forth. On the other hand non-design students interact with blocks more often on an individual basis, isolated from the model.

However, we do not find a progression across the three groups; why do the novices perform more Modify interactions and fewer Relocate interactions than the architects? First, there may be different kinds of Modify and Relocate interactions that our framework does not distinguish between. For example, there may be additional Modify interactions that only serve to fix positioning errors. A second possible answer may lie in the context in which interactions proceed. To explore this possibility we look to the sequence of interactions.

Experimental and Model-Centric Interaction Sequences

Overall, we find that as a percentage of total interactions there is a trend from architects to novices to non-designers in structuring their interactions as Consecutive and Compound sequences. For architects, this is 60.7% $\pm$ 3.1; for novice designers, 50.8% $\pm$ 1.5; and for non-design students 43.8% $\pm$ 3.2. This suggests that architects, more so than the other groups, have top-down interaction strategies.

Looking more closely at the individual sequence types listed in Table 3, we find further trends across the three groups. For example, in the Consecutive sequences there are trends in Manage, Assemble and Explore interactions (Fig. 4). As a percentage of their total, the architects Manage and Assemble the blocks the least, and Explore with the blocks the most. The architects and the novice design students are grouped closely, suggesting overlapping behaviors amongst the individuals; however, there is clear distinction between the architects and the non-design students. For example, the non-design students perform Assemble interaction

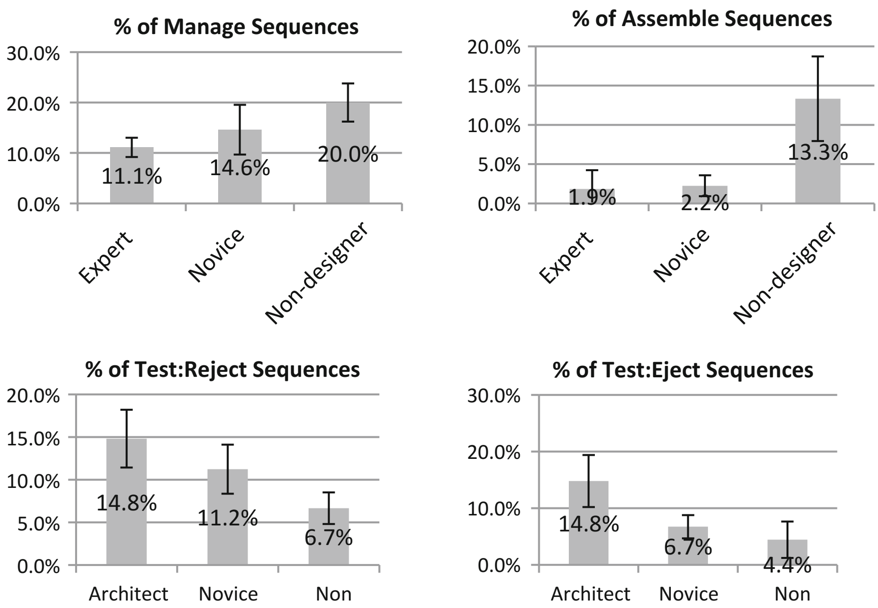


Fig. 4 Percentage of consecutive (*above*) compound (*below*) sequences types by group. *Error bars* show standard error of the mean

sequences seven times as much as the architects; they perform Explore sequences less than half as much.

With Compound sequence types there are similar trends across the groups (Fig. 4). Most revealing are trends found in Test:Reject and Test:Eject sequences. Architects perform over two times as many Reject interaction sequences and over three times as many Eject sequences compared to non-design students. It seems the architects are more selective when choosing which blocks to keep in the model.

The Role of Modify Interactions

These trends in selectivity also provide a more nuanced explanation of the role of Modify interactions for the architects. At first glance one would think that architects would perform Modify interactions the most out of the three groups (see Fig. 3). Instead of just adding and assembling blocks, as if constructing a preconceived idea, a designer would 'think on the fly', discover 'unexpected' ideas, and do more with less by modifying blocks within the configuration. However, the architects performed fewer Modify interactions than the novice design students. How do we account for this?

Framed within a Compound sequence a Modify action can be thought of as an experimental embodied interaction after which one decides to either accept or reject a block. While the novice may perform more Modify interactions they may also pay less attention to the results of the experiment. Evidence for this is suggested by the novices' lower rate of Reject and Eject sequences (Fig. 4) and higher rate of Accept sequences (novice = 56.2% $\pm$ 4.7; expert = 48.1% $\pm$ 1.1). In other words, if one more often accepts the experiment result, i.e. keeping a block in the configuration, it may be because they set up more predictable experiments. We believe architects set up better embodied interaction experiments.

Discussion

Limitations of Making Physical Models with Computational Tools

Current digital design and fabrication model-making processes are set up for automation and assembly and do not support hands-on model making interactions such as the ones analyzed in our study. The limiting factor in digital fabrication is that a model must be defined before it is made physical. To make a physical model, it must first be designed visually on a computer screen; then discretized into manufacturable parts, and then finally assembled together (Sass 2008).

Robotic manipulator arms currently implemented in digital fabrication processes either replace CNC machines as more advanced fabrication tools, such as in multi-axis routing or 3D printing, or they are used to automate the assembly of complex geometric structures (Kohler et al. 2014).

Material-Based Robotic Interaction

We take a different approach. Our vision is to integrate robotic tools into the early exploratory stages of a design process where design ideas are still being formed (Fig. 5). The challenge is to computationally support the material-based interactions that we observed in hands-on model making. To do so we need to altogether rethink the way designers use computational tools and processes. Instead of visualizing a design on a computer screen before making it physical, designers ought to be able to work directly with material. As roboticist Rodney Brooks famously said, "The world is its own best model." We interpret this to mean that visualization should happen in real-time as the physical model develops. In other words, the physical model should be its own visualization.

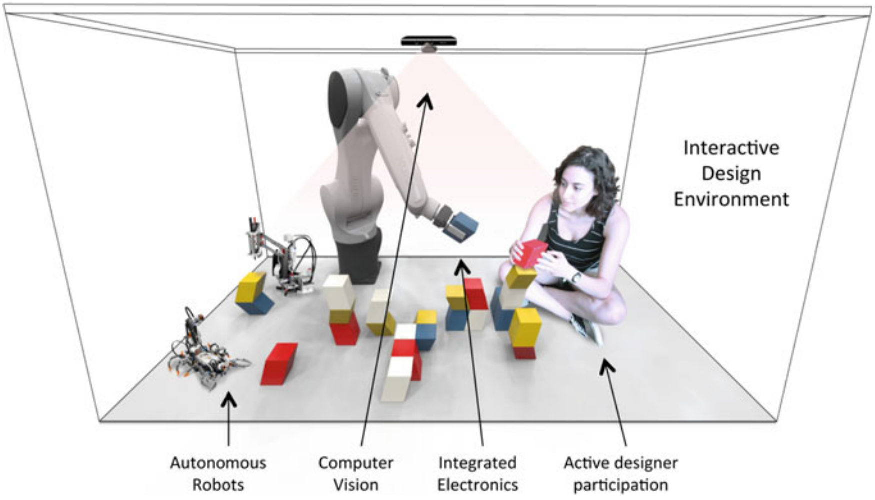


Fig. 5 Imagined robotic-aided design environment

Robotic manipulator arms have the potential to enhance material interaction only if they are equipped with vision and object recognition functionality. What it needs to be able to 'see' is if material has been added to a model or subtracted from it; whether material has been modified within the model, or whether material has been relocated independent of the model. These are the ways we observed designers to interact with materials in model making activity.

The question still remains: how can designers control a robotic manipulator arm without expending more energy in communicating their intent than simply doing the action themselves? What we propose builds on typical CAD-based commands such as creating geometric reflections, copies, arrays, or rotations, but uses material interaction itself as the control interface. For example if a vision system can see that the designer has added material, the robotic arm can be programmed to copy that particular interaction, or array that interaction, or rotate it. In this sense the designer learns to think through potential robotic actions that duplicate and extend his own.

To achieve these goals we must re-imagine the designer's work environment. It must consist of a site that is physically accessible to the designer and to a robotic manipulator arm. The manipulator arm must be fitted with an end-effector appropriate to the material. Some material can be rotated and translated, e.g. blocks, but other material can be folded, casted, or cut. A camera must be able to sense the work environment. With 3D material multiple cameras will be needed to support interaction. Lastly, safety is a concern if designers are to physical interact with robotic tools. The ability to sense the designer in the work environment will be needed to avoid collision.

Conclusion

When working with physical media we found that architects work with material in ways that can be characterized in four simple interactions: adding, subtracting, modifying, and relocating material. We claim their interactions are 'designerly' and therefore supportable by design tools by distinguishing their interactions from those of non-designers. To simplify the analysis we created a blocks world in which model making activity is constrained to picking, transforming, and placing blocks on a site. By formalizing and distinguishing an architect's interactions our work sets the stage for developing interactive robotic tools that can enhance exploratory model making.

Acknowledgements The authors would like to thank the MIT-SUTD International Design Center for funding this research.

References

Anderson M (2003) Embodied cognition: a field guide. Artif Intell 1–40

Bernal M, Haymaker J, Eastman C (2015) On the role of computational support for designers in action. Des Stud 41:163–182

Bilda Z, Demirkan H (2003) An insight on designers' sketching activities in traditional versus digital media. Des Stud 24:27–50

Bilda Z, Gero J, Purcell T (2006) To sketch or not to sketch? That is the question. Des Stud 27:587–613

Clark A (2008) Supersizing the mind. Oxford University Press, Oxford

Fjeld M, Barendregt W (2009) Epistemic action: a measure for cognitive support in tangible user interfaces? Behav Res Methods 41(3):876–881

Goldin-Meadow S, Beilock S (2010) Action's influence on thought: the case of gesture. Perspect Psychol Sci 5(6):664–674

Goldschmidt G (1991) The dialectics of sketching. Creat Res J 4(2):123–143

Goldschmidt G (1994) On visual design thinking: the vis kids of architecture. Des Stud 15(4): 158–174

Gursoy B, Ozkar M (2015) Visualizing making: shapes, materials, and actions. Des Stud 41:29–50

Kavakli Gero J (2002) The structure of concurrent cognitive actions: a case study on novice and expert designers. Des Stud 23:25–40

Kim M, Maher M (2008) The impact of tangible user interfaces on spatial cognition during collaborative design. Des Stud 29:222–253

Kirsh D (2011) Thinking with the body. In: Proceedings from the 30th annual cognitive science society

Klemmer S, Everitt K, Landay J (2008) Integrating digital and physical interactions on walls for fluid design collaboration. Hum Comput Interact 23:138–213

Knight T, Stiny G (2015) Making grammars: from computing with shapes to computing with things. Des Stud 41:8–28

Kohler M, Gramazio F, Willmann J (2014) The robotic touch: how robots change architecture. Park Books, Zurich

Lemons G (2010) The benefits of model building in teaching engineering design. Des Stud 31:288–309

Maher M, Gonzalez A, Grace, K, Clausner T (2014) Tangible interaction design: can we design tangibles to enhance creative cognition? In: Sixth international conference on design computing and cognition, London, UK, 23–25 June 2014

Martin T, Schwartz D (2005) Physically distributed learning: adapting and reinterpreting physical environments in the development of fraction concepts. Cogn Sci 29:587–625

Mills C (2011) Designing with models. Wiley, New York

Sass L (2008) A physical design grammar: a production system for layered manufacturing machines. Autom Constr 17:691–704

Schon D (1992) Designing as reflective conversation with the materials of a design situation. Behav Res 5(1):3–14

Suwa M, Tversky B (1997) What do architects and students perceive in their design sketches? A protocol analysis. Des Stud 18:385–403

Suwa M, Purcell T, Gero J (1998) Macroscopic analysis of design processes based on a scheme for coding designer's cognitive actions. Des Stud 19:455–483

Willis K et al (2011) Interactive fabrication: new interfaces for digital fabrication. Tangible, embedded, and embodied interaction Conference, Portugal

Yang M (2005) A study of prototypes, design activity and design outcome. Des Stud 26:649–669

Youmans (2011) The effects of physical prototyping and group work on the reduction of design fixation. Des Stud 32:115–138

Zoran A, Paradiso J (2013) FreeD—a freehand digital sculpting tool. In: The 31st international conference on human factors in computing systems. ACM, Paris, France

Knowledge Distribution and the Effect of Design Tools on the Design Process

Mina Tahsiri, Jonathan Hale and Chantelle Niblock

Abstract This paper compares the cognitive performance of architecture students when designing tasks using one of the three design tools: pencil and paper, software Sketch Up and Rhinoceros 3D. It questions if a design tool can affect when *knowledge* is generated and used in the duration of design activity. This is explored through a protocol 'think aloud' study for which a new coding scheme was developed. The methodology is grounded on the theory of Distributed Cognition and Zhang and Norman's (Cognit Sci 18(1):87–122, 1994) method of 'representational analysis', based on which, *knowledge* is either 'internal' in that it is actively memorized by the designer or is 'external' in that it is implicitly made available via a stimuli like a design tool. Using an Analysis of Variance (ANOVA) test, for the five participants of this study, external *knowledge* generated significantly earlier on within the process when using Sketch Up compared to the other two tools.

Introduction

The design process is understood to be a co-evolving process between problem definition and solution generation, taking the designer from an abstract to a more concrete form of thinking (Dorst and Cross 2001; Maher and Poon 1995). This indicates that the structure of the task space is not fully defined at the outset of the process and is susceptible to change (Gero and Fujii 2000). As Coyne et al. (1987) state, exploring within a space that is only partially defined is the subject of creativity. In their view, creativity in concerned with a search for the "paraphernalia" that can define that space, which they express to be *knowledge*. They further explain that a design system's potential for creativity is in its ability to acquire knowledge but also to control its process and change its own structure. In doing so, a design system makes use of its own components, *knowledge* being one of them, to control the operations and implement required changes. As is explained by Edmonds

M. Tahsiri (✉) · J. Hale · C. Niblock
University of Nottingham, Nottingham, UK
e-mail: mina.tahsiri@nottingham.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_24

(1993), a knowledge-based creative system makes use of *knowledge* to further revise and refine *knowledge*. These accounts suggest that *knowledge* is not only a structural component of the system, but a product of the processes and also an operational tool. Therefore how *knowledge* is created, particularly in the earlier stages of a design process, can be said to influence the designer's evaluation on the fitness of their ideas as they unfold (Coyne et al. 1987; Coyne and Gero 1986; Goel and Pirolli 1992). Arguably it will also have semantic implications, as it will both affect the issues the designer attends to and the depth and generality of their application (Dorst and Cross 2001; Suwa et al. 1998).

The initial stages of a design process are important in cohesively framing the task as it enables deliberation over a wider breadth of issues before committing to the details (Dinar et al. 2011). To ensure this good practice, it is common in design schools to encourage students to spend a good chunk of their time in framing the problem as best as they can, early on in the process. It has been reported that students that succeed in this section are more creative in their productions (Dorst and Cross 2001). This approach reinforces a notion that the earlier *knowledge* is exerted, the more impact it will have on the structure of the task space.

Respectively this paper questions whether a design tool can control the design process by affecting how *knowledge* is distributed throughout the duration of the design process. It questions where along the process, the mean distribution of *knowledge* falls if a designer was to conduct a task using one design tool compared to another. It then examines design behaviour under the influence of the design tool that enabled the earlier average occurrence of *knowledge*.

Given the importance of the topic for design education, five final year architecture students from the University of Nottingham were chosen to participate in a 'think aloud' protocol design experiment. The experiment required each student to conduct three similarly structured tasks, each time with one of the design tools assigned to that task. These design tools were pencil and paper, software SketchUp and Rhinoceros 3D.

Design Tools, Knowledge and Distribute Cognition

Besides the on-going debate on what design tools are better suited for nurturing creative thinking of students, the main interest of this paper in questioning the role of design tools arises from the speculation that they can be more than a peripheral aid to cognition (Baber et al. 2014). In other words, the role of a design tool transcends beyond enabling the representation and communication of ideas formed in the mind into an external and visual format to acquiring epistemic credit (Clark and Chalmers 1998). This notion is grounded on the theory of Distributed Cognition, which defines cognition as a phenomenon extending the boundaries of the skull and the mind, into the environment. Based on this theory, cognition is distributed across people, artefacts and time involved in the execution of a task

(Hollan et al. 2000). The manifestation of cognition into the world is therefore a reflection of an aggregating contribution of all its components and not just processes in the 'mind'.

Notably, the accumulated *knowledge* which the mind holds, is argued by environmental psychologists to be informed by the experience of performances in the real world. Noe (2006) asserts that our *knowledge*, is not only determined by what is done but also our know-how for doing something. This denotes that in thinking, we incorporate both perception of the past and future into the instantaneous present. To explain how future actions can inform present *knowledge*, Grafton et al's (1997) study is relevant. Their study showed that just the naming of the 'use' of a tool has motor valence and effects cognitive processes. Dartnall's (2005) philosophical account also helps understand this further. He states that the world sometimes leaks into the mind, in the sense that we can deploy our sensory abilities offline. Therefore designers can structure their task space and predict their course of action by their *knowledge* of the tool. In this regard Jonson's (2005) study demonstrated how the structure in which digitally native designers attend to design differs from its traditional alternative. He reports that these designers use verbalization rather than freehand sketching as a conceptual tool in the early stages of their process.

On the other hand it could also be claimed that the *knowledge* a designer brings to the process holds a preconception of a manner of performance. Arguably for most designers this would be the *knowledge* accumulated through conducting design tasks via freehand sketching, particularly since it is the method employed by many design schools in the early years of educating students. In this regard, an alternative explanation could be provided for studies such as that conducted by Bilda et al. (2006) where they compared the performance of designers when freehand sketching and when blindfolded (using imagery) and concluded that the actual physical activity of drawing made no significant difference in the overall outcome of the process. Similarity in the designers' cognitive performances could in fact be due to deploying their *knowledge* of freehand sketching both times regardless of using imagery or an actual pencil to design.

Thereby in understanding how a design tool manifests its cognitive role within a process, design has to be regarded as a distributed cognitive task and the role of the design tool has to be incorporated into the scheme by which the process is analysed.

Methodology

In acquiring insight into designer's *knowledge*, it is common in design studies to rely on the designer's verbalization which is the basis of the protocol study methodology. In this methodology, verbal utterances and design actions of structured tasks are segmented and coded based on a coding scheme. The coding scheme provides an ontological perspective of the design process and therefore allows for a specific analysis of cognitive activity (Dickson et al. 2000; Ericsson and Simon 1993). The 'think aloud' method was chosen over retrospective verbalization to obtain a direct correspondence between thought content (*knowledge*) and immediate actions.

Presuming that design can be seen as a distributed cognitive activity, the structure of the tasks for this study had to be designed in a way that enables the *knowledge* provided (through verbalization) to be further categorized into its ‘internal’ and ‘external’ components. The internal components are retrieved from internal memory (the mind), in other words what is memorized and the external components from external memory (the design environment), for example through drawing with the design tool. The *knowledge* the designer manifests through verbalization and/or drawings are an aggregated and abstract representation of processes within both of these task spaces. Figure 1 provides an illustrated diagram of this concept.

The necessity of this categorization rises from the fact that analysis of processes conducted at the level of the aggregated task space, cannot fully demonstrate the underlying cognitive differences. In this respect, Zhang and Norman (1994) in their study on ‘*Representations in distributed cognitive task*’ show how tasks with different internal and external structures can have the same structure when observed at the level of the aggregated task space. They criticize methods used up to that date (1994) in cognitive science studies in analysing distributed cognitive tasks, in that those methods, either focused on internal structure, or when taking external structure into account, failed to separate them from internal structure. Therefore they insist on the importance of “decomposing the representation of a task into its internal and external components so that the different functions of internal and external representations can be identified” (20:89).

Respectively studies conducted thus far on the effect of design tools on the design process suggest that the overall quality of the design process, in terms of idea productivity and frequency of attending to design issues is not significantly affected by the design tool used (Bilda et al. 2006; Yu et al. 2013). Therefore it is suggested that should a design tool affect cognition it will be more on lower level

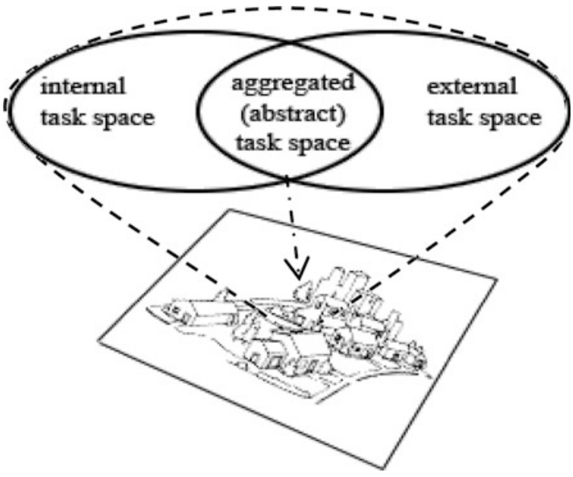


Fig. 1 Task spaces within a distributed cognitive system

activity such as perception or physical activity rather than conception (Tang and Gero 2001). However it may be that we are facing a similar issue as that identified by Zhang and Norman (1994) regarding analysis methods used in cognitive science studies. Similarly it can be argued that in the context of a design experiment, when a designer conducts similar tasks with different design tools, they compensate for the micro internal and external changes in a way that the overall aggregated structure is not significantly affected.

Alternatively Zhang and Norman (1994) propose a method called 'representational analysis'. In their account, a task comprises of a series of rules and the structure of the task space represents the relationships between these rules. In the process of solving a problem, the problem solver thus applies these rules to progress towards the solution. To enable use of their method, they structure the tasks of their study so that some of the requirements require active attention and memorization, which they label as "internal rules". These rules are explicitly made available by the conditions of the task. On the other hand some of the requirements are arranged to be implicitly made available by the internal rules or conditions of the task and therefore do not require to be memorized. This category is labelled as the "external rules". In this research a similar approach is employed, based on which a three phase design task and a coding scheme is proposed. The design processes will then be compared and analyzed at the level of the aggregated, internal and external task spaces.

Hypothesis

Based on what has been discussed thus far, the null hypothesis of this research would be that there is no significant difference between the mean *knowledge* distribution across the design tools. Should this be approved when analysing distribution at the level of the aggregated task space, it will provide evidence for the veracity of this researcher's scheme and degree of reliability of its method of coding as the result is coherent with results from similar research which suggested that tools do not have a significant effect on the overall cognitive activity measures (Bilda et al. 2006; Yu et al. 2013). On the other hand the expectancy is to find differences at the level of internal and external task spaces. In particular, the research is interested with differences in the mean *knowledge* distribution in the external task space, where *knowledge* exclusive to the situation of that particular task is generated and put to use.

The Design Experiments

Each participant was required to conduct three tasks, each with its own assigned design tool. To ensure that performances in each task were comparable to one another, the three tasks all had the same structure but represented in different contexts. This means that the requirements in the briefs were the same, but displayed differently so that the designer is under the impression that they are designing a totally different task each time. There was a 2 week break in between the time each task was taken to ensure that the information learnt in one task does not affect the next. Also, to compensate for the possible affect that the order in which tasks are taken in can have on the results, the order each participant executed the tasks was randomized.

It has been reported that designers' individual approach and their expertise affects the process (Eisentraut 1999). What is suggested here is that each designer would read and interpret the requirements of the task differently. This calls for a mechanism through which an understanding of the mental model the designer creates from the briefing information is obtained and later used as a reference to analyse the distribution of *knowledge* in the duration of actual design activity. Therefore each task was comprised of the following three phases:

1. **Briefing phase:** Participants were asked to watch a short three minutes video clip which display a walk through in the 3D model of the site and explained the brief to them. They were asked to listen and watch carefully. They were then given one minute to recite as much of the information that they had managed to memorize or pick up from the contextual information.
2. **Interpretation phase:** Here participants were given the written format of the brief with a site plan and three images of the site. They were required to read the brief aloud to provide them a chance to revise the requirements. Repeating the briefing information for a second time is used in similar studies which required participants to enter the design phase without access to the brief (Bilda et al. 2006). According to the Modal model of memory, out of the receive information through our sensory registers, only information which is repeated is processed by memory (short term/long term) otherwise it is forgotten (Gazzaniga et al. 2014). The participants were also asked to talk about which information is important to them and why. This enabled an understanding of their rationale in interpreting and analysing, which would make judging the process easier. This phase lasted about 5 min.

   Together these two phase enable the participant to construct a mental model which withholds the primary *knowledge* they will take onto to the design phase.
3. **Design phase:** In this phase, all the documents except for the plan were taken away from the participants. They were given 40 min to develop a concept only using the tool provided and on the 3D model of the site pre constructed for them. They were specifically asked to plan their time so no process lasts any less than 30 min to ensure fairness of comparison. The removal of the written brief and other documents allows the information that is inputted into the process to be controlled.

The Tasks

The tasks asked for a small cultural center on a square shaped site in close proximity to a river and boxed in by east and west neighbors. The freehand sketching process was located in London and asked for the design of an exhibition hall. The Sketch Up process was located in Nottingham and asked for a performing arts center. The Rhinoceros 3D task was located in Leeds and required the design of a public atrium and reading hall. Figure 2 illustrates the 3D models of each site.

All tasks were located on sites in the UK, where the participants had more cultural familiarity with. This had proved to be an important factor in a similar pilot study conducted previously, where the task was set in Vietnam and the participants struggled because of the lack of precedent knowledge.

The tasks were designed based on design issues the researchers had known students at the University of Nottingham to have had practice in. Therefore all tasks were centered on incorporating maximum diffused daylight. Each task required one main central space which would cover the entire ground floor of the site and had to be the most public place of the building. Other complementary smaller spaces that were more private were also needed. A few of these spaces did not need any natural daylighting, such as film and documentary show room for the Rhinoceros 3D task for example. All tasks had the same height restriction. Therefore participants were encouraged to locate the required functions in a creative way that allows each space to receive its required light while maintaining a meaningful narrative in how the building denotes its overall function.

Figure 3 presents an example of the site plans given to the participants. Notably to ensure that the information that can affect external *knowledge* is similar across the tasks, participants were provided with plans that had been stripped down to some essential information, presented in the same way. This was also the case for the 3D models of the sites. All the architectural elements in the site such as windows had the same details across the tasks.

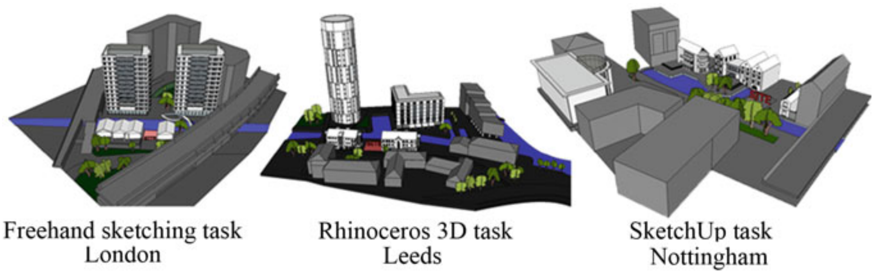


Fig. 2 The 3D models of the three tasks

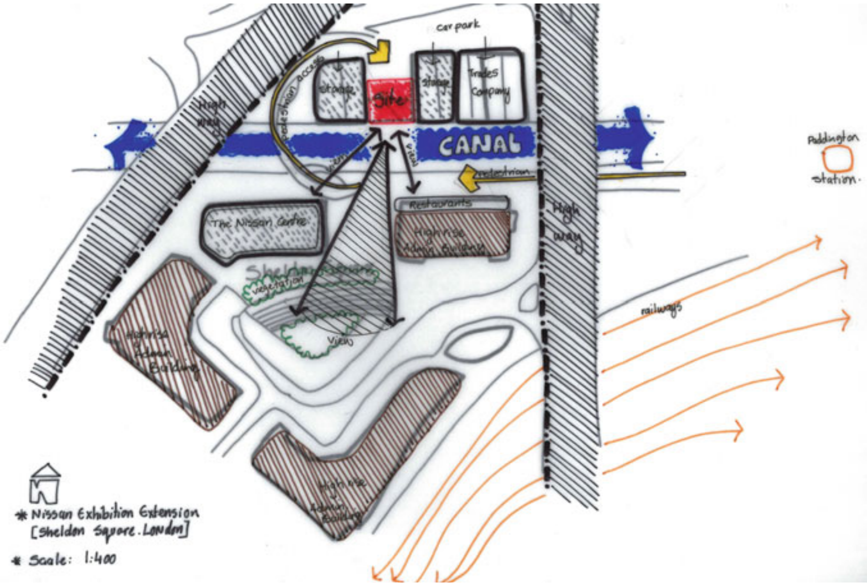


Fig. 3 Site plan of the freehand sketching task. Plan redrawn by researchers to ensure equality of information across tasks

Participant Recruiting

A call for participants was sent to final year students in the Diploma Architecture Programme at the University of Nottingham with experience in using the three tools and confidence in developing a concept in those mediums. Out of the volunteers, the suitability of candidates was conferred with two lecturers who knew each student's design skill level. All the students had undergone their undergraduate degree at Nottingham and so came from the same education background. The shortlisted candidates were then invited to an introductory workshop on how to 'think aloud', they were given short 5 min tasks to accomplish using the software Rhinoceros 3D to practice thinking aloud.

Why the Three Tools: Pencil on Paper, Sketch up and Rhinoceros 3D

In this study, the researchers chose to compare designing with pencil on paper, Sketch Up and Rhinoceros 3D. The latter two are 3D modelling software, which architecture students at the University of Nottingham often use to assist them in developing initial design concepts as an alternative to freehand sketching. These three tools were mainly chosen because of their fundamental differences in how they allow lines and shapes to be manipulated. The speculation is that each method of manipulation will conjure a different mind-set and potential for intervention and this will be the base cause of difference in the distribution of *knowledge* across the cognitive system.

For example when a designer freehand sketches on paper they inscribe lines into the material surface of the paper. That line is a true, scaled representation of the object it intends to be and its properties remain constant. If they choose to omit the line and rub it out, somehow the indentation will leave its trace behind. With another tool such as Sketch Up, lines are more like narrow sticky tapes. They can be picked up and moved around and if one line is drawn over the other on the same plane, it will dissect the first line into pieces; basically it will stick to it. A continuum of lines leading to a closed shape on one plane will always lead to the appearance of an opaque surface, suddenly making that shape more apparent than any other drawing in the medium. In Rhinoceros 3D however, lines are more like rubber band, they have topological properties and can be stretched, pulled and freely manipulated.

Analysis

The Coding Scheme

Table 1 presents the coding scheme developed for this research. This scheme is *knowledge* based and process oriented. It describes *knowledge* as a series of rules applied to the three processes in Asimov's model of 1962, "Analysis", "Synthesis" and "Evaluation". In the proposed scheme, both verbalizations and the design actions are expressed by rule categories. The designer's verbalization is mainly used to understand their rationale and approach, therefore it is seen as a reflection of their "Analysis" processes, which as extensively explained earlier on can be further divided into internal or external rules.

Coyne et al. (1987) discuss that a design system consists of five sets of components: "Designs", "Vocabulary of Elements", "Knowledge", "Interpretations" and "Contextual Information". A synthesis of these components can generate new *knowledge.* Newly generated *knowledge* needs to be evaluated for its fit in the design. As a result this may induce structural changes in the relationship of other components, which therefore encourages an analysis of the overall structure. In this light Coyne et al. (1987) indicate that the design system's potential for creativity can be strongly connected to moments of induction of *knowledge.*

Respectively, in this scheme, the role of the synthesizer is relegated to the design tool in use, in that the tool enables *knowledge* to be expressed through the tool's specific *vocabulary of elements* and via the designers' drawing actions. "Evaluation" processes can also be executed through the medium of the tool.

Table 1 A knowledge based, process oriented coding scheme

Category	Name	Description	
Analysis	Primary internal rule (IRp_x)	Rules created directly from reading the design brief for the first time, without any drawing taken out	
	Generated internal rule (IRg_x)	Rules created as a result of evaluation	
	Modified internal rule (IRm_x)	Rules from the previous design instances modified as a result of evaluation	
	Primary expected external rule (ERp_x)	Implicit Rules in the design brief, driven from the primary internal rules (rules other than the internal rules which justify the designers proceeding actions)	
	Generated expected external rule (ERg_x)	Implicit Rules created as a result of evaluation	
	Modified expected external rule (ERm_x)	Implicit rules from the previous design instance modified as a result of evaluation	
Synthesis	Applied structural external rule (ERs_x)	Drawing actions leaving a mark on the representational medium as rules	ERs_1-closed lines
			ERs_2-open ended lines
			ERs_3-appearing lines
			ERs_4-continuous closed lines
			ERs_5-continuous open ended lines
	Potential structural external rule	Drawing action or bodily actions (eye movement/hand movement) not leaving a mark or actions executed mentally	
Evaluation	Transformed image evaluation ($\mathrm{ERs'}_1$)	When no new rule is added to the task space but an action is executed. These are evaluation actions occurring [immediately] after an element has been depicted and do not involve the drawing of lines	
	Primary image evaluation ($\mathrm{ERs'}_2$)	When no new rule is added to the task space but a drawing action is executed on the current configuration (design) and usually happens concurrently with the act of drawing an element	
	Task space evaluation ($\mathrm{ERs'}_3$)	When new rules are introduced to the task space or old rules refined as a result of transforming actions on the entire configuration (design)	

An example of which can be rotating a design artefact while thinking. Both “Synthesis” and “Evaluation” actions of such can be regarded as external in that they are readily available and do not require memorization and also structural as they contribute to the generation and modification of *knowledge*.

The "Synthesis" actions are also further divided based on drawing actions. The intention was to base the division on a language common to drawing regardless of the design tool used. Therefore the most primitive token by which all design action can be describes was chosen: *lines* (Arnheim 1977; Mitchell 1990). The different line type categories in this scheme denote an understanding of the degree of certainty the designer displays while doing that act. For example when a designer is certain about an idea, this is almost always illustrated by a crisp line with two definite end points.

The subdivision of the "Evaluation" actions is temporal. It reflects the scale of evaluation by revealing whether that action occurred concurrently to a drawing action or after. It is also notable that some "Synthesis" actions, namely those conducted in the mind are not codable using this scheme, as analysis must be based on what is observable and represented through verbalization or actual physical activity.

Method of Analysis

For every recorded process a code reference list based on the first two phases of the experiment was made. A partial example of the code reference list for one participant can be seen in Fig. 4. This is then used by the coders to interpret which segments embrace which rules. Table 2 shows two coded segments with the code reference list from Fig. 4, where the coding of rules IRg_2 and ERg_2 can be seen in action. Where ever a rule is introduced for the first time, it has been indicated by a bold font. In segments where there is more than one rule used they are placed in the order they occur from left to right separated by the > sign.

For the comparison of *knowledge* distribution, the variable was the number of segments each rule is distanced from the beginning of the design stage. The means and standard deviations of the distributions for each participant's tasks were calculated regardless of the rule type, for all three task spaces: internal, external and the aggregated. However since segments were not of equal time length across participants, the data necessary for analysis (the mean and standard deviations) was re-expressed in terms of the percentage of time passed when they occurred in their process. The aim was to underpin possible significant difference in the mean *knowledge* distribution based on the design tool used. Therefore a one-way ANOVA (Analysis of Variance) test using IBM SPSS 21 was utilized here.

To not violate the assumptions of the ANOVA test, gaining assurance of the normality of the data was required. The Shapiro–Wilk test was used for this purpose. Also the homogeneity of variance of the means had to be checked for which Levene's test (Spiegel and Stephens 2008) was used. Where the ANOVA indicates a significant difference at a level of $\alpha = 0.05$, this was followed up by a pairwise comparison applied through the LSD (Least Significant Difference) post hoc test to determine where within the data the difference lies.

Rule number	Primary Internal Rule [IRpx]	*dependency-IRpx*	Generated Internal Rule [IRgx]	*dependency-IRgx*	Modified Internal Rule [IRmx]	*dependency-IRmx*
1			stepping façade (bottom to top)	ERp10+ERp15	integration of stairs in north façade	IRg4+ERp16+ERg4
2			atrium as plaza-continuum of open public space	IRp3+IRp37+IRp44+ERp15		

Rule number	Primary External Rule [ERpx]	*dependency-ERpx*	Generated External Rule [ERgx]	*dependency-ERgx*	Modified External Rule [ERmx]	*dependency-ERmx*
1	Neighbour 1-a mix of high and low rise commercial building	–	thresholding	IRp15+IRg1	use of structure as framing object	ERp16+ERg1
2	Neighbour 2-railways	–	atrium as central feature	IRp37+IRg3	atrium as an open habitual space	IRg2+ERg2

Fig. 4 An example of part of the code reference list for one participant

Analysis was conducted at these levels: aggregated task space, Internal task space, External task space and External task space (new rules only).

It is also notable that due to a small study group (15 experiments) the intention was not to generalize findings from this study. The ANOVA was utilized only to show that a difference in the means of distribution, is a difference that is considerable and worth further investigation.

Results

Figure 5 displays the mean *knowledge* distribution of the 15 processes at the level of the aggregated task space, where the error bars represent the standard deviation.

On average, in all processes regardless of the tool used or the participants' individual approaches, the mean *knowledge* distribution occurs just under half way

Table 2 An example of two coded segments

Verbal utterance	Rule code (internal)	Rule code (external)	Rule code (structural external)	Segment distance from start of process
Or maybe or maybe it could become a multi-functional space [*moves cursor to explain*]	_	Erg_1	$\mathrm{ERs'}_3$	18
And erm, then I need to create this vertical link between the atrium and library, and er,[*moves cursor as if drawing, zooming in and out*] let me just, because it's all square and erm, and we have this solid wall, I will place them, [*goes to plan view port and moves cursor to explain*] the stairs in the, sort of middle, but not exactly central but the, in the middle of this solid wall. Just see what happens, it's just kind of idea of, I guess right now	$\mathbf{IRg_2} > \mathbf{IRp_{34}}$	$\mathrm{ERp}_6 > \mathbf{ERg_2}$	$\mathrm{ERs'}_3 > \mathrm{ERs'}_3$	19

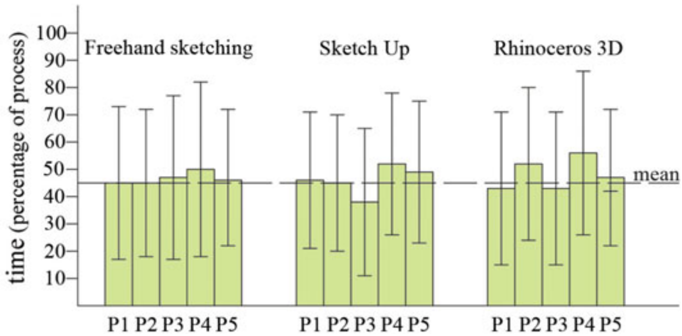


Fig. 5 Mean *knowledge* distribution and standard deviation of design processes

within the process, at 45%. Each process on average also held a standard deviation of 28%. This fairly wide clustering of *knowledge* around the mean indicates that participants are cognitively active throughout the process. A closer look at the distributions showed that all throughout the process, the participants managed to generate new *knowledge*. Therefore it was not just the case that the creation of new *knowledge* were dedicated to a specific interval within the process. This alternation between reuse and generation of *knowledge* would thereby suggest that an executed action through the use of the design tool has a contributing role.

Based on the Shapiro–Wilk test, processes conducted with each tool represent a normal population (null hypothesis rejected since *p value* $> \alpha = 0.05$ for each tool. The *p values are*; freehand sketching:0.71, SketchUp:0.80, Rhinoceros 3D:0.37). The 15 processes also indicate a homogenous population using Levene's test (null hypothesis rejected with a *p value* of 0.34). Having passed these two tests, the ANOVA can be applied.

Studying Fig. 5, the effect of the participants' individual differences is also more apparent in tasks executed using the 3D modelling software, whereas in the freehand sketching process their performance in terms of *knowledge* distribution is very similar. However the ANOVA shows the differences at this level of analysis, to be insignificance (null hypothesis accepted with a *p value* of 0.75). Therefore the tool has no significant effect on the overall (aggregated) cognitive process.

However it was still of interest to this study to understand why the participants' performance in freehand sketching were more similar and consistent compared to the other two. Therefore, three categories of their design actions were further observed; ERs_1-the drawing of closed lines, which displays certainty in the participants' decision making; ERs'_2- evaluation concurrent to a drawing action, which displays constant consideration of small scale spatial relation; ERs'_3-evaluation of the overall task space, which represents consideration of fitness of ideas with the requirements and large scale spatial relation. Figure 6, shows that all participants, were noticeably evaluating their concurrent actions more in the free hand sketching process. They were also excessively engaged with drawing actions using this tool. In the case of the 3D modelling tools, there is proportionally more evaluation of the task space compared to the amount of lines drawn.

Based on the observation of the processes, in the freehand sketching process, the participants are more dependent on the tool to inform their external *knowledge*. They break the problem into smaller chunks and explore small scale options by excessive drawing (rapidly re-drawing over an already depicted shape) without simultaneously being concerned about the effects of the actions on the overall configuration. However using the other tools, constant concern about the overall configuration (which is encouraged through the possibility of zooming in and out for example), means that each drawn line plays more of an importance within the overall picture of the design. Therefore the participant only commits to drawing that line when assurance of its necessity is reached. In this sense, as each drawing action using these tools would bear more semantic value, it can also dramatically change the structure of the design process compared to when freehand sketching. As a result more divergence between participants using the 3D modelling software can be expected.

The first part of the study shows that regardless of differences in actions, participants had regulated their processes so that the overall mean *knowledge* distributions were not affected. The second part of the study requires looking at whether

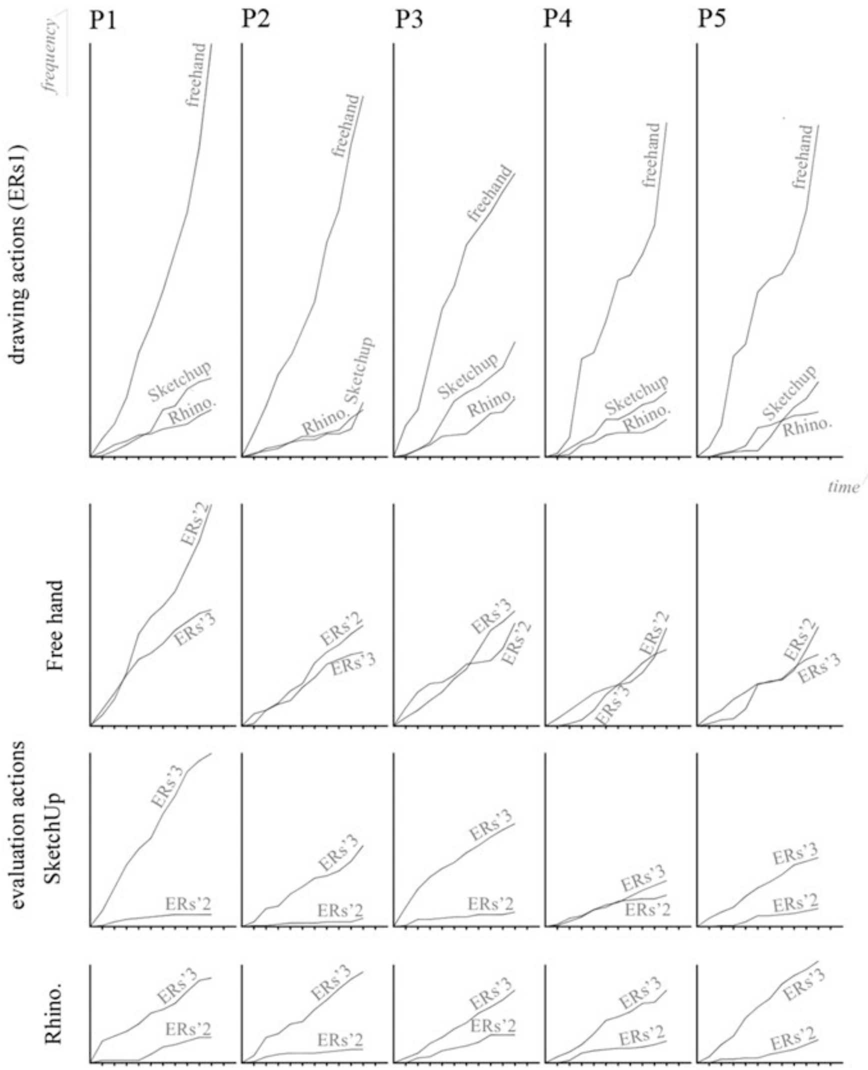


Fig. 6 The participants' frequency of drawing and evaluating actions

the effect of these differences can be picked up when analysing distributions within internal and external task spaces. Figure 7 illustrates the findings. When comparing the mean *knowledge* distribution of all rules in the internal and external task spaces the ANOVA indicated no significance difference (internal task space-*p value* 0.39, external task space-*p value* 0.37) However when we conducted a filtered analysis on the moments new rules were introduced, the differences between the mean *knowledge* distribution across the two task spaces for each process were more dramatic (the hatched areas in Fig. 7).

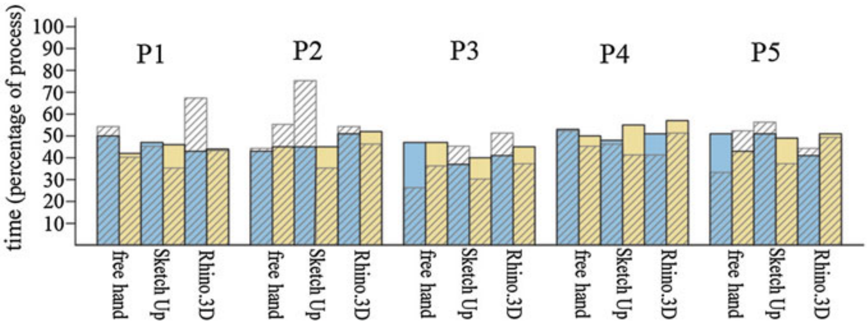


Fig. 7 Comparing mean *knowledge* distribution across internal and external task spaces (*blue* internal task space; *yellow* external task space; *hatched* mean distribution of new rule introduction)

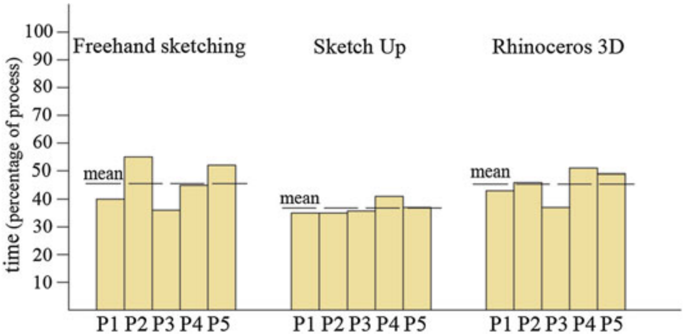


Fig. 8 Comparing differences for the mean *knowledge* distributions of new external rules

In analysing the differences in the distribution of new rules across the task spaces, there was a case of violation of normality for the mean *knowledge* distribution of new internal rules, so we further looked to test the ANOVA for differences in the mean *knowledge* distribution of new external rules only. Here, it was shown that there is a significant difference (*p value* 0.04). The LSD post hoc test, showed this difference to lie within the mean *knowledge* distribution when comparing freehand sketching to Sketch Up, and Rhinoceros 3D to Sketch Up but not in between Rhinoceros 3D and freehand sketching, which is reflected in Fig. 8.

This would imply that in design, the elements external to the task such as the design tool and the environment it creates for design do affect the process. For these participants, the fact that Sketch Up significantly entices external *knowledge* generation (which was discussed to be dependent on the tool in the case of this study) early on in the process, together with the semantics that as described drawing actions bear, means that this tool has had more of an impact on the structuring of the process and how ideas are defined.

Discussion

Based on the results, the question of interest for further discussion is on how design behaviour using Sketch Up in the earlier stages of the process differs from that conducted using the other two tools. Achieving a neat design—that is one where drawings do not lead to frustrating issues, such as surfaces not forming in the same plane or objects becoming integrated into other objects unintentionally—using Sketch Up requires attention and planning. Using this tool—more than the other tools—the designer is encouraged to break down the process based on the affordance of the workspace, because the cost of errors can be quite high. Errors impose hindrance in the creative process and the sequential flow of ideas (Nijstad et al. 2003). Retrieving from a mistake in free hand sketching requires less effort and time and therefore an inappropriate decision would not affect the process as much as it would using Sketch Up.

Therefore as observed in this study, to ensure sound decisions, the participants relied heavily on contextual information made available to them when using Sketch Up. Objects were often created as visual references to help the participant not diverge from the domain which they believed will yield in their desired design concept. An example of this can be seen in participant Jon's process in Fig. 9.

What is postulated in this paper is that how the visual information at each moment is framed also affects the process. In Sketch Up the designer works on one 3D canvas, constantly exposed to all information. In freehand sketching, only so much can be depicted in each perspective drawing or on one canvas. Therefore the designer can assess and read the information more effectively because the load has been broken down. Sometimes it is easier to solve superficial and present issues first before attending to more complex issues. Working on one canvas, holds the designer back from focusing on delving into the deeper issues as there are many trivial issues that can be attended to. There is a tendency in designers to shift between frames or to minimize the frame so that it can help them focus on a small cluster of information at each given time. In this respect Rhinoceros 3D's provision of four view ports is more effective than Sketch Up. Figure 9 shows instances when Jon has zoomed into exclude contextual information in Sketch Up so as to focus on a specific issue.

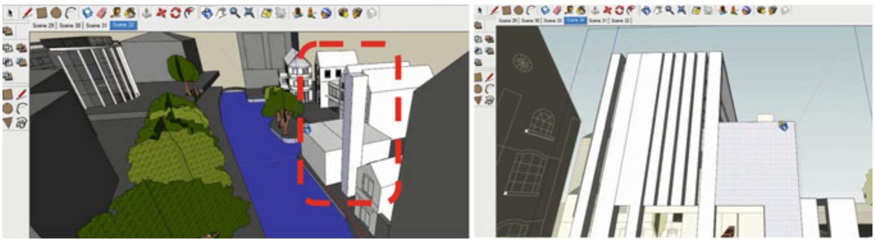


Fig. 9 *Left* An example of creating visual references in Sketch Up. *Right* zooming in, in Sketch Up to reduce presence of contextual information in frame

In the case of this study, interestingly new *knowledge* distribution for both free hand sketching and Rhinoceros 3D process was the same. However as discussed, these tools provide different workspaces and encourage different methods of manipulation. This suggests that different methods of manipulation can be isomorphs of one another, meaning that we do not necessarily require the assistance of intuitive or physically tangible tools such as pen and pencil to aid creative thinking. Therefore it can be concluded that tools have a cognitive role in the design process which goes beyond the physical nature of the tool. The change they apply is in the breaking down of the process and affordance for effectively handling information.

Conclusion

This paper looked at how design processes can be affected by the tools used to execute them. The study was limited to a small sample size. Therefore, the findings are not considered to be truly representative of what happens in design processes. However, it did manage to show the necessity of decomposing cognitive activity into its internal and external components for an effective study of differences in processes under the influence of varying external conditions.

A new coding scheme was also introduced based on the theory of Distributed Cognition that can be used in protocol design studies, where the role of tools is incorporated into the structure of the scheme. The scheme assisted in categorizing the *knowledge* used in the process as a series of rules. Studying the distribution of *knowledge* (rules), the results suggested there to be no significant difference between the tools when processes are compared at level of the aggregated task space. Research in this field, is currently focused on analysis at this level. However a significant difference was in how the tools had affected the introduction of new *knowledge* within the external task space.

The main argument here was that the reason for such outcomes, surpasses differences in the physicality of the tools. The tools afford different controlling qualities in the decision making structure and incorporation of information available to the designer, which affects creativity by determining when and how new *knowledge* is introduced to the process. Acknowledging this conclusion, we can investigate how different methods of manipulation can result in isomorphs for a creative cognitive system.

References

Arnheim R (1977) The dynamics of architectural form: based on the 1975 Mary Duke Biddle lectures at the Cooper Union. University of California Press, London

Baber C, Parekh M, Cengiz TG (2014) Tool use as distributed cognition: how tools help, hinder and define manual skills. Front Psychol 5:1–14

Bilda Z, Gero JS, Purcell T (2006) To sketch or not to sketch? That is the question. Des Stud 27 (5):587–613
Clark A, Chalmers D (1998) The extended mind. Analysis 58(1):7–19
Coyne RD, Gero JS (1986) Semantics and the organization of knowledge in design. Design Comput 1(1):68–89
Coyne RD, Rosenman MA, Radford AD et al (1987) Innovation and creativity in knowledge based CAD. Expert systems in computer aided design, pp 435–465
Dartnall T (2005) Does the world leak into the mind? Active externalism, “internalism” and epistemology. Cognit Sci 29(1):135–143
Dickson J, Mclennan J, Omodei MM (2000) Effects of concurrent verbalization on a time-critical, dynamic decision making task. J Gen Psychol 127(2):217–228
Dinar M, Shah JJ, Langley P, et al (2011) A structure for representing problem formulation in design. In: Impacting society through engineering design DS 68-6: proceedings of the 18th international conference on engineering design, ICED11, vol 6, design information and knowledge, Copenhagen, Denmark
Dorst K, Cross N (2001) Creativity in the design process: co-evolution of problem-solution. Des Stud 22(5):425–437
Edmonds EA (1993) Knowledge-based systems for creativity. Modelling creativity and knowledge-based creative design, pp 259–271
Eisentraut R (1999) Styles of problem solving and their influence on the design process. Des Stud 20(5):431–437
Ericsson KA, Simon HA (1993) Protocol analysis verbal reports as data. MIT Press, Cambridge
Gazzaniga M, Ivry RB, Mangun GR (2014) Cognitive neuroscience: the biology of the mind. W. W. Norton & Company Ltd, New York
Gero JS, Fujii H (2000) A computational framework for concept formation for a situated design agent. Knowl-Based Syst 13(6):361–368
Goel V, Pirolli P (1992) The structure of design problem spaces. Cognit Sci 16(3):395–429
Grafton ST, Fadiga L, Arbib MA et al (1997) Premotor cortex activation during observation and naming of familiar tools. Neuroimage 6(4):231–236
Hollan J, Hutchins E, Kirsh D (2000) Distributed cognition: toward a new foundation for human–computer interaction research. ACM Trans Comput Human Interact 7(2):174–196
Jonson B (2005) Design ideation: the conceptual sketch in the digital age. Des Stud 26(6):613–624
Maher ML, Poon J (1995) Co-evolution of the fitness function and design solution for design exploration. In: IEEE conference on evolutionary computation, vol 1, pp 240
Mitchell WJ (1990) The logic of architecture: design, computation and cognition. MIT Press, Cambridge
Nijstad BA, Stroebe W, Lodewijkx HF (2003) Production blocking and idea generation: does blockage interfere with cognitive processes? J Exp Soc Psychol 39(6):531–548
Noe A (2006) Action in perception. The MIT Press, Cambridge, Mass
Spiegel MR, Stephens LJ (2008) Schaum’s outline of statistics, 4th edn. McGraw-Hill, New York
Suwa M, Gero JS, Purcell T (1998) Macroscopic analysis of design processes based on a scheme for coding designer’s cognitive actions. Des Stud 19(4):455–483
Tang HH, Gero JS (2001) Roles of knowledge while designing and their implications for CAAD. In: Proceedings of CAADRIA, vol 1, pp 81–89
Yu R, Gero JS, Gu N (2013) Impact of using rule algorithms on designers’ behavior in a parametric design environment: preliminary results from a pilot study. Global design and local materialization, pp 13–22
Zhang J, Norman DA (1994) Representations in distributed cognitive tasks. Cognit Sci 18(1): 87–122

Part VII
Design Synthesis

A Heuristic Approach for the Automated Generation of Furniture Layout Schemes in Residential Spaces

Sherif Abdelmohsen, Ayman Assem, Sherif Tarabishy and Ahmed Ibrahim

Abstract A variety of heuristic methods and algorithms have been developed for space layout planning problems. Recent efforts to generate furniture layout schemes in existing spatial configurations have mostly relied on exhaustive search and are likely to produce dysfunctional or counter-intuitive solutions. In this paper, we propose a heuristic approach for the automated generation of furniture layout schemes, with specific focus on residential spaces. First, we present an operational definition for furniture entities, space configurations, and space entities. Then we introduce a heuristic algorithm for generating furniture layout schemes based on a set of space subdivision rules, object-object relations, and object-space relations. Using Grasshopper, we generate a group of possible schemes for a sample residential living space. A discussion follows, outlining current limitations, expanding the context of the study, and possibilities for development.

Introduction

As a subset of space layout planning, furniture layout design involves a continuous process of divergence and convergence of solution space in order to achieve maximum diversity of alternatives along with informed decision making and near optimum solutions. Benchmarking the quality of layout schemes and design alternatives requires a balance between exhausting a plethora of possibilities, and achieving rational optimality.

Furniture layout organization as a task is traditionally similar to the problem of placing an object (with complex geometry) in space (implying topological

S. Abdelmohsen (✉) · S. Tarabishy
The American University in Cairo, Cairo, Egypt
e-mail: sherifmorad@aucegypt.edu

S. Abdelmohsen · A. Assem
Ain Shams University, Cairo, Egypt

S. Abdelmohsen · A. Assem · S. Tarabishy · A. Ibrahim
UNii Engineering Consultancy, Cairo, Egypt

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_25

relations). This problem, according to Flemming et al. (1988), implies an approach that aims at systematically enumerating alternative solutions while simultaneously considering a wide spectrum of criteria; both beyond human cognition. Some of the main challenges in this problem involve the need for efficient search strategies while dealing with the infinite solution space resulting from the inherent growing geometrical complexity (Flemming et al. 1988), and the need to associate with the natural design process of an architect while attempting to address basic topological solutions (Medjdoub and Yannou 2000).

Research addressing the automation of furniture layout schemes in existing spatial configurations stems from two lines of work; (1) facility layout and space layout planning, and (2) furniture layout optimization based on interior design guidelines. In essence, we approach the furniture layout problem as a space planning problem, where space boundary is perceived as a container for subzones that hold furniture arrangements.

A variety of heuristic methods have been developed to address space layout planning problems, including greedy algorithms (Boswell 1992; Ahuja et al. 2000), branch and bound methods (Kim and Kim 1999; Xie and Sahinidis 2008), dynamic programming (Rosenblatt 1986), and single-solution metaheuristic methods such as tabu search (Abdinnour-Helm and Hadley 2000; Chiang and Kouvelis 1996). Some recent efforts to generate furniture layout schemes in existing configurations (Kjølaas 2000; Akazawa et al. 2005; Germer and Schwarz 2009; Larive et al. 2004; Sanchez et al. 2003; Merrell et al. 2011; Yu et al. 2011) have adopted an exhaustive search process that is likely to produce illogical or uninhabitable arrangements. Others focused on merely ergonomic factors. Some others relied on object-object relations without much attention to the analysis of space boundaries, or required manual user intervention.

In this paper, we propose a heuristic approach for the automated generation of furniture layout schemes that involve a thorough analysis of spatial configurations and furniture objects, object-object relations, and space-object relations, to produce a habitable layout scheme. We identify a set of rules for the logical and intuitive placement and arrangement of furniture objects in a given space based on these relations. We claim that the resulting range of possible furniture layout schemes satisfy an intuitive furniture layout process, without the need for an exhaustive search through all possible—and perhaps likely dysfunctional—solutions. Our goal is to emulate how architects would perceive a given spatial configuration, analyze it and propose basic habitable furniture layout schemes that are diverse enough and at the same time meet the existing spatial constraints and conditions.

First, we present an operational definition for furniture entities, and space configurations and entities. Then we introduce a heuristic algorithm for generating furniture layout schemes based on space subdivision rules, object-object relations, and object-space relations. We use Grasshopper to generate furniture layout schemes for a sample residential living space.

Related Work

A variety of approaches have been proposed for space layout planning that apply for our context of study. The graph theory approach is one example, where form emerges from the arrangement of spaces in a planar graph. Another approach involves quadratic assignment problems, where entities are assigned to cells in a matrix representation (Balakrishnan and Cheng 2000; El-Baz 2004). Contrary to space emergence approaches, we focus on methods that employ the generation of spaces or partial spaces within a fixed boundary. Developed originally within the facility planning realm, the slicing tree approach is one of the early methods that attempt to continually subdivide a given space through horizontal and vertical lines to obtain new subzones (Tam 1998; Azadivar and Wang 2000; Al-Hakim 2000; Wu and Appleton 2002; Shayan and Chittilappilly 2004; Honiden 2004; Aiello et al. 2006; Banerjee et al. 2008; Aiello et al. 2012). In this approach, terminal nodes represent spaces and internal nodes represent the vertical and horizontal subdivision of the children nodes.

Following the slicing tree approach which was limited in its configuration geometry, other approaches emerged such as combining the space filling curves (SFC) technique with genetic algorithms, where a continuous path is defined that generates space on a matrix (Buscher et al. 2014; Hu et al. 2007; Islier 1998; Kochhar et al. 1998; Wang et al. 2005). Others used evolutionary techniques to solve the planning of departmental spaces (Dunker et al. 2003), where fixed or flexible spatial blocks are allocated within a certain boundary. Other factors were considered in later approaches like hierarchical organization of layout elements (Koenig and Schneider 2012), calculation of distances between departments through aisles using graph algorithms and genetic algorithms (Lee et al. 2005), and multi-level space allocation using hybrid evolutionary techniques (Rodrigues et al. 2013).

An important factor to consider is how architects attempt to visualize and subdivide spaces to establish possible layout configurations. According to Indraprastha and Shinozaki (2011), space boundaries and inter-relationships between different architectural elements and planned activities constitute an essential component of space composition. In their method, architectural space is composed of subdivided enclosed territorial spaces defined by internal circulation paths, and each of these spaces has a set of distinct physical properties that are impacted by different elements.

Early attempts to generate furniture layouts include the work by Kjølaas (2000), where functional space is represented as a nested hierarchy of rectangular templates that are swapped by eight predetermined mutation functions, while free space is represented using empty boxes in front of doors and windows. This approach is limited however to strictly rectangular configurations. A later approach introduced parent-child relationships between furniture objects and used a semantic database to store these relations but with manual control on inter-object relations and constraints (Akazawa et al. 2005). Germer and Schwarz (2009) introduced an agent-based approach, where each furniture object was viewed as an agent seeking to attach to a parent object. Parent-child relationships of each object had to be manually defined.

Merrell et al. (2011) introduced a furniture layout system that generates proposed arrangements interactively according to a set of developed interior design guidelines, such as balance, alignment, emphasis and conversation. Yu et al. (2011) introduced an automated furniture layout synthesis approach using realistic indoor scenes, whereby they considered human factors including visibility, constraints, accessibility, and pathways. While these approaches focused on ergonomics and studied further aspects of visibility and accessibility, they tend to conduct an exhaustive search for all possible schemes regardless of spatial analysis or space-object relations, and generate solutions that may be counter-intuitive to how architects perceive space and its closely coupled relation with furniture layouts to produce habitable space. Our approach attempts to address this gap, and adopts a heuristic approach to automatically generate basic furniture layout schemes in a given spatial configuration, taking into consideration an in-depth analysis of space entities, furniture entities, and space-object relations.

Approach

In order to study furniture arrangement within a given space, we first identified a number of possible configurations and space boundaries within which furniture groups are to be positioned. Below are some possible families of space boundaries and their potentials and constraints (Fig. 1).

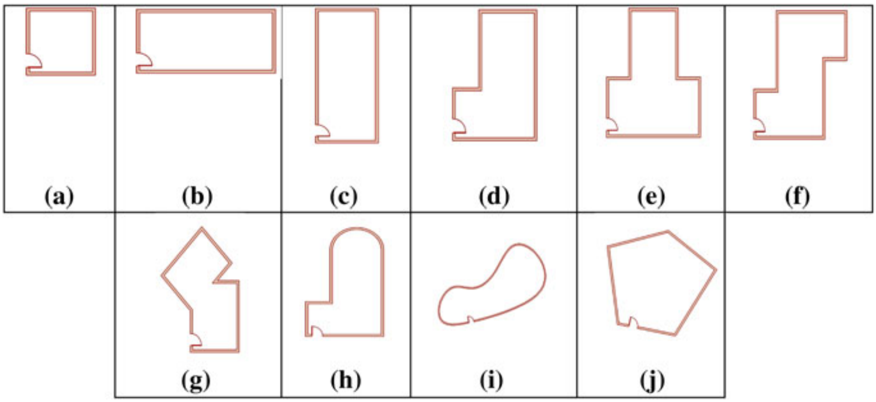


Fig. 1 Possible space boundaries for furniture arrangement: **a**–**c** single rectangular boundary, **d**–**f** clustered rectangular boundary, **g** multi-grid boundary, **h** circular boundary, **i** organic boundary, **j** angled boundary

1. Regular Space Boundary
   (a) Single rectangular boundary: Our assumption is that the space boundary in this configuration is the basic envelope for the bounding box of a single furniture group (e.g. sofa and two armchairs, dining table with chairs, etc.), or multiple groups, depending on space proportion. For example, a $4\ \mathrm{m} \times 4\ \mathrm{m}$ configuration is assumed to host a single furniture group, while a $6\ \mathrm{m} \times 8\ \mathrm{m}$ configuration is assumed to host two or more furniture groups.
   (b) Clustered rectangular boundary: The space boundary in this configuration is assumed to host multiple furniture groups due to the inherent space division into subzones. For example, an L-shaped or a Z-shaped configuration can host two or more groups depending on space proportion.
   (c) Multi-grid boundary: The space boundary in this configuration is assumed to host multiple furniture groups with different alignments on its walls that take different angles, provided clear paths for circulation in between the furniture groups.
   (d) Circular boundary: The space boundary in this configuration allows for furniture groups with different alignment situations, where circular walls constrain the placement of furniture objects.
2. Irregular Space Boundary
   (a) Organic boundary: In this configuration, the space boundary is more constraining in terms of furniture layout possibilities, but defines an implicit subdivision of space based on organic virtual walls that define possible subzones for furniture placement.
   (b) Angled Boundary: The space boundary in this configuration, which is characterized by its sharp edges and angled walls, is assumed to host multiple furniture groups with constrained possibilities related to alignment and spatial relationships.

In these configurations, two main factors affect the possibilities of furniture arrangement: the feasibility of hosting single or grouped furniture elements in space based on physical dimensions, and the divergent/convergent approach exercised by the architect to generate rational arrangement alternatives within the given spatial conditions.

Our basic assumption for these spatial configurations involves single-level spaces with one or two entry points, windows and columns that are either embedded inside the walls or protruding within the space so as to implicitly define subzones. However, there exist other configurations with special elements that have direct implications for furniture arrangement, such as spaces with free-standing columns, multi-level spaces, and filleted or chamfered spaces, as illustrated in Fig. 2.

In these configurations, the column, the difference in level, and the chamfered wall imply virtual walls that either introduce subzones as in the case of the column and multi-level space, or introduce a virtual corner point for the space boundary. These have consequences on the logic of furniture arrangement within the newly

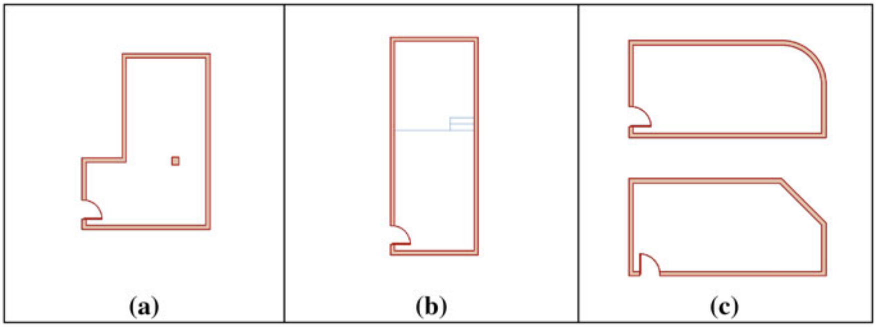


Fig. 2 Spatial configurations with implications for furniture arrangement: **a** space with free-standing column. **b** multi-level space, **c** chamfered space

perceived space. For the scope of this paper, we focus on single and clustered rectangular boundaries. In order to study possible furniture arrangements within a given space, we first provide operational definitions for three main components of the furniture-space setting: (1) furniture entities, (2) space entities, and (3) object-space relations. Then we define rules for each component that constitute the main algorithm for the proposed automated furniture layout system.

Furniture Entities

As shown in Table 1, we describe some of the main furniture entities. We classify elements of furniture in a given space into *Furniture Objects*, such as TV, chair, sofa, table, etc, and *Furniture Groups*, such as seating groups (including sofa, two armchairs, and table), or dining groups (including dining table and six chairs). Some of these objects and groups necessarily require clear orientation logic (*Front* and *Back*) for wall alignment purposes, such as sofas and chairs, while others do not, such as tables. We explicitly embed this logic for the purpose of our system, as this affects how furniture is placed within a given space and in relation to other elements.

We identify a *Furniture Base Point* by which the furniture object or group is spatially allocated. For each, we define a *Furniture Internal Boundary* and *Furniture External Boundary*. The internal boundary specifies the exact bounding box enclosing the object or group, and the external boundary includes circulation and use space (Fig. 3). This becomes significant in addressing circulation issues within a given space. We also define an *Entry Point* for furniture groups, as this defines a likely access point and has implications on circulation patterns within the space. For each internal boundary, we identify a *Corner* and *Center* for the purpose of alignment of furniture with a given wall or virtual wall.

Table 1 Furniture entity definitions

Definition	Description
Furniture object	A single furniture element (e.g. chair, table, TV, etc.)
Furniture group	An arrangement that includes a number of furniture objects (e.g. dining table with chairs, sofa with table and armchairs, etc.)
Furniture base point	A reference point for spatial allocation of the furniture object or group
Furniture internal boundary	A bounding box that encloses a furniture object or group
Furniture external boundary	A bounding box that includes a furniture object or group, and their corresponding circulation and use space
Center of furniture internal boundary	Center of gravity of bounding box of furniture object or group
Corner of furniture internal boundary	Corner of internal bounding box of furniture object or group
Corner of furniture external boundary	Corner of external bounding box of furniture object or group
Entry point of furniture internal boundary	A point that defines access from circulation space to a furniture object or group
Front of furniture external boundary	A line segment on the external bounding box that defines a non wall-aligning face of a furniture object or group, i.e. that faces another object or group in the space
Back of furniture external boundary	A line segment on the external bounding box that defines a wall-aligning or virtual wall-aligning face of a furniture object or group

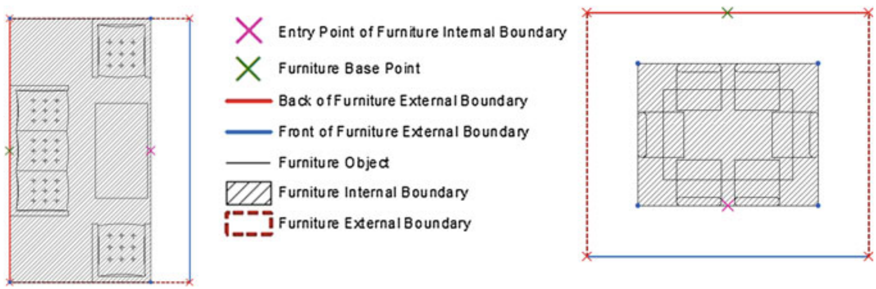


Fig. 3 Furniture group with distinguishable front and back external boundaries (*left*) and non-distinguishable front and back external boundaries (*right*)

After defining the basic furniture entities, we implement three categories of rules for furniture entities: (1) object-object rules, (2) object-group rules, and (3) group-group rules. For the object-object rules, we implement rules for distance constraints based on interior design standards data. For example, we define a range of 2–3 m as an allowed viewing distance between a sofa and a TV, measured from the *Center of Furniture Internal Boundary* for both objects.

For an object-group relation, such as seating group and TV, we assign the distance constraints to the median of the distances between each object of the seating group and the TV. The priority of sequencing of furniture objects and groups is significant in this case, as the seating group or sofa follows the TV object space allocation.

For group-group relations, we define distance/viewing constraints in addition to alignment and attachment constraints. For example, in a seating-dining group relation, we define alignment conditions (which can allow for alignment from Corner, Center, Front or Back of each furniture external boundary to allow for through circulation).

These rules define interrelationships between furniture objects and groups, yet not in context. In the next section, we analyze the space boundary and enclosure, and the logic of furniture allocation within that boundary.

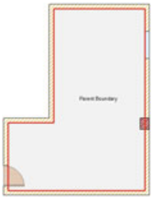
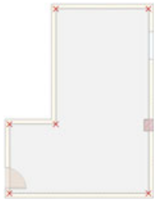
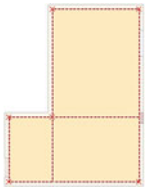
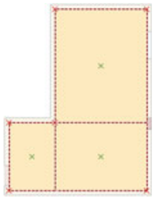
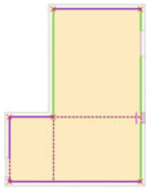
Space Entities

For our space analysis, we use the clustered rectangular configuration (an L-shaped space organization) as an example. To implement the logic of furniture allocation, we differentiate between two main entities: space boundary (or *Parent Boundary*) and *Wall Segment*. For an architect attempting to allocate furniture within space, we assume that two simultaneous processes take place; (1) *space decomposition*: a process of perceiving space and its boundaries and subdividing it visually into potential subzones for single or multiple furniture arrangements (depending on space proportion, area and specific layout configuration), and (2) *wall decomposition*: a process of aligning furniture objects or groups onto wall segments (rather than just full walls), which are physically divided by columns and openings, in addition to virtual walls, through basic geometric subdivisions.

Rather than an exhaustive approach of arbitrarily placing furniture objects in all possible points in a given area or aligned on all possible points on the wall, we constrain the furniture allocation possibilities to logical attempts by architects during the process of perceiving any configuration.

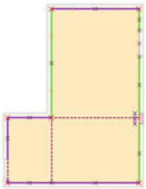
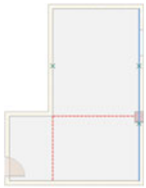
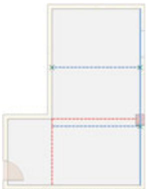
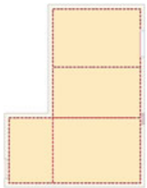
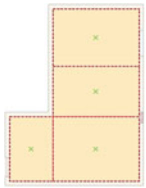
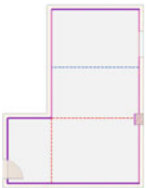
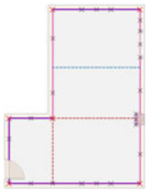
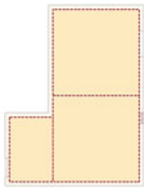
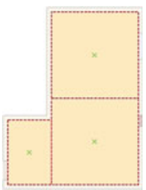
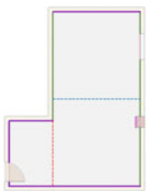
The logic of subdividing and decomposing space depends typically on two factors: the area/proportion of the space, and furniture size. As mentioned earlier, we take into consideration two main furniture components; objects and groups. For furniture groups, we assume that the minimum acceptable area of a group is $2.5 \text{ m} \times 2.5 \text{ m}$, measured from its external boundary to account for circulation and use. This becomes the basic unit upon which the space subdivision logic is based. To carry out the subdivision, we use virtual walls from key points on the parent space boundary (such as corner points, midpoints, one-third, and so on). The number of subdivisions, *child boundaries* and potential furniture allocation points is directly proportional to space area, where the larger the area, the more the virtual walls, and the higher the division level (DL), where DL1 is a subdivision of the first level with one virtual wall and two subzones, DL2 is a subdivision of the second level with two virtual walls and three subzones, and so on.

Table 2 Space entity definitions

Parent boundary The full boundary of a given space, excluding any openings or protruding columns	**Parent boundary corner points** Corner points on the space parent boundary	**Virtual wall from corner** Virtual wall from parent boundary points that subdivides a given space into subzones
		
Parent boundary segment for virtual walls Line segment on parent boundary resulting from space subdivision by a virtual wall	**Child boundary DL1** The full boundary of a subzone resulting from space subdivision by a virtual wall of first division level	**Child boundary points DL1** Points on the space child boundary resulting from space subdivision by a virtual wall of first division level
		
Virtual Wall from parent boundary points DL1 Virtual wall that subdivides space into two subzones	**Child boundary center DL1** The center of gravity of a child boundary subzone	**Wall segments DL1** Wall segments resulting from space subdivision by a virtual wall of first division level, in addition to subdivision by a column or window or door opening
		
Wall segment points DL1 Points on wall resulting from subdivision level 1 (dividing wall into two wall segments)	**Parent boundary segments DL2** Line segment on parent boundary resulting from space subdivision level 2	**Child boundary points DL2** Points on the space child boundary resulting from space subdivision by a virtual wall of second division level

(continued)

Table 2 (continued)

		
Virtual wall from parent boundary points DL2 Virtual wall that subdivides space into three subzones	**Child boundary DL2_1** The full boundary of a subzone resulting from space subdivision by virtual wall of second division level (alternative 1)	**Child boundary center DL2_1** The center of gravity of a child boundary subzone resulting from space subdivision by a virtual wall of second division level (alternative 1)
		
Wall segments DL2_1 Wall segments resulting from space subdivision by a virtual wall of second division level, and subdivision by a column or window or door opening (alternative 1)	**Wall segment points DL2_1** Points on wall resulting from subdivision level 2 (alternative 1)	**Child boundary DL2_2** The full boundary polyline of a subzone resulting from space subdivision by a virtual wall of second division level (alternative 2)
		
Child boundary center DL2_2 The center of gravity of a child boundary subzone resulting from space subdivision by a virtual wall of second division level (alternative 2)	**Wall segments DL2_2** Wall segments resulting from space subdivision by a virtual wall of second division level, in addition to subdivision by column, window or door opening (alternative 2)	**Wall segment points DL2_2** Points on wall resulting from subdivision level 2 (alternative 2)
		

We define below our space decomposition rules:

- If the length of the largest parent boundary line (the largest segment in space to determine subdivision) is greater than 5 m (to accommodate two furniture groups of minimum dimensions), draw virtual wall from the midpoint of the parent boundary line. If length is greater than 8 m, divide into three subzones with two virtual walls. If length is less than 5 m, no subdivision should be done (available space is sufficient for just one furniture group).
- In this layout configuration, virtual walls can also be drawn from parent boundary corners. Subdivision from corners is only allowed if the resulting subzones allow for a furniture group minimum dimension. Therefore, only if the vertical length of the shorter parent boundary line is greater than 2.5 m, a horizontal virtual wall can be drawn to introduce a new subzone, and vice versa.
- If DL1 or DL2 virtual walls are within near proximity (50 cm) to virtual walls from a corner, the virtual wall from corner precedes. The corner tends to define space and subzones physically more than visual subdivisions from the space boundary. If the distance however is from 50 cm to 2.50 m, both alternatives are considered.

Upon subdivision and the definition of child boundaries and new points on these boundaries, the virtual walls establish a new set of wall segments. The following rules are then applied on these segments:

- For wall segments larger than 0.25 m and less than 8 m, divide segment by 2. For segments between 3 m and 10 m, divide by 3, and for segments between 4 m and 12 m, divide by 4. A 6 m wall segment for example would contain furniture allocation placeholders at 3 points resulting from subdividing it into four segments.
- For columns and window openings, their midpoints and endpoints should be added to the list of potential points for allocation.
- If the virtual wall happens to hit the middle of the window, the system should allow for a subdivision after or before the window and at the midpoint as well to accommodate for visual placement of furniture groups or objects in relation to the window segment.

Table 2 describes the different space entities and the subdivision process of boundaries and walls to identify potential points for furniture allocation.

Object-Space Relations

We describe in this section rules for two basic relations between the space and furniture objects or groups: (1) alignment and (2) circulation. Regarding alignment, we distinguish between three types of furniture arrangements: wall-aligned furniture, free-standing furniture, or furniture that accepts both arrangements. The nature of how these furniture objects or groups align with the resulting child space boundaries differs according to these arrangements. If the furniture object or groups is free-standing, it can align with the child boundary in one of four possible

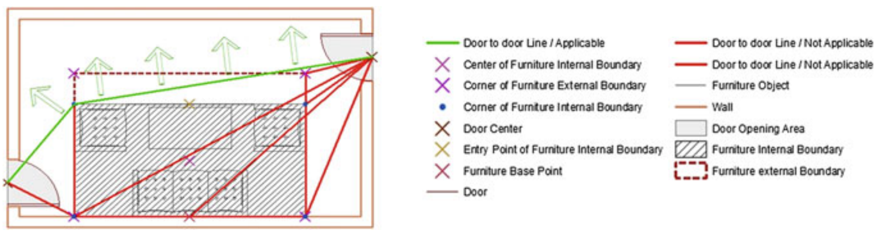


Fig. 4 Circulation rule for an object-space relationship

scenarios: (a) center of gravity of furniture to center of gravity of child boundary, (b) furniture boundary corner to child boundary corner, (c) furniture boundary base point to child boundary point, and (d) furniture boundary base point to nearest wall segment point DL1, DL2, DL3, etc. Wall-aligned furniture objects and groups can only align based on conditions (c) and (d).

Regarding circulation, we introduce this layer of analysis to narrow down possibilities of furniture allocation, considering how an architect would approach the problem. We put forward that circulation within a habitable space containing furniture objects is considered by the architect in two instances; (1) as a pre-checking mechanism, where basic walkability is determined through circulation area percentages, and (2) as a detailed study of possible circulation paths around furniture arrangements and their entry points. For the second more detailed circulation study, we propose the following rules for a given space with two doors or access points (Fig. 4):

- Identify *Furniture Boundary Internal* edges for furniture objects
- Connect from midpoint of first *Door Opening* to the nearest *Furniture Internal Boundary Corner* of each group.
- For each group or object, if there is direct access from nearest corner to second access point in space, without intersecting any internal boundary within that group or object, then draw a valid circulation edge. Else, search for next corner of the same group or object, and apply again until valid circulation edge is identified.
- The identified circulation edge represents one boundary of the circulation path. To identify a valid circulation path (with at least 80 cm), search for wall segment, column, or other edge (that defines the other boundary of the circulation path). If the distance between both boundary lines is equal to or more than 80 cm, add this path as a valid circulation path. Else, eliminate the alternative that contains this furniture arrangement from the possible solutions.

Case Study

We selected a 39 m^2 living area L-shaped configuration with two doors, one window, and a protruding column, as shown in Fig. 5. We introduced two furniture groups: a seating group (sofa, two armchairs, table), a dining group, including a dining table and six chairs, and a TV object.

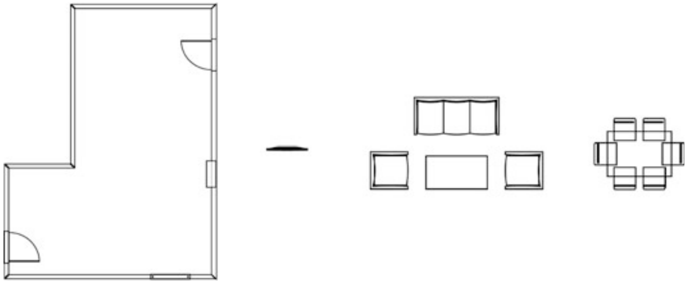


Fig. 5 Case study spatial configuration and introduced furniture groups and objects

We used Grasshopper to define the algorithm for generating possible furniture arrangements, as shown in Fig. 6.

For the algorithm, as illustrated in Fig. 7, we extend lines and draw virtual walls from Parent Boundary Corner Points.

We identify new space subdivisions and child boundaries based on the virtual walls. We draw *Virtual Wall from Parent Boundary Points DL1*, and identify the resulting *Child Boundary DL1*, *Child Boundary Center DL1*, *Wall Segments DL1*, and *Wall Segment Points DL1*. After applying *Virtual Wall from Parent Boundary Points DL2*, we identify the resulting *Child Boundary DL2_1*, *Child Boundary DL2_2, Child Boundary Center DL2_1*, *Child Boundary Center DL2_2, Wall Segments DL2_1*, *Wall Segments DL2_2, Wall Segment Points DL2_1,* and *Wall Segment Points DL2_2*.

After getting all possible furniture allocation points, we apply bounding boxes for *Door Opening Area* entities to exclude them from any possible furniture

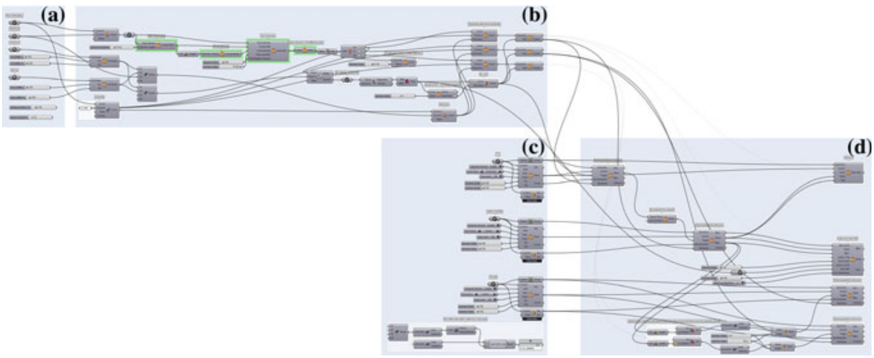


Fig. 6 The Grasshopper definition used to implement the algorithm for generating alternatives for furniture arrangement: **a** defining curves for room boundary, openings buffer, and other distance parameters controlling the placement of furniture groups, **b** extracting virtual walls, creating all possible combinations of subzones, **c** defining the furniture groups that will be used and setting their priority of placement, **d** placing furniture groups in all possible subzones while testing for collisions with room boundary and openings buffer

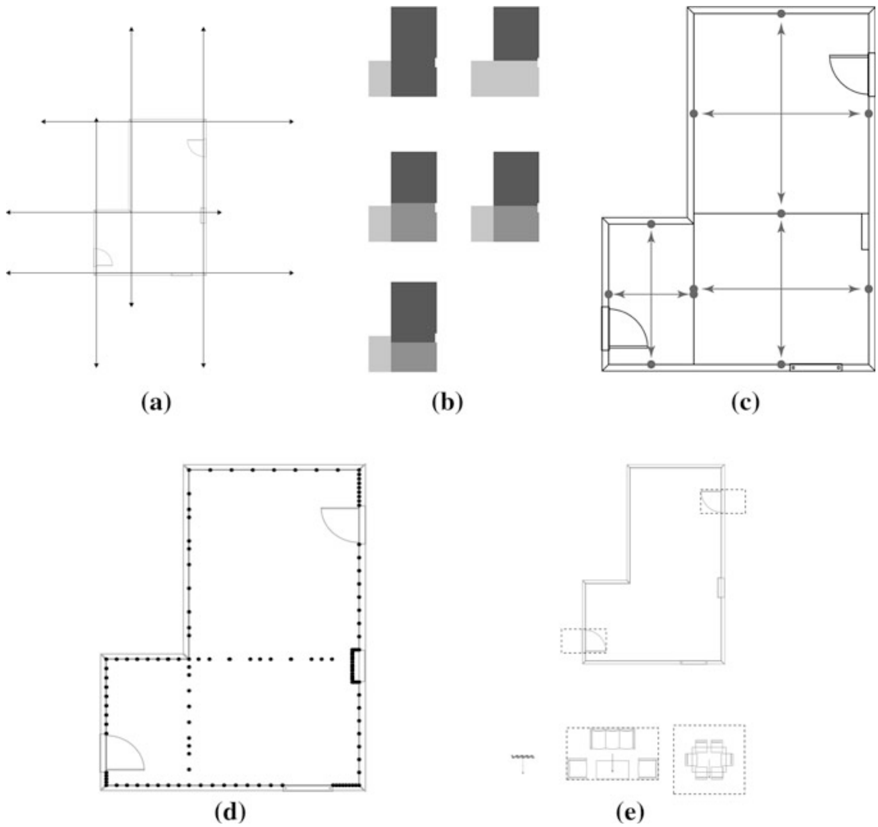


Fig. 7 Algorithm for generating possible furniture arrangements: **a** draw virtual walls from parent boundary corner points, **b** identify child boundaries, **c** identify center points of child boundary and child boundary points DL1, **d** identify resulting child boundaries DL2 and DL3, wall segments DL2 and DL3, and wall segment points DL2, DL3; **e** identify door opening bounding box, apply circulation rules and apply algorithm to dining group, seating group and TV

Fig. 8 Set of possible furniture arrangements generated by the algorithm

allocation. Then we apply the circulation rules to check for valid paths. We run the algorithm, given the object-group relations between the TV and seating group, and the group-group relations between the dining group and seating group, as defined earlier. For this case, 10 possible alternatives for furniture arrangement were generated, as shown in Fig. 8.

Discussion and Future Work

Our research addressed the automated generation of furniture arrangements within a given space, taking a residential living area as a sample with an L-shaped spatial configuration. The logic presented in this paper, which involves space decomposition and wall decomposition, is assumed to respond to the divergent/convergent conversation conducted by architects while attempting to analyze a given space, its boundaries, possible subdivisions and virtual subzones, in addition to walkability scenarios, while taking into consideration basic object-space and object-object relations. This comes in contrast to other previous approaches that computationally exhaust all possible solutions but might fall short of the logical and geometrical iterations that an architect experiences during a space planning or furniture layout process.

Our algorithm applies to mostly regular space configurations, with the exception of circular boundaries, which require—along with irregular space configurations—further analysis of space perception, subdivision logic, and object-space relations. We attempt to address different typologies of space boundaries, as well as different furniture and equipment typologies. As they currently stand, our furniture entities exhibit only a subset of all possible furniture classes (furniture with a fixed position in space such as sofas and dining tables, movable furniture such as chairs, etc.). Other possible classes that we have not addressed include multi-level furniture, dynamic and extendable furniture, furniture with dynamic use, and transformable furniture, only to mention a few. This also applies to equipment and accessories in space. Further investigation of these entities and their object-object and object-space relations will allow for an all-encompassing logic for a comprehensive allocation of any object typologies in space.

We addressed space subdivision, circulation and walkability in space and around furniture objects, according to design guidelines and standards, and based on how architects would simply attempt to perceive a given configuration. More evidence is needed to corroborate this basic assumption. Our ongoing research involves assigning a group of configurations and furniture requirements to experienced architects in the form of a graphical survey, where they are required to allocate furniture objects in those configurations based on intuition and experience. Results from these surveys—in the form of object and space coordinates—will be used to infer statistically the guidelines and rules for our automated furniture generation system.

Another venue to explore is deducing rules and guidelines from patterns of use and behavior in different spatial configurations, perhaps extending the context of study into other settings other than residential units, where the complexity of behavior and space dynamics are expected to highly inform the automated space planning and furniture layout generation.

Our approach in general relied on a consistent dataset resulting from geometric subdivision of spaces and allocating furniture objects based on key points on a spatial configuration. Our next step involves expanding the scope of our input parameters to include complex and inconsistent datasets, including volumetric spatial configurations, visibility, access, behavior patterns, environmental aspects, etc. These dimensions cannot be incorporated into our system as it stands, especially when the objective does not only involve generation of configurations and furniture layout schemes, but also an informed evaluation and optimization of those schemes. We assume that the integration of fuzzy logic would contribute to achieving a more true representation of spatial configurations and complex use scenarios rather than just geometrical relations informed by non context-specific design guidelines.

Acknowledgements We would like to thank UNii Engineering Consultancy for sponsoring and supporting this research.

References

Abdinnour-Helm S, Hadley SW (2000) Tabu search based heuristics for multi-floor facility layout. Int J Prod Res 38:365–383

Ahuja RK, Orlin JB, Tiwari A (2000) A greedy genetic algorithm for the quadratic assignment problem. Comput Oper Res 27:917–934

Aiello G, Enea M, Galante G (2006) A multi-objective approach to facility layout problem by genetic search algorithm and Electre method. Robot Comput-Integr Manuf 22:447–455

Aiello G, La Scalia G, Enea M (2012) A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding. Expert Syst Appl 39:10352–10358

Akazawa Y, Okada Y, Niijima K (2005) Automatic 3D scene generation based on contact constraints. In: Proceedings of conference on computer graphics and artificial intelligence, pp 593–598

Al-Hakim L (2000) On solving facility layout problems using genetic algorithms. Int J Prod Res 38:2573–2582

Azadivar F, Wang J (2000) Facility layout optimization using simulation and genetic algorithms. Int J Prod Res 38:4369–4383

Balakrishnan J, Cheng CH (2000) Genetic search and the dynamic layout problem. Comput Oper Res 27:587–593

Banerjee A, Quiroz JC, Louis SJ (2008) A model of creative design using collaborative interactive genetic algorithms. In: Design computing and cognition'08. Springer, Berlin, pp 397–416

Boswell SG (1992) TESSA: a new greedy heuristic for facilities layout planning. Int J Prod Res 30:1957–1968

Buscher U, Mayer B, Ehrig T (2014) A genetic algorithm for the unequal area facility layout problem. In: Operations research proceedings 2012. Springer, Berlin, pp 109–114

Chiang W-C, Kouvelis P (1996) An improved tabu search heuristic for solving facility layout design problems. Int J Prod Res 34:2565–2585

Dunker T, Radons G, Westkämper E (2003) A coevolutionary algorithm for a facility layout problem. Int J Prod Res 41:3479–3500

El-Baz MA (2004) A genetic algorithm for facility layout 706 problems of different manufacturing environments. Comput Ind Eng 47:233–246

Flemming U, Coyne R, Glavin T, Rychener M (1988) A generative expert system for the design of building layouts—version 2. Technical Report. Carnegie Mellon University, Engineering Design Research Center (EDRC-48-08-88)

Germer T, Schwarz M (2009) Procedural arrangement of furniture for real-time walkthroughs. Comput Graph Forum 28(8):2068–2078

Honiden T (2004) Tree structure modeling and genetic algorithm-based approach to unequal-area facility layout problem. Ind Eng Manag Syst 3:123–128

Hu MH, Ku M-Y, Chen C-C (2007) A genetic algorithm for optimizing facility layout in a wafer fab. In: IMECS, pp 2026–2031

Indraprastha A, Shinozaki M (2011) Elaboration model for mapping architectural space. J Asian Archit Build Eng 10(2):351–358

Islier A (1998) A genetic algorithm approach for multiple criteria facility layout design. Int J Prod Res 36:1549–1569

Kim J, Kim Y (1999) A branch and bound algorithm for locating input and output points of departments on the block layout. J Oper Res Soc 50:517–525

Kjølaas KAH (2000) Automatic furniture population of large architectural models. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Cambridge, MA

Kochhar JS, Foster BT, Heragu SS (1998) HOPE: a genetic algorithm for the unequal area facility layout problem. Comput Oper Res 25:583–594

Koenig R, Schneider S (2012) Hierarchical structuring of layout problems in an interactive evolutionary layout system. Artif Intell Eng Des Anal Manuf 26:129–142

Larive M, Roux OL, Gaildrat V (2004) Using meta-heuristics for constraint-based 3D objects layout. In: Proceedings of conference on computer graphics and artificial intelligence, pp 11–23

Lee K-Y, Roh M-I, Jeong H-S (2005) An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. Comput Oper Res 32:879–899

Medjdoub B, Yannou B (2000) Separating topology and geometry in space planning. Comput Aided Des 32(1):39–61

Merrell P, Schkufza E, Li Z, Agrawala M, Koltun V (2011) Interactive furniture layout using interior design guidelines. ACM Trans Graph 30(4): 87:1–87:9

Rodrigues E, Gaspar AR, Gomes Á (2013) An approach to the multi-level space allocation problem in architecture using a hybrid evolutionary technique. Autom Constr 35:482–498

Rosenblatt MJ (1986) The dynamics of plant layout. Manag Sci 32:76–86

Sanchez S, Roux O, Luga H, Gaildrat V (2003) Constraint-based 3D-object layout using a genetic algorithm. In: Proceedings of conference on computer graphics and artificial intelligence

Shayan E, Chittilappilly A (2004) Genetic algorithm for facilities layout problems based on slicing tree structure. Int J Prod Res 42:4055–4067

Tam K (1998) Solving facility layout problems with geometric constraints using parallel genetic algorithms: experimentation and findings. Int J Prod Res 36:3253–3272

Wang M-J, Hu MH, Ku M-Y (2005) A solution to the unequal area facilities layout problem by genetic algorithm. Comput Ind 56:207–220

Wu Y, Appleton E (2002) The optimisation of block layout and aisle structure by a genetic algorithm. Comput Ind Eng 41:371–387

Xie W, Sahinidis NV (2008) A branch-and-bound algorithm for the continuous facility layout problem. Comput Chem Eng 32:1016–1028

Yu L-F, Yeung S-K, Tang C-K, Terzopoulos D, Chan T, Osher S (2011) Make it home: automatic optimization of furniture arrangement. ACM Trans Graph 30(4): 86:1–86:11

3DJ: An Analytical and Generative Design System for Synthesizing High-Performance Textures from 3D Scans

Sayjel V. Patel, Mark K.M. Tam and Caitlin T. Mueller

Abstract This paper presents “3D Sampling” as a new creative design process. Analogous to sample-based music, 3D Sampling provides conceptual and technical tools for hacking, mixing, and re-appropriating the material behavior, performance features, and structures of real world objects. Building on previous research, this paper demonstrates a new 3D Sampling design system—“3DJ”. 3DJ implements user-guidance features within an evolutionary design process to synthesize new texture designs from 3D scans. A case study demonstrates the use of 3DJ to generate novel designs for a site-specific architectural canopy from 3D scanned textures.

Introduction

3D Sampling reconsiders low-cost, hand-held 3D scanning technologies, such as photogrammetry, as the generative design tools of the future (Patel and Mueller 2015). While 3D scanning technologies, such as photogrammetry, allow anyone with a smartphone to digitize real world objects as point-cloud data, conventional computer-aided design approaches fail to capitalize on a growing treasure trove of free 3D scan data drawn from real world objects. Typically, CAD designers start from abstraction; relying on first principles, known procedures, and classical models to determine the final form of design artefacts. However, 3D Sampling allows a designer to reverse this methodology; making detailed digital replicas of physical objects the starting point for design. By contributing new methods and tools to re-appropriate 3D scan geometries as new design variables, 3D Sampling is a creative design process (Cross 1997) and constructs a new solution space beyond the conventional calculus-driven, rule-based logic, of design modelling. In contrast

S.V. Patel (✉) · M.K.M. Tam · C.T. Mueller
Massachusetts Institute of Technology, Cambridge, USA
e-mail: svpatel@mit.edu

S.V. Patel · M.K.M. Tam
SUTD, Singapore, Singapore

J.S. Gero (ed.), *Design Computing and Cognition ’16*,
DOI 10.1007/978-3-319-44989-0_26

to other methods, such as bio-mimicry and bio-inspired design (Oxman and Rosenberg 2008), 3D Sampling equips designers with conceptual methods and tools to immediately hack, filter, and mix the design features and performance characteristics of real-world objects gathered through 3D scanning. While technological developments enable greater fusion of digital-physical design and manufacturing, conventional "scan-model-print" approaches involve deterministic, verification or measurement oriented applications (Boulaassal et al. 2009), and do not provide an explicit design method to transform 3D scan data into new compositions. Thus, 3D Sampling contributes new possibilities towards the design of creative artifacts.

This paper demonstrates how 3D Sampling offers new design computing and materialization possibilities and extending 'sampling' as general design procedure. Called "3DJ," the proposed design system contributes: (1) a new design modelling approach, combing user interaction with semi-automated statistical characterization, performance analysis, and generative design methods for 3D scan textures; (2) new design rules, including procedural hybridization, collage, and morphing rules, for 3D texture synthesis; and (3) case study evaluation of these tools and strategies.

Related Work

This research draws on developments in two areas of design and technical research: (1) trans-disciplinary sampling-based design practices; and (2) developments in the fields of texture classification and synthesis.

Sampling as a General Design Approach

Sampling vast troves of online "big data", open-source networks have facilitated a new digital culture that has revolutionized several design fields:

In music, sampling is the act of taking a portion of a sound recording and reusing it to create original compositions. Miller provides a comprehensive overview of the field of sample-based music (Smith et al. 2011); its motivations, compositional strategies, and history. The central design operations applied by artists include: digital signal processing, convolution, algorithmic filtering, looping, and mixing, serve as analogies for the generative rules and design grammar we develop in our research.

In architecture, sampling first-hand observations, external precedents, and research are strategies that inform the architectural design process. A topic of recent research activities, "critical appropriation" (Miljacki et al. 2013) has been examined as a central design strategy to generate new design ideas from existing examples, transfer knowledge, and contextualize buildings within a broader critical and cultural framework.

In computer graphics, sampling algorithms and methods are used to synthesize new designs from 3D object libraries; "Zoomorphic Design" (Duncan et al. 2015) is a recent field of inquiry that develops a new modelling approach to procedurally collage and morph of 3D models together to produce new designs. Similarly, "Chairgenics" (Sanchez et al. 2013), is a methodology that employs user-guidance, feedback, and morphing features to breed design alternatives from a historical database of chair samples.

In artificial intelligence, sampling to synthesize novel design artifacts for humans is a key feature of computational creativity systems. An example of these systems, culinary recipe generation systems implement new analytical and generative procedures to process recipe databases in order to create novel recipes and provide corresponding preparation procedures and ingredient quantities (Varshney et al. 2013).

The breadth of these example demonstrate sampling as a generalizable design principal. Recognizing that the design process involves consideration of objective, subjective, and intersubjective criteria (Heylighen 1997), a central aim of this work is to develop a 3D Sampling design method to facilitate both quantitative and qualitative design space exploration.

Texture Synthesis and Classification

Image-based texture synthesis explores methods for transforming 2D photographic images samples into entire virtual worlds (Efros et al. 1999). Synthesis algorithms have two primary goals: sampling texture data to generate new image data, and texture transfer. The literature provides a detailed technical description of procedural texture synthesis methods and algorithms considered for this research (Efros and Freeman 2001; Ashikhmin 2001; Ruiters et al. 2010; Wei and Levoy 2000).

Texture classification has been the subject of extensive study in the field of image processing, involving statistical and psychological measurement approaches. A standard procedure for texture characterization remains open-ended (Hertzmann et al. 2001), as texture characterization is a multi-dimensional, subject-specific, problem. The literature summarizes texture evaluation methods (Varma and Zisserman 2005; Varma 2003; Liu and Wang 2003). Feature detection, encompassing partitioning methods to group similar pixel information, is a particularly important feature of texture classification systems. The literature contains current research into texture segmentation methods considered for our design system (Mooneghi et al. 2014; Srinivasan and Shobha 2008; Rajalakshmi and Subashini 2014; Sklansky 1978).

Research Aims

Building on early 3D Sampling research (Patel and Mueller 2015), this research develops a design methodology, concerned with the generation, evaluation, and selection of texture designs based on performance-based criteria, while seeking to (1) demonstrate statistical and analytical methods for characterizing, and extracting design features, material characteristics, and performance attributes, (2) demonstrate simulation methods for predicting performance behaviors of 3D textures, and (3) develop design syntax to link operations with respect to the performance of synthesized designs.

Paper Outline and Conceptual Overview

3DJ is a design framework developed for 3D Sampling textures—employing procedural texture generation methods to create new design ideas and selection methods to determine the ones have best performance attributes. Similar to computational creativity systems (Ventura 2011), 3DJ aims to use both the innate of ability for a computer to generate design alternatives, and the ability of a human to discern between options. As such, it is critical to develop both user interaction, and robust generation and evaluation methods (Fry 2011).

The proposed *scan-sample-print* methodology (Fig. 1) contains the following steps: (1) extraction of texture samples from 3D scan model; (2) characterization and initialization of sample within data-base; (3) design rule application and generation of sample offspring; (4) design scenario specification, optimization, and selection; (5) design modelling refinement; and (6) texture materialization. The organization of this paper follows the order of this methodology—focusing particularly on the generation and optimization stage; covered respectively in Implementation Step I, and Implementation Step II. The design rules, analysis, and performance-based simulation methods that are applicable to both implementation steps are respectively explained in Fundamentals I, II, and III.

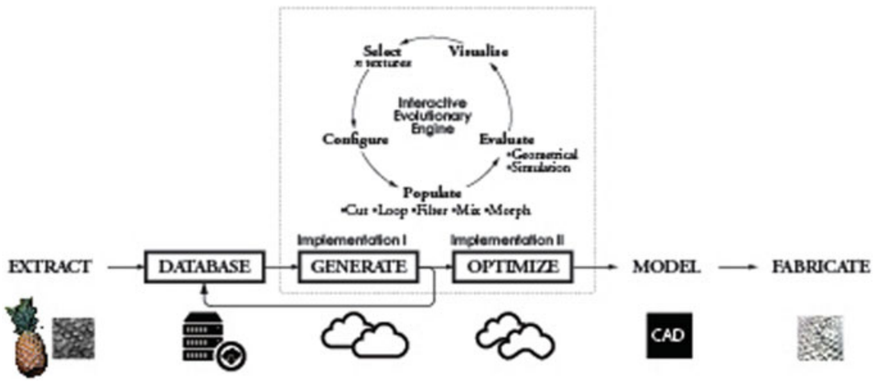


Fig. 1 The 3D Sampling methodology incorporates quantitative and qualitative user feedback and user-guidance features

Extraction Methods

3D Sampling begins with the "Extraction" stage to facilitate the user-collection of planar, and functional surfaces drawn from 3D scan models. A functional surface is continuous over its domain, meaning it does contain have overhang conditions; there is only one z value per coordinate [x, y]. The focus on functional surfaces allows for the use of 2D matrix-based image processing methods in the characterization and procedural synthesis of textures. Textures selected in this implementation (Fig. 2) are chosen to demonstrate common examples encountered in daily life.

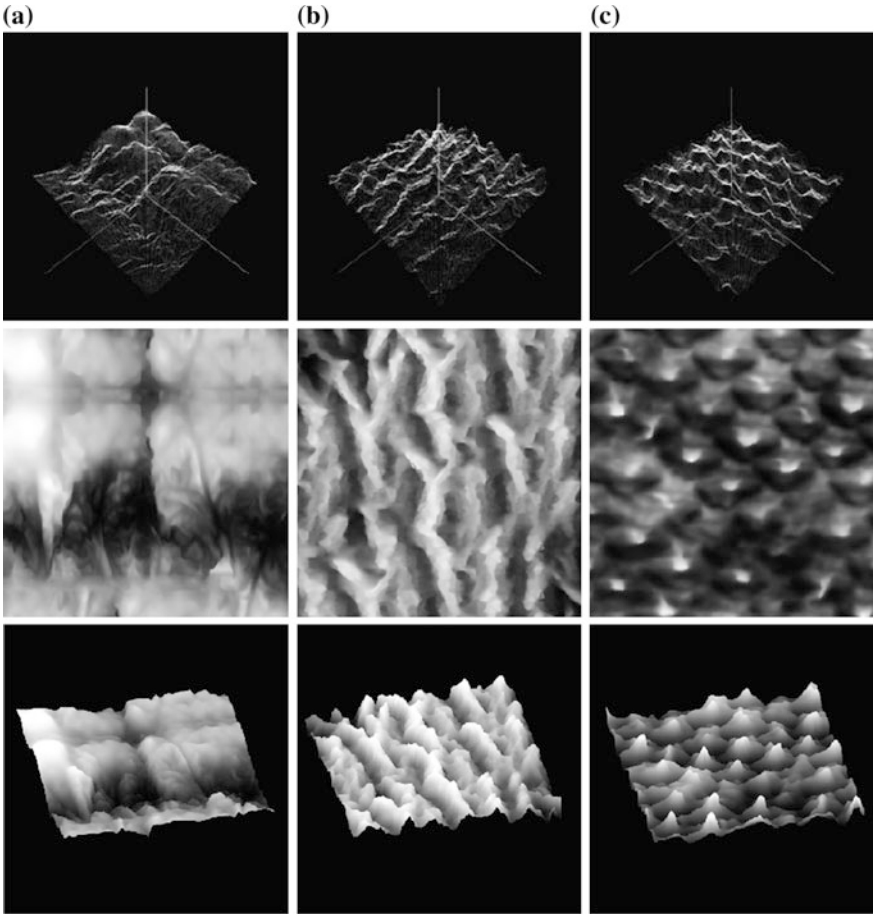


Fig. 2 Extraction and initialization of textures: **a** Lettuce, smooth, continuous; **b** Bark, sharp, continuous; **c** Pineapple, rough, discrete. Representations; point-clouds (*top*), 3D depth images (*middle*), meshes (*bottom*)

Design Rules, Texture Characterization and Simulation Methods

To create new texture designs from a finite number of input samples, the 3DJ methodology is separated into the two primary steps: generation and optimization. The objective of the 'generation' stage is to provide designers with the ability to quickly synthesize a large number of texture variations in a generic context absent of design constraints. Terminating with the final selection of *n* textures, the process is followed by the optimization stage, where textures are mapped to a specific design problem.

This texture-synthesis system provides both the generation and optimization stage with analytical methods for evaluation and design rules for geometric manipulations. As demonstrated in Fig. 3, both generation and optimization share a similar workflow consisting of the (1) population of options by design rules; (2) evaluation of generated options; (3) visualization of evaluation results to users; (4) selection of *n* surfaces by users, and (5) the breeding and repopulation of options, or the termination of the process. To facilitate user interaction, both processes rely on an interactive evolutionary generation engine. This approach differs from standard evolutionary algorithm insofar as designer inputs are required for specimen selection and parameters configuration. Interactive evolutionary algorithms have proven effective in a number of disciplines that incorporate both quantitative and qualitative goals, including structural design (Mueller 2013). Of

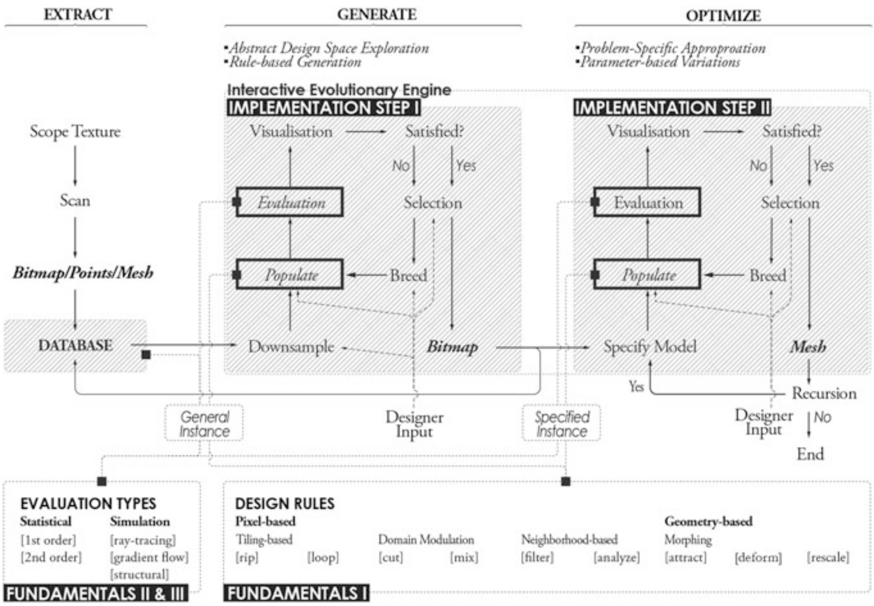


Fig. 3 3D Sampling, detailed overview; extraction, generation, and optimization stages of the methodology

particular significance, this approach allows the optimization process to be adjusted based on unformulated or emerging goals, such as visual impact or constructability requirements. Another difference between the generation and optimization relates to rules application the generation stage involves design space, while the optimization stage requires the specification to a particular site or context. The following 'fundamental' sections describe the process in greater depth.

Fundamental I: Design Rules

The following 'fundamental' sections summarize the design, analysis, and simulation methods used to generate variations, characterize, and analyze performance. To extend the 3DJ design space of texture samples, design rules are required. The generative rules used in this implementation are organized into the two categories of pixel-based and geometric-morphing rules:

Pixel-based rules include both first and second order statistical manipulation of texture bitmap data. Figure 4 summarizes the rules considered for texture synthesis. Using sample-based music as an analogy, the rules are categorized with respect to music terminology and image processing methods: (1) Rip; subset extraction, (2) Loop; feature repetition (Efros and Freeman 2001), (3) Cut; signal distortion, (4) Mix; texture blending and transformation (Hertzmann et al. 2001), and (5) Filter; texture convolution and spectral texture analysis (Liu and Wang 2003).

Geometric morphing rules are procedures for remapping or distorting textures according to continuous global geometries (Fig. 5). Geometric control points and parametric functions provide morphing parameters. These methods are useful for calibrating a generic texture to a specific design problem. They also make computation more efficient, especially in optimization, because it is expensive to set individual translation vectors for all mesh vertices, due to geometric noise (Tam et al. 2016).

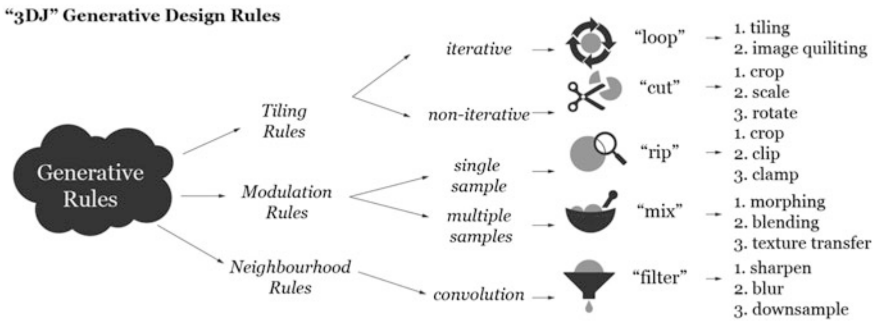


Fig. 4 Summary of "3DJ" pixel-based design rules

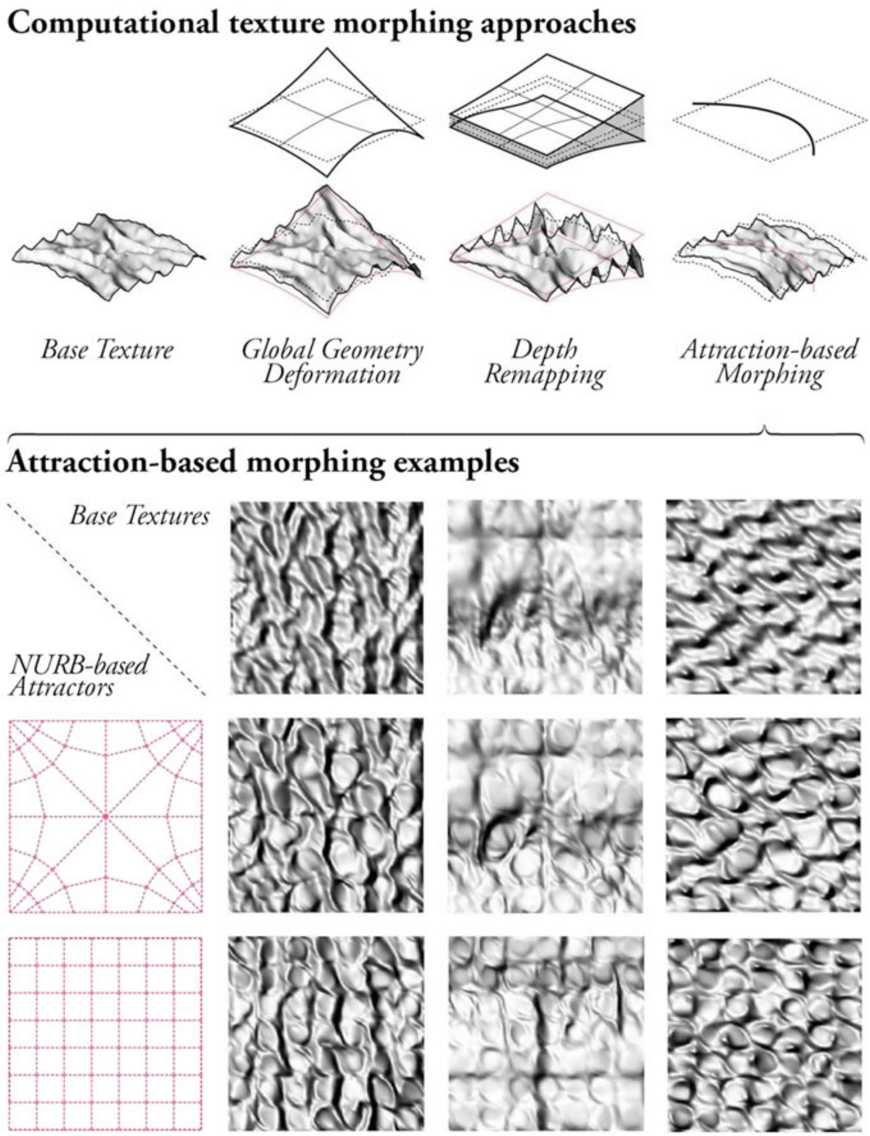


Fig. 5 Different classes of morphing methods, including global geometry deformation, depth remapping, and attraction-based morphing. These procedures make it possible to morph of a texture to supplementary topology with performance characteristics, such as structural patterns; networks of curvature indicating paths of desired material continuity or reinforcement (Tam et al. 2016)

Fundamental II: Texture Characterization

The aim of texture characterization in 3DJ is to allow a designer to rank potential designs with respect to different design criteria. Texture characterization is an important topic in the field of computer vision that is concerned with developing methods for identifying meaningful visual patterns, features and characteristics using first and second-order statistical evaluation of texture images (Varma and Zisserman 2005; Srinivasan and Shobha 2008). Drawing from these existing methods, "3DJ" attempts to extrapolate criteria such as roughness, coarseness, symmetry and flow, to possible performance features. Organized with respect to selection criteria, Table 1 summarizes texture evaluation methods that were considered for this implementation.

Simulation Methods

Simulation methods are used to predict the physical performance of texture samples. As simulation procedures of require computationally expensive specificities, they are often unavailable to designers during the abstract stages of conceptual design (Mueller and Oschendorf 2013). A set of generalized proxy simulation scenarios based on possible design applications were developed to approximate the performance of texture for a number of key phenomena important to designers and engineers:

Optical refraction: Ray-tracing is used to determine the refractive capacity of the texture materialized as a translucent thin shell (Fig. 6). The simulator projects a grid of incidental rays to the texture, and calculates the final location of the

Table 1 Summary of class selectors and statistical and geometric methods

Method	Objective criteria	Subjective criteria	Inter-subjective criteria
Numerical, 1st order	Intensity; average, range, probability; standard deviation, variance, entropy; spectral histogram; co-occurrence matrix	Appearance; symmetry, contrast	Subject blending; similarity comparison
Numerical, 2nd order	Filters; convolution, Gabor filter, Fourier transforms; image pyramids; Markov random fields	Shape-description; coarseness, fineness, regularity, irregularity directionality; gradient flow	Subject-morphing; similarity comparison
Geometrical, simulation based	Performance; reflection, refraction, flow	Feel; roughness, slippery, silky, smooth, regularity, irregularity	Hybridization of performance features

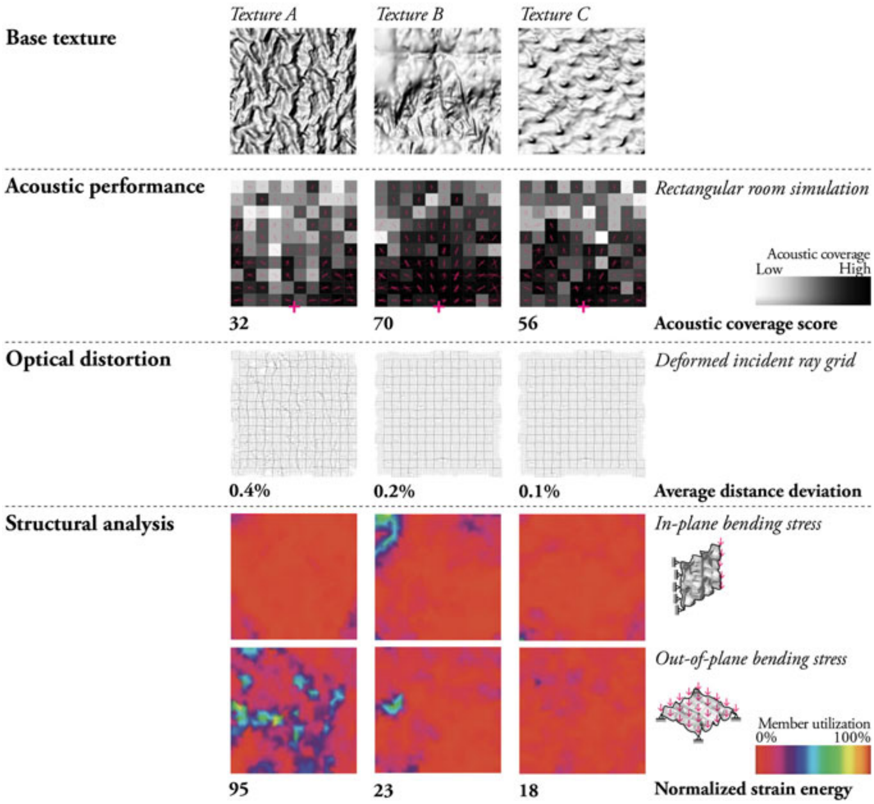


Fig. 6 Various performance simulation types and example results

doubly-refracted rays at a predefined distance from the backside of the texture using Snell's Law (De Greve 2016). The average distance between all source and destination points of traced rays is a quantifiable measure of optical distortion.

Acoustic scattering: A common objective in the engineering design of acoustically performative tiles is the maximization of acoustic scattering (LMNts 2016). Using ray-tracing, sound waves propagate across a virtual rectilinear room tiled with the examined texture from a centralized spherical source. The reflection of the sound waves are measured on a grid of 'listening' points distributed across the room's floor (Fig. 6). The procedure outputs an average and variance figure to indicate respectively the fullness of the sound per point, and the evenness of sound distribution.

Structural simulation: Finite element analysis (FEA) procedures analyze the textures for their potential performance as structural elements for a number of particular types of structural actions: (1) in-plane membrane stress, (2) out-of-plane bending stress, and (3) in-plane bending stress. Structural efficiency is indicated by strain energy normalized to account for total mesh surface area, as the minimization

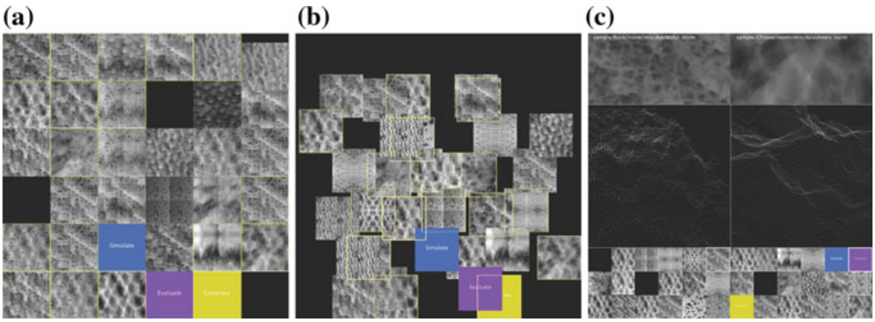


Fig. 7 User experience, **a** guidance mode, user selects seed tiles, **b** sorting of tiles by objective criteria, **c** 3D evaluation and visualization

of strain energy—also referred to as maximization of stiffness (Rozvany et al. 1995; Achtziger 1997)—is frequently used as the objective function in structural optimization processes.

Implementation Step I: Sample Texture Generation

The 'generation' stage develops procedures to guide the production of new texture designs. A prototype design tool was developed to rapidly synthesize new texture alternatives from a limited set of input textures (Fig. 7). Employing an interactive evolutionary design approach, the tool incorporates user guidance and feedback features within a generative engine to facilitate user-interaction in the selection stage. Soliciting qualitative and quantitative-based selection and feedback from the user, the design tool displays candidate solutions as interactive 2D texture tiles. The user selects desirable tiles from the seed pool, while unselected tiles are replaced in subsequent generations. To assist users with subject comparison and ranking, tiles are sorted, in real-time, based on their objective, statistical-characteristics such as bi-axial symmetry, variation, roughness. The tool features both 2D selection modes, featuring a 6 × 6 matrix of tiles, as well as a 3D view mode to provide detailed comparison up to four samples.

Behind the graphical user interface, the tool contains three major classes of automated, real-time operations.

Texture Synthesis, Classification, and Ranking

Texture breeding and generation (Fig. 8): The iterative design tool allows a user to curate the synthesis of new texture designs based on subjective and objective

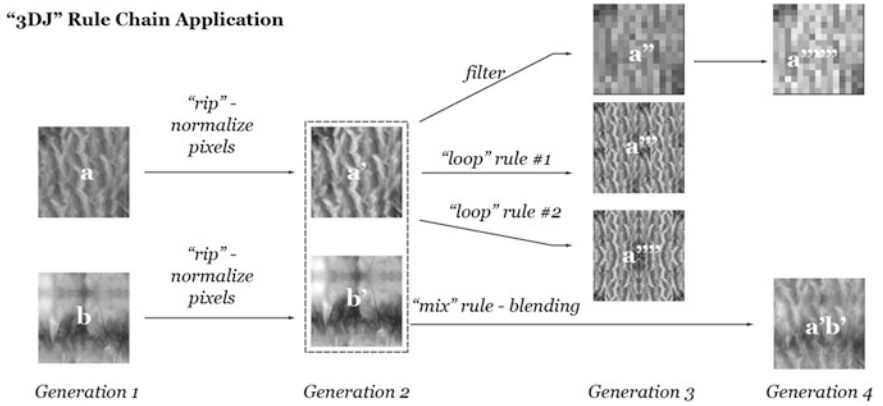


Fig. 8 Operation 1, texture breeding and generation. The *arrows* correspond to transformational rules

criteria. In each iteration, texture samples are drawn from a seed bank and transformed using design rules. The goal of this stage is to incorporate the user in the selection of candidate solutions to ensure desirable solutions are propagated in future generations.

Texture evaluation and characterization (Fig. 9): Calculation of first-order and second-order statistical properties to infer geometric texture characteristics such as coarseness, fineness, directionality, and symmetry. This step aims to assist the

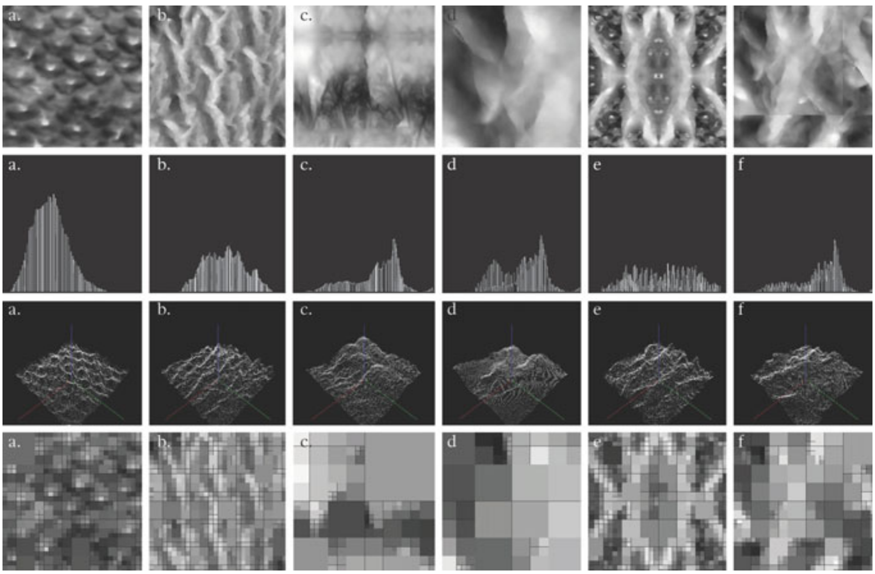


Fig. 9 Operation 2, texture characterization and analysis (by *row*, starting from the *top*): 3D depth-map texture image, 3D point-cloud, pixel histogram, neighborhood-based segmentation

designer with connecting the generic design space of generation with design-oriented properties and application.

Sorting, ranking, and comparison: Sorting operations provide multi-dimensional ranking of attributes that are visualized to the user. For instance, the entropy of a texture may describe the regularity or irregularity of a sample features, while standard deviation (SD) may relate the texture roughness. Another important metric is 'tile-ability' to characterize how well a texture sample can be arranged in combination with other textures. Figure 10 shows one such ranking whereby the pixel SD is plotted (Y-axis) versus the tile ability of a sample image, calculated using a simple b-axial comparison of the edge pixel values.

Implementation Step II: Optimize Textures

This section presents an architectural case study to test the proposed design methodology. The project is to construct an amphitheater canopy shell over a 10 m × 10 m square footprint. The canopy is to be fabricated using a composite thermoforming process (Engelsmann et al. 2010), and its main fixities are at the four corners corresponding to the square footprint: where the canopy will be bolted to four steel truss columns. Thus, the canopy is expected to simultaneously address structural and acoustic concerns. For the case study, three textures were generated

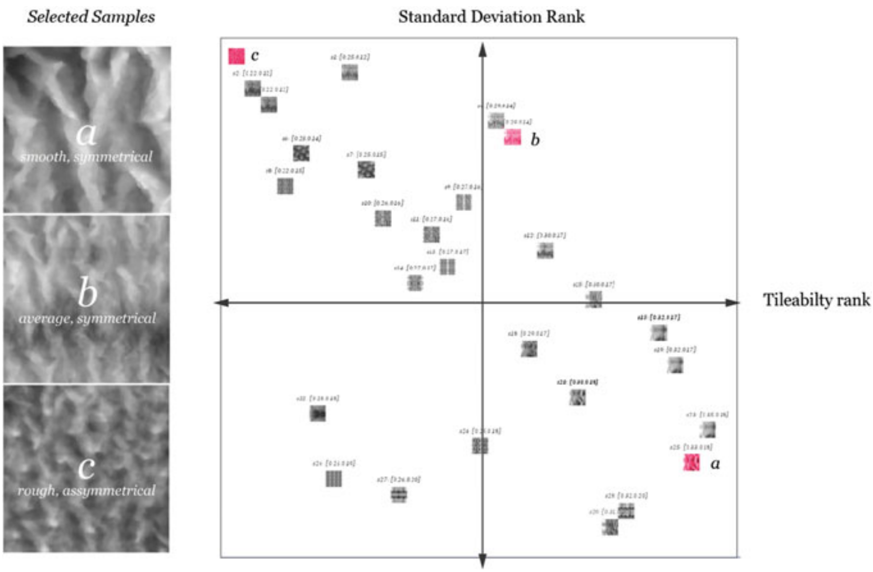


Fig. 10 Operation 3, sorting. Interactive graphs plot texture samples generated by the design tool with respect to design criteria such as variation and tileability

(Fig. 11), and the optimization process is used to finalize texture selection, and to determine their optimal translation.

In order to address both structural and acoustic objectives of the canopy design, the optimization process is organized in two phases occurring at two scales: addressing structural objectives, the first round initially creates the roof-like canopy by mapping a macro-texture to an elevated plane. To address acoustic concerns, the second step superimposes a micro-texture onto the first-round optimization output. By distinguishing between the scales of texture applications, the optimization step aims to select the tiling scale leading to the most impactful performance changes. In this design case, the macro tiling scale primarily achieves architectural and structural objectives, where deep corrugations are desirable both for the maintenance of aesthetic legibility, and their structural depth and for creating robust bending capacity. Since structural behavior is most affected by global geometrical form, morphing-based design rules are utilized extensively. Following the first stage, the three initial textures are superimposed on the initial canopy geometry at a finer tiling to achieve additional acoustic improvements. Both rounds of optimization use a similar set of design variables: (1) texture orientation angle; (2) geometrical morphing by adjustment of the control points defining the NURBS surface where the texture is mapped; and (3) depth morphing by adjustment of the control points of an underlying NURBS surface that defines the scaling of texture point projection values. The optimization procedure relies on genetic algorithms (GA) implemented in the Grasshopper 3D plug-in Galapagos.

Results

The implementation of GA adopts an explorative strategy with high population size and mutation rate in order to encourage the generation of a large diverse design space of potentially viable texture designs (Mueller and Ochsendorf 2015). It can be seen from the results that the process has generally resulted in a two-fold improvement in structural efficiency, as indicated by normalized strain-energy (SE), whereas the process for texture B produced the most efficient design (Fig. 11). The top performing candidate was then used for the second stage of optimization to improve acoustic performance. It can be seen that the application of texture C on the surface of texture B achieved the greatest acoustic performance, among three options with similar attributes.

Discussion

The results demonstrate the use of 3D Sampling to produce a diverse, high-performance solution space of synthesized textures. From a small set of input textures, a user can generate hundreds, or even thousands of new designs. As the

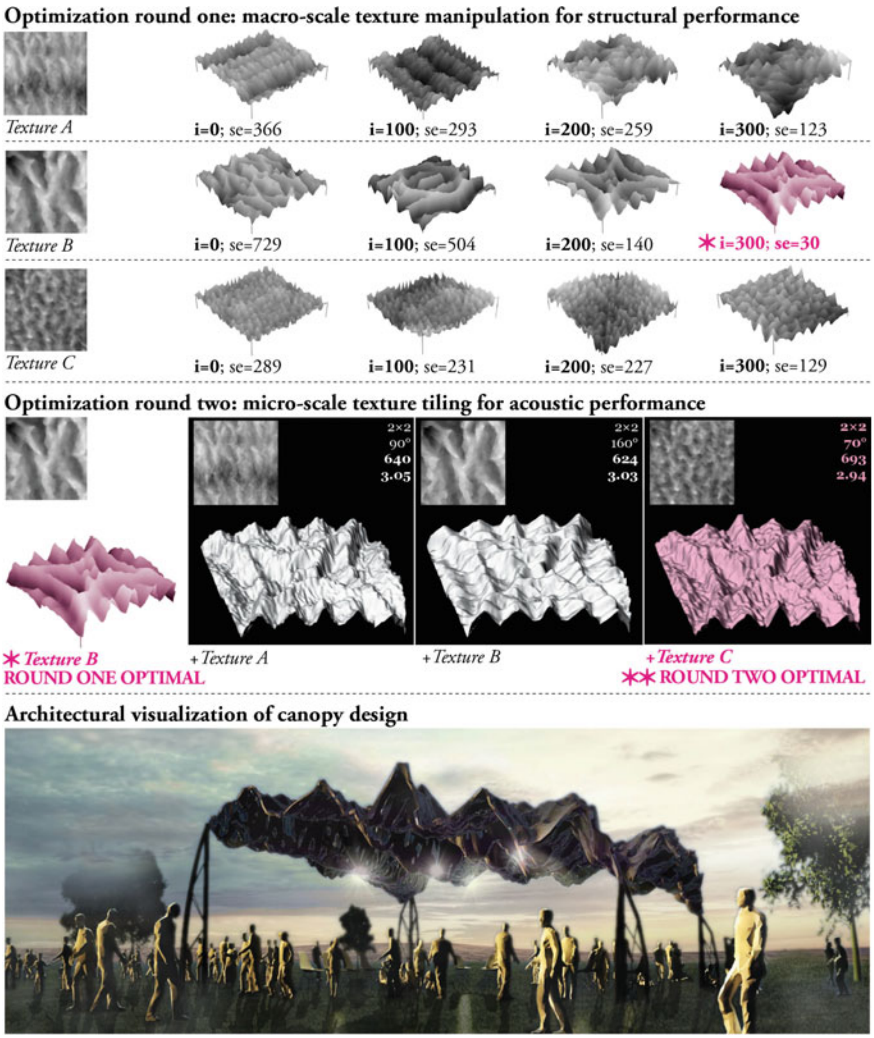


Fig. 11 Optimization results from round one and two. These stages apply macro- and micro-scale textures. The *bottom image* shows how the design case study in context

second round of the case study shows, multiple textures can be mapped to the same geometry to produce unique variations of comparable performance. This process facilitates the emergence of unexpected geometries and options, since the optimal site conditions for applying a texture are variable, and therefore cannot be predicted 'a priori'. Thus, the process demonstrates feedback and how initial design assumptions might be reformulated in light of the design tool.

Preliminary experimentation with the design methodology also suggests a syntactic link between performance of new texture designs, and design rules; this could

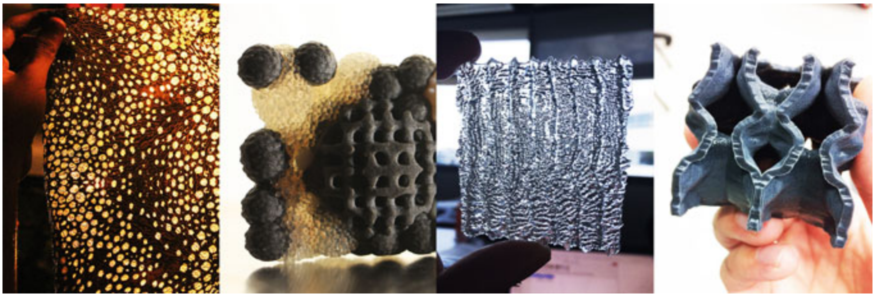


Fig. 12 Fragments of prototype designs implementing the 3D Sampling methodology

be further explored future works. For example, looping rules increase both the roughness, and consistency of a texture sample. Regarding optical performance, convolution blur filters were observed to decrease roughness, resulting in greater transparency, while looping, which increases point density, creates a texture with greater diffusion and specular noise. In future work, rigorous and extensive parameterization studies will have to be conducted before statistically substantiated claims can be made about the relationship between design rules and material performance. Particularly, there is a need to further investigate the efficacy of the proxy-simulation methods in predicting the eventual performance behaviors of the appropriated textures.

While the case study demonstrates the application of 3DJ to the design of an acoustic and structural, canopy structure, in-progress work (Fig. 12) suggest design applications of 3D Sampling beyond architectural design; Presently, this methodology is being used develop a novel surfboard prototype design. In this project, sample texture information contributes to the surfboard's structural infill pattern, the global geometry for fluid-dynamic performances, friction-fit modular connections, ergonomics, water-shedding gripping surface, and optical camouflage.

Conclusion

The work presented in this paper makes the following contributions: (1) a new computational sampling-based design methodology called 3DJ; (2) a technical compilation of meaningful evaluation and simulation methods that are relevant to the appropriation of textures as high-performance design products; (3) a texture synthesis tool allowing users to generate new 3D textures from 3D scans; and (4) texture case studies implementing the design framework, with initial results and evaluation, and a final design case study for applying the framework to an architectural design brief. The speculative tools, frameworks, and experiments presented and assessed here are first steps that exemplify the potential future of a new design and research field which re-contextualizes 3D scanning, once conceived as a process of measurement, as a method for creative production. While the solutions developed within the case study—structural, acoustic, and optical—were framed within an architectural discourse, the implications of the 3DJ approach will have far reaching applications outside of these disciplines. Considering how sample-based music practices have made music creation accessible to a wider audience, 3D Sampling could offer a simplified and more intuitive way for non-experts to edit, manipulate, and synthesize 3D objects.

Acknowledgements The authors would like to thank MIT-SUTD Collaboration, the MIT Digital Structures group, and SUTD Digital Design and Manufacturing and Design Centre for supporting this work. As well as our undergraduate research collaborators: Kelly Khoo, Jonathan Ng, Melissa Mak Li Ping, Sanjay Pushparajan, and Vivek Kalyan Sureshkumar.

References

Achtziger W (1997) Topology optimization of discrete structures: an introduction in view of computational and nonsmooth aspects. In: Rozvany GIN (ed) Topology optimization in structural mechanics. Springer, Vienna, pp 57–100

Ashikhmin M (2001) Synthesizing natural textures. In: Proceedings of the 2001 symposium on interactive 3D graphics

Boulaassal H, Grussenmeyer P, Landes T (2009) Automatic extraction of planar clusters and their contours on building facades recorded by terrestrial laser scanner. Int J Archit Comput 07:1

Cross N (1997) Descriptive models of creative design: application to example. Des Stud 18 (4):427–440

De Greve B (2016) Reflections and refractions in ray tracing. N.p., 2016. Web 19 Apr. 2016

Duncan N, Yu LFI, Yueng SF, Terzopoulos D (2015) Zoomorphic design. ACM transactions on graphics. In: Proceeding of SIGGRAPH 2015

Efros A, Freeman WT (2001) Image quilting for texture synthesis and transfer. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques (SIGGRAPH '01)

Efros A, Leung AA, Leung TK (1999) Texture synthesis by non-parametric sampling. In: Proceedings of the seventh IEEE international conference on computer vision

Engelsmann S, Spalding V, Peters S, Stein R (2010) Plastics: in architecture and construction. Birkhäuser, Basel (print)

Fry C (2011) Closing the gap between analytics and action. Informs Anal Mag 4(6):4–5

Hertzmann A, Jacobs CE, Oliver N et al (2001) Image analogies. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques (SIGGRAPH '01)

Heylighen F (1997) Objective, subjective and intersubjective selectors of knowledge. Evol Cognit 3(1):63–67

Liu X, Wang D (2003) Texture classification using spectral histograms. IEEE Trans Image Process 12(6):661–670

LMNts (2016) Acoustic scattering research—the setup. N.p., 2013. Web 19 Apr 2016

Miljacki A, Reeser A, Berman I, Mitchel E (2013) Critical appropriation: discursive networks of architectural ideas. In: New constellations/new ecologies, Proceedings of the 101st annual meeting of the ACSA

Miller P, Kid TS, Spooky DJ (2008) Sound unbound: sampling digital music and culture. MIT Press, Cambridge, MA

Mooneghi SA, Saharkhiz S, Varkiani SMH (2014) Surface roughness evaluation of textile fabrics: a literature review. J Eng Fibers Fabr 9(2)

Mueller C (2013) structure FIT design tool. www.caitlinmueller.com/structurefit

Mueller C, Oschendorf J (2013) An integrated computational approach for creative conceptual structural design. In: Proceedings of the international association for shell and spatial structures (IASS) symposium 2013, Wroclaw, Poland

Mueller C, Ochsendorf J (2015) Combining structural performance and designer preferences in evolutionary design space exploration. Autom Constr 52(2):70–82

Oxman N, Rosenberg JL (2008) Material-based design computation: an inquiry into digital simulation of physical material properties as design generators. Int J Archit Comput 5(1):26–44

Patel SV, Mueller CT (2015) 3-D sampling: haptic and optically performative textures remixed from 3-D scans. In: Modelling behaviour: proceedings of the design modelling symposium, Copenhagen

Rajalakshmi M, Subashini P (2014) Texture based image segmentation of Chili pepper X-ray images using gabor filter. IJASCSE 3(3):44–51

Rozvany GIN, Bendsøe MP, Kirsch U (1995) Layout optimization of structures. Appl Mech Rev 48(2):41–119

Ruiters R, Schnabel R, Klein R (2010) Patch-based texture interpolation. Comput Graph Forum 29 (4):1421–429

Sanchez M, Oleg F, Turlif V, Pakso A (2013) Morphological shape generation through user-controlled group metamorphosis. Comput Graph 37(6):620–627

Sklansky J (1978) Image segmentation and feature extraction. IEEE Trans Syst Man Cybernet 8 (4):237–247

Smith G, Whitehead J, Mateas M (2011) Computers as design collaborators: interacting with mixed-initiative tools. In: Proc. ACM creat. cogn. worshop semi-autom. creat. (SAC 2011)

Srinivasan GN, Shobha G (2008) Statistical texture analysis. In: Proceedings of world academy of science, engineering, and technology, vol 36

Tam KMM, Coleman JR, Fine NW, Mueller CT (2016) Robotics-enabled stress line additive manufacturing. In: D Reinhardt, R Saunders (eds) Robotic fabrication in architecture, art and design. Springer, Cham

Tuceryan M, Jain AK (1990) Texture segmentation using Voronoi polygons. IEEE Trans Pattern Recognit Mach Intell 12:211–216

Varma, M, Zisserman A (2003) Texture classification: are filter banks necessary? In: IEEE computer society conference on computer vision and pattern recognition, 2003

Varma M, Zisserman A (2005) A Statistical approach to texture classification from single images. Int J Comput Vis Int J Comput Vis 62(1–2):61–81

Varshney LR, Pinel F, Varshney KR et al (2013) A big data approach to computational creativity. In: IEEE international conference on cognitive informatics and cognitive computing, New York

Ventura, D. (2011). No free lunch in the search for creativity. In: Proceedings of the 2nd international conference on computational creativity, Mexico City, Mexico, pp 108–110

Wei LY, Levoy M (2000) Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the 27th annual conference on computer graphics and interactive techniques (SIGGRAPH ‘00)

Automated Best Connected Rectangular Floorplans

Krishnendra Shekhawat and José P. Duarte

Abstract As part of a larger research aimed at developing design aids for architects, this paper presents the "automated" generation of the "best connected" rectangular floor plans, satisfying given topological and dimensional constraints. It has been seen that architects, knowingly or unknowingly, have often used either the golden rectangle or the Fibonacci rectangle in their works throughout history. But it was hard to find any specific reason for such use, other than aesthetic. In 2015, Shekhawat showed that they are among the best connected rectangular arrangements (dimensionless rectangular floor plans) and that this may well be another reason for their frequent use in architectural design. In this work, an alternative algorithm is presented which generates $n - 3$ best connected rectangular arrangements, being n the number of rooms. Then, this concept is further extended for constructing the best connected dimensioned rectangular floor plans. The goal is to provide an optimal solution for the rectangular space allocation problem, while satisfying given topological and dimensional requirements.

Introduction

A floor plan is a polygon, the plan boundary, divided by straight lines into component polygons called rooms. The edges forming the perimeter of each room are termed walls. The region not enclosed by the boundary is called exterior. For a detailed discussion regarding definitions related to floor plans, refer to Rinsma (1987a).

K. Shekhawat (✉) · J.P. Duarte
Universidade de Lisboa, Lisbon, Portugal
e-mail: krishnendra.iitd@gmail.com

Present Address:
K. Shekhawat
BITS Pilani, Pilani Campus, Pilani, India

Present Address:
J.P. Duarte
The Pennsylvania State University, State College, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_27

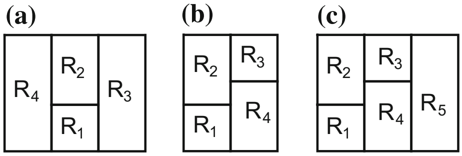


Fig. 1 Best connected rectangular arrangements with four rooms

One of the well-known problems associated with the generation of floor plans is called space allocation problem. It is concerned with the computational arrangement of rooms in a floor plan while satisfying given topological and dimensional constraints. The topological constraints are usually given in terms of adjacencies between rooms and between them and the exterior of the plan. Mathematically, two rooms of any floor plan are adjacent if they share a wall or a section of wall; it is not sufficient for them to touch at a point only. Architecturally, we can say that the portion of shared wall should permit one to insert a door to go through. Therefore, to have an architectural meaning and to guarantee the sufficient space for insertion of a door between adjacent rooms, in this paper, we are generating floor plans where the adjacent rooms share a complete wall of at least one of the rooms. For example, in Fig. 1a, R_1, R_2 and R_3 are adjacent; R_1 is sharing a complete wall with R_2 and R_3 but R_3 is not sharing a complete wall with R_1. The dimensional constraints involve shapes or sizes of each room and the actual floor plan.

For every floor plan there exist an adjacency graph where vertices represent rooms and the exterior, and two vertices are joined by an edge whenever the corresponding rooms are adjacent. The adjacency graph is the dual of the floor plan which is treated as a graph. If in an adjacency graph, the exterior is ignored then it is called weak dual or dual of the graph, i.e., the dual graph of a floor plan is a simple undirected graph obtained by representing each room as a vertex and then drawing an edge between any two vertices if the corresponding rooms are adjacent. Clearly, the dual graph is always planar and connected.

It is easy to perceive why adjacency relations are important in architectural designs; if two rooms are adjacent then it is possible to make them accessible to each other via a door. Also, the overall patterns of adjacency relations determine circulation routes in a building. Furthermore, the rooms having adjacencies to the exterior can have windows, thus enabling natural lighting and ventilation. Therefore, connectivity of a floor plan is defined in terms of adjacency relations among rooms and the comparison of the connectivity of different floor plans is done by comparing the connectivity of the corresponding dual graphs. If two connected dual graphs have the same number of vertices, then the dual graph having more edges is considered to be more connected, i.e., if two floor plans are made up of the same number of rooms, then the floor plan whose dual graph has more edges is more connected.

A rectangular floor plan is a floor plan in which the plan's boundary and each room are rectangles. This work is concerned with the space allocation problem associated with the rectangular floor plans. Kalay (2004) talked about two approaches for solving the space allocation problem, namely, an additive approach and a permutation approach [for the details, refer to March and Steadman (1971)]. The additive approach starts with an empty floor plan and builds up a low cost layout with one solution at a time while the other approach goes through every possible layout and searches for one having the least cost. In both approaches, the aim is to minimize the sum of the weighted distances between different rooms, measured from the center of one room to the center of an adjacent one. The least cost solution is the one in which the average sum of all the distances between adjacent rooms is minimal. In this paper, a new approach is introduced where the aim is to construct the best connected floor plans only. Here, we first generate dimensionless rectangular floor plans with maximum connectivity and from these floor plans, we construct dimensioned floor plans. Please note that, in this paper, a dimensionless rectangular floor plan is called “a rectangular arrangement”.

Any rectangular arrangement with n rooms, denoted by $R^A(n)$, has at most $3n - 7$ edges in its dual graph provided that $n > 3$ [for proof, refer to Shekhawat (2015a, Theorem 2)]. It implies that any $R^A(n)$ is best connected if its dual graph has $3n - 7$ edges. A best connected $R^A(n)$ is denoted by $R^A_{BC}(n)$.

It may not be difficult to find and construct a $R^A(n)$ having $3n - 7$ edges in its dual graph. However, to automatically generate $R^A_{BC}(n)$ is different and even more difficult than finding a $R^A_{BC}(n)$. In fact, by automated generation of $R^A_{BC}(n)$, we mean to provide an algorithm that can construct a $R^A_{BC}(n)$ for any given value of n, i.e., a $R^A(n)$ which is rectangular and best connected for any number of rooms. For example, the $R^A(4)$ in Fig. 1a is best connected and can be further extended to remain best connected. For better understanding, consider Fig. 5a where four rooms are added to the $R^A(n)$ in Fig. 1a to generate a best connected $R^A(8)$ and using the algorithm given in Sect. 2.1, the $R^A(4)$ in Fig. 1a can be further extended to remain best connected for any number of rooms. Therefore, the $R^A(4)$ in Fig. 1a is an automated best connected rectangular floor plan. Now consider the $R^A(4)$ in Fig. 1b which is best connected for four rooms but in any case it cannot be extended to remain best connected for 5 rooms because to remain best connected the 5th room should be adjacent to 3 existing rooms, which is not possible for Fig. 1b (consider adding a room so that the plans remains rectangular, see Fig. 1c). It means that Fig. 1b is a $R^A_{BC}(4)$ but it is not an “automated” $R^A_{BC}(4)$. By “automated” we mean that when a solution is being built, the arrangements in intermediate steps are also rectangular and best connected. Now, we are going to talk about the golden rectangle and the Fibonacci rectangle because they are automated $R^A_{BC}(n)$ (Shekhawat 2015a).

The golden rectangle is a rectangle whose ratio of width over height is equal to $\phi = \frac{1+\sqrt{5}}{2}$ with a geometric property as follows:

A square with side length one, can always be removed from a rectangle of sides $1 \times \phi$, to obtain a new rectangle with sides $\frac{1}{\phi} \times 1$, which is similar to the original one. Hence, the construction can be repeated (Walser 2001).

In a Fibonacci sequence, each of its terms is obtained from the sum of the two preceding terms, i.e., $F_{n+1} = F_n + F_{n-1}$ for $n > 1$ where $F_0 = F_1 = 1$. A Fibonacci rectangle is a rectangle with side lengths x and y such that either x/y or y/x is equal to (F_{n+1}/F_n) for some non-negative integer n. A sequence of Fibonacci rectangles can be constructed by starting from two unit squares one above another, one first adjoins a square of side length two to their right, so as to obtain a Fibonacci rectangle with sides $F_3 \times F_2$. Then one adjoins a square of side length F_3 below, so as to obtain a Fibonacci rectangle with sides $F_5 \times F_3$, etc. following a clockwise movement (Shekhawat 2015a).

It is interesting to see that the golden rectangle and the Fibonacci rectangle are similar, geometrically (in terms of the position of rooms) and topologically (in terms of adjacency relations between rooms). Dimensionally they differ slightly, in terms of areas of rooms but for a larger value of n, the ratio of two successive Fibonacci numbers (F_{n+1}/F_n) approaches ϕ.

The work of many researchers in the literature shows that, many buildings and architectural designs (from ancient to contemporary time) have been developed using the golden rectangle or the Fibonacci rectangle. As an illustration, consider a very well-known work of Palladio, the Villa La Rotonda, shown in Fig. 2. It can be seen in Fig. 3 that, the Villa Rotonda is constructed using the following four golden rectangles:

For the golden rectangle at the upper right side, the second square is situated above the first one and the other squares are arranged in clockwise direction. In the golden rectangle at the lower left side, the position of the first and second squares is swapped and remaining squares are arranged in clockwise direction. In the golden rectangle at the upper left side, the second square is situated above the first one and anti-clockwise movement is considered. For the golden rectangle at the lower right side, the second square is situated below the first one and the other squares are arranged in anti-clockwise direction.

Fig. 2 Villa La Rotonda in Vicenza

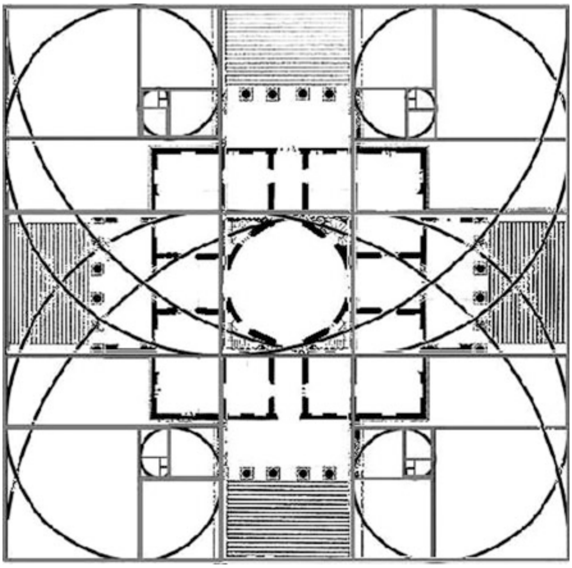


Fig. 3 A golden section diagram of Villa La Rotonda

On the other hand, the use of the Fibonacci rectangle in architectural design is not found as often. Nonetheless, recent findings suggest that it might have been used more often than it has been realized (Corinna and Christopher 2002; Fern´andez-Llebrez and Fran 2013; Bartoli 2004; Park and Lee 2009). In all these works, however, we have not found a clear logical or mathematical reason for the use of the golden rectangle and the Fibonacci series so often by architects. A very common, psychological reason that has been pointed out is that they guarantee visually pleasing constructions (Green 1995). Recently, Shekhawat (2015a) showed that they are best connected rectangular arrangements for any number of rooms, i.e., both of them are automated $R_{BC}^{A}(n)$. We can say that, this may be another reason, at least mathematical, for their high presence in architectural designs. From here onwards, $R_{BC}^{A}(n)$ stands for an automated $R_{BC}^{A}(n)$.

As such, in this paper, we propose an algorithm called C-RLAB (the explanation of the name is related to the way rooms are allocated which is given after the algorithm is described below), to generate $n - 3$ best connected rectangular arrangements for n rooms, that are geometrically (in terms of the position of rooms) and topologically (in terms of adjacency relations between rooms) different from the golden rectangle and the Fibonacci rectangle. Further, from these rectangular arrangements, we construct best connected rectangular floor plans while considering given dimensional constraints.

In 1970s and 1980s, many researchers have worked over the problem of constructing a floor plan for a given adjacency graph using graph theoretical methods. Still the generalization of this problem and its solution are open, i.e., for all adjacency graphs, it is possible to have a generic solution that satisfies the adjacency relations set by any adjacency graph. The larger aim of this research is to construct a generic rectangular floor plan. The construction of a best connected rectangular floor plan is a first step towards the representation of generic rectangular floor plans.

The described work represents a mathematical approach to the design of layouts that does not intend to substitute designers but, once the algorithm is implemented, work as a design assistant that suggest best connected floor plans for the architect to work with. Best connected floor plans are useful because in best connected rectangular floor plans, the total number of connections among the rooms is maximum and they keep topological flexibility. This means that once a floor plan is generated the architect is free to define appropriate connections between rooms, that is, which adjacent spaces to connect through doors.

Best Connected Rectangular Arrangements

In this section, first we present an algorithm that generates $n - 3$ rectangular arrangements for n rooms. Then we will show that the proposed arrangements are best connected.

The C-RLAB Algorithm

In this algorithm, we deal with two types of rectangles. Each type of rectangle is given by its position which is the upper left vertex, and its size which consists of width (the dimension measured horizontally) and height (the dimension measured vertically).

The first kind of rectangle is a new room R_i drawn at stage i of the construction process; whose position is denoted by (x_i, y_i), and whose size is the given pair (ℓ_i, h_i). The second kind of rectangle is a new rectangular composition $R_{BC}^{A}(i)$. Its position also depends on i; we denote it by (X_i, Y_i), and the size of $R_{BC}^{A}(i)$ is denoted by the pair (L_i, H_i). In the discussion that follows, n is the total number of rooms to be allocated, i is the index of the room that is being considered for allocation, and k is the number of rooms successively arranged one above the other. The values of (L_i, H_i) are computed as follows:

For $2 \leq i \leq k$ or $(i - k) \equiv 3 \pmod 4$ or $(i - k) \equiv 0 \pmod 4$, we have

$$L_i = L_{i-1}, \quad H_i = H_{i-1} + h_i$$

For $(i - k) \equiv 1 \pmod 4$ or $(i - k) \equiv 2 \pmod 4$, we have

$$L_i = L_{i-1} + \ell_i, \quad Hi = H_{i-1}$$

Here, $L_1 = \ell_1$ and $H_1 = h_1$.

The steps of the algorithm are as follows:

1. We start by placing the first room say R_1 at position (x_1, y_1) in a two-dimensional plane with size (ℓ_1, h_1). See Fig. 4a.
2. Let $k = 2$ and $i = 2$.
3. If $k < n - 1$, go to Step 4, otherwise, terminate the algorithm (we will see in next section that, for best connected rectangular arrangements, we should have $1 < k < n - 1$).
4. Consider $(x_i, y_i) = (X_{i-1}, Y_{i-1} + h_i)$, $(\ell_i, h_i) = (L_{i-1}, h_i)$.
   In this step, room R_i is drawn above and adjacent to R_{i-1} with width $\ell_i = \ell_{i-1}$ in such a way that its lower left vertex is the upper left vertex of R_{i-1}. Here, the idea is to arrange k rooms one above the other, from bottom to top. For example in Fig. 4, for $k = 3$, R_2 is arranged above R_1 (see Fig. 4b) and R_3 is arranged above R_2 (see Fig. 4c).
   It should be clear from this step that, in this algorithm, after positioning each room, our aim is to obtain a rectangular composition of all the positioned rooms; and to achieve this, the width or height of rooms are adjusted accordingly.
5. Increase i by one. If $i \leq k$, go to Step 4 otherwise go to Step 6.
6. After arranging k rooms, we arrange all the remaining rooms one by one. The next room is arranged as follows:

   (i) If $(i - k) \equiv 1 \pmod 4$, then $(x_i, y_i) = (X_{i-1} + L_i, Y_{i-1})$, $(\ell_i, h_i) = (\ell_i, H_{i-1})$
   The next room R_i is placed right to the obtained rectangular com-position $R^A_{BC}(i-1)$ such that it is adjacent to $R^A_{BC}(i-1)$ and its upper left vertex is the upper right vertex of $R^A_{BC}(i-1)$. For example in Fig. 4d, R_4 is placed to the right of $R^A_{BC}(3)$.

   (ii) If $(i - k) \equiv 2 \pmod 4$, then $(x_i, y_i) = (X_{i-1} - \ell_i, Y_{i-1})$, $(\ell_i, h_i) = (\ell_i, H_{i-1})$
   The next room R_i is placed left to the $R^A_{BC}(i-1)$ such that it is adjacent to $R^A_{BC}(i-1)$ and its upper right vertex is the upper left vertex of $R^A_{BC}(i-1)$. For example in Fig. 4e, R_5 is placed to the left of $R^A_{BC}(4)$.

   (iii) If $(i - k) \equiv 3 \pmod 4$, then $(x_i, y_i) = (X_{i-1}, Y_{i-1} + h_i)$, $(\ell_i, h_i) = (L_{i-1}, h_i)$
   The next room R_i is positioned above $R^A_{BC}(i-1)$ such that it is adjacent to $R^A_{BC}(i-1)$ and its lower left vertex is the upper left vertex of the $R^A_{BC}(i-1)$. For example in Fig. 4f, R_6 is placed above $R^A_{BC}(5)$.

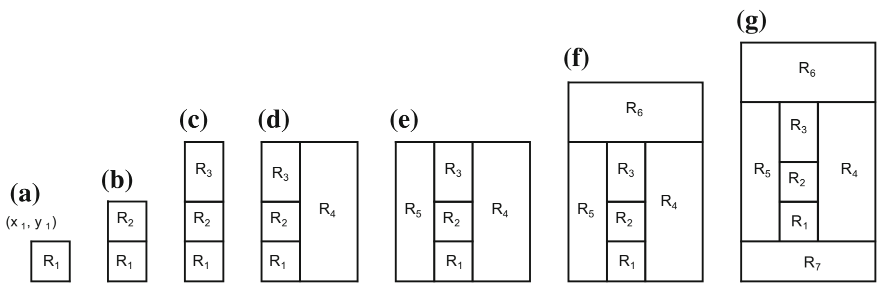


Fig. 4 Construction of rectangular arrangements using the C-RLAB algorithm

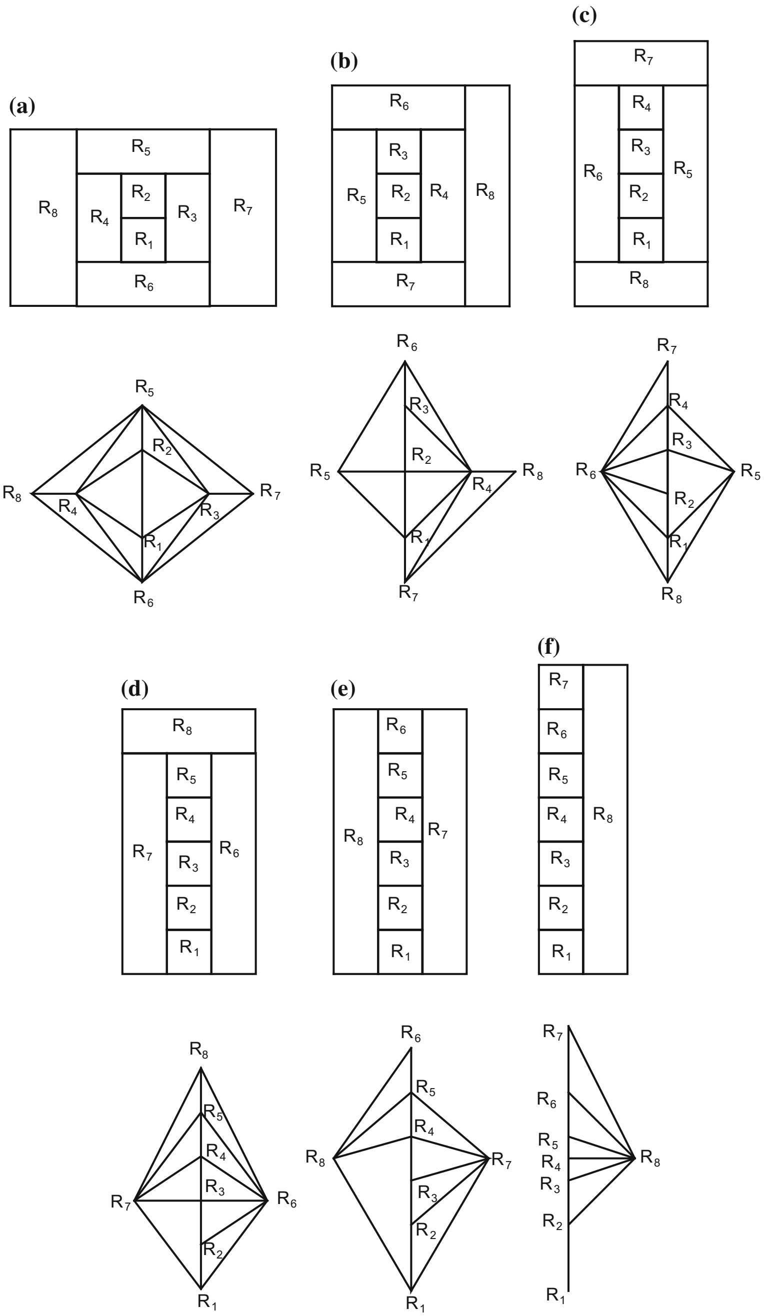


Fig. 5 Rectangular arrangements generated using the C-RLAB algorithm and their dual graphs

(iv) If $(i - k) \equiv 0 \pmod 4$, then $(x_i, y_i) = (X_{i-1}, Y_{i-1} - H_{i-1}), (\ell_i, h_i) = (L_{i-1}, h_i)$ The next room R_i is placed below $R^A_{BC}(i-1)$ such that it is adjacent to $R^A_{BC}(i-1)$ and its upper left vertex is the lower left vertex of the $R^A_{BC}(i-1)$. For example in Fig. 4g, R_7 is placed below $R^A_{BC}(6)$.

7. Increase i by one. If $i \leq n$ go to Step 6 otherwise increase k by one and go to Step 3. For example, for $n = 8$, 5 possible $R^A(8)$ constructed using the C-RLAB algorithm, are illustrated in Fig. 5a–e. We can see that, in these arrangements, the value of k varies from 2 to $n - 2 = 6$.

In the above algorithm, k rooms are first arranged in the center, then the remaining rooms are arranged in the following order, i.e., right, left, above and below, therefore, we call it C-RLAB algorithm.

Connectivity

In this section, we prove that the rectangular arrangements generated by the C-RLAB algorithm are best connected.

To proceed further, refer to Fig. 5 with six $R^A(8)$. We can verify that, the number of edges in dual graphs corresponding to the $R^A(8)$ in Fig. 5a–e is equal to $3n - 7 = 3 \times 8 - 7 = 17$, hence these $R^A(8)$ are best connected. But, the dual graph of $R^A(8)$ in Fig. 5f has only 13 edges, it means that the corresponding $R^A(8)$ is not best connected. It is interesting to see that, the values of k for the $R^A(8)$ in Fig. 5a–e are 2, 3, 4, 5 and 6 respectively, i.e., $1 < k < n - 1$. Corresponding to all these values of k, all $R^A(8)$ in Fig. 5a–e (generated by the C-RLAB algorithm) are best connected; but the value of k for the $R^A(8)$ in Fig. 5f is $n - 1 = 7 > n - 2$ and it is not best connected. Hence, in the C-RLAB algorithm, to have the best connected rectangular arrangements, we must have $1 < k < n - 1$.

Clearly, Fig. 5 presents $n - 3 = 5$ best connected rectangular arrangements for n = 8 rooms, i.e., each $R^A(8)$ has $3n - 7 = 3 \times 8 - 7 = 17$ edges (refer to Fig. 5 for the dual graphs of each of the $R^A(8)$). It can be easily verified that, in all the Fig. 5a–e, if the next room is drawn according to the C-RLAB algorithm, it would be adjacent to exactly three existing rooms. That is, if the dual graph of any $R^A(k)$ constructed using the C-RLAB algorithm has $3k - 7$ edges and next room, say R_{k+1}, is adjacent to exactly three rooms of the $R^A(k)$, then the dual graph of $R^A(k + 1)$ has $3k - 7 + 3 = 3(k + 1) - 7$ edges. Hence, the number of edges would always be $3n - 7$ for n rooms. By considering mathematical induction on n, we can conclude that:

Observation 1 The $R^A(n)$, $n > 3$, generated by the C-RLAB algorithm are always best connected.

Also, from Fig. 5, we can see that, a $R^A(n)$ constructed using the C-RLAB algorithm is best connected and $1 < k < n - 1$. This concludes that:

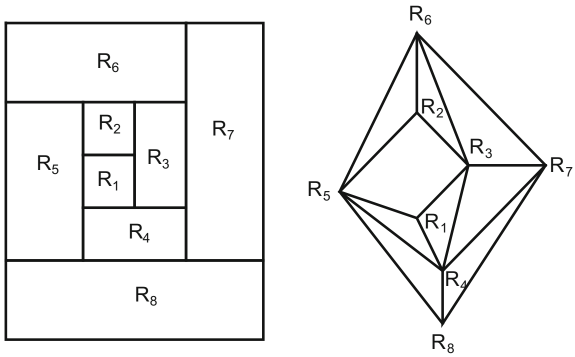


Fig. 6 The dimensionless Fibonacci rectangle and its dual graph

Observation 2 The C-RLAB algorithm generates $n - 3$ best connected rectangular arrangements for n rooms when $n > 3$.

Due to different values of k in all $n - 3$ rectangular arrangements generated by the C-RLAB algorithm, we can say that:

Observation 3 The C-RLAB algorithm generates $n - 3$ best connected rectangular arrangements with n rooms, which are geometrically and topologically different from each other and from the Fibonacci rectangle and the golden rectangle (for a better understanding, refer to Fig. 6 where the Fibonacci rectangle and its adjacency graph for $n = 8$ are demonstrated).

Best Connected Rectangular Floor Plans

In the above section, we presented an algorithm for constructing best connected rectangular arrangements. In this section, by adding few steps to the C-RLAB algorithm, we construct the best connected rectangular floor plans, i.e., the best connected rectangular arrangements that satisfy given dimensional constraints.

In this work, we are using a graph theoretical approach for the generation of rectangular floor plans. In literature, many well-known researchers have used this approach for the same purpose. Some of the related works are as follows:

The application of graph theory to architectural design was first presented by Levin (1964). Then, Grason graphs (1970) and rectangular dualization techniques were developed to address the adjacency requirements (Baybars and Eastman 1980; Kozminski and Kinnen 1984; Bhasker and Sahni 1988; Lai and Leinwand 1988). Roth et al. (1982) presented a method for constructing a dimensioned plan from an adjacency graph. In the same year, Baybars (1982) presented a method for enumerating floor plans with circulation spaces. In 1985, Robinson and Janjic (1985) showed that, if areas are specified for rooms with a given maximal outer-planar adjacency graph, then any convex polygon with the correct area can be divided into convex rooms to satisfy both area and adjacency requirements. In 1987, Rinsma (Rinsma 1987b) showed that, for any given maximal outer-planar graph with at most four vertices of degree 2, it is not always possible to find a rectangular floor plan satisfying the adjacency and area conditions. In 1987, Rinsma (Rinsma 1988) provided conditions for the existence of rectangular and orthogonal floorplans for a given tree, representing the required adjacencies between rooms and areas for each room (any connected graph without cycles is a tree). In recent works, Terzidis (2007) developed a computer program called "auto-PLAN" which generates architectural plans, given a site's boundary and an adjacency matrix. Wong and Chan (2009) proposed an evolutionary algorithm for finding adjacencies between functional spaces and showed that it can be a very useful tool in architectural layout design tasks. In this paper, we are looking for the floor plans corresponding to the graphs with $3n - 7$ edges. Specifically, our aim is to construct best connected floor plans with minimum area.

From the work of past researchers, we can say that, a rectangular floor plan, denoted by R^F, can be generated in the following ways:

- Addition: It concerns the addition of rectangular pieces, like tiles, to produce a rectangular plan (see Krishnamurti and Roe 1979)
- Dissection: It concerns the division of a large rectangle into smaller rectangular pieces. This process is called rectangular dissection (see Mitchell et al. 1976; Earl 1977; Flemming 1978a, b, 1980); Bloch and Krishnamurti 1978; Bloch 1979).

To introduce the dimensional constraints in the obtained R^A_{BC}, we construct required floor plans by addition of rooms. Suppose, we have given n rooms and the size of each room, and the problem is to automatically fit all the rooms inside a rectangle, without changing the size of each room, in such a way that the obtained rectangular composition should be best connected. In this case, most likely there will be some extra spaces inside the obtained floor plan. Therefore, our aim is to construct minimum area best connected rectangular floor plans, i.e., the best connected rectangular floor plans with minimum area for extra spaces.

Extended C-RLAB Algorithm

Let us consider that the position and size of room R_i drawn at stage i of the construction process are denoted by (x_i, y_i) and (ℓ_i, h_i) respectively. The position and size of rectangular composition $R^F_{BC}(i)$ obtained at stage i of the construction process be (X_i, Y_i) and (L_i, H_i) respectively. In the discussion that follows, n is the total number of given rooms to be allocated, i is the index of the room that is being considered for allocation, and k is the number of rooms successively arranged one above the other. The values of all (L_i, H_i) are computed as follows:

- For $2 \leq i \leq k$ or $(i - k) \equiv 3 \pmod 4$ or $(i - k) \equiv 0 \pmod 4$, we have

$$L_i = \max(L_{i-1}, \ell_i), \quad H_i = H_{i-1} + h_i$$

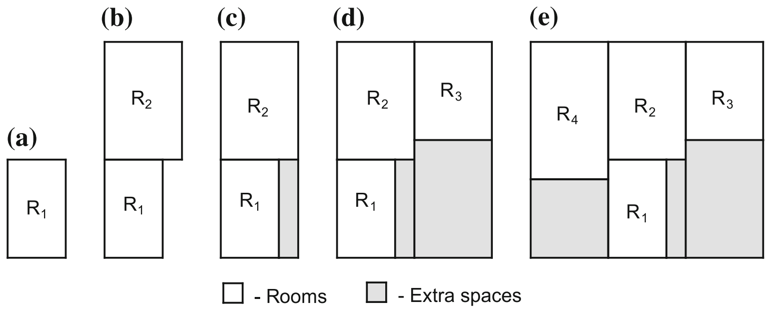


Fig. 7 A best connected rectangular floor plan with four rooms

- For $(i - k) \equiv 1 \pmod 4$ or $(i - k) \equiv 2 \pmod 4$, we have

$$\mathrm{L_i} = \mathrm{L_{i-1}} + \ell_\mathrm{i}, \mathrm{H_i} = \max(\mathrm{H_{i-1}}, \mathrm{h_i})$$

Here, $\mathrm{L_1} = \ell_1$ and $\mathrm{H_1} = \mathrm{h_1}$.

For n given rooms with their widths and heights, the dimensional constraints are introduced in a $R^A_{BC}(n)$ (constructed from the C-RLAB algorithm), by adding the following steps described below and depicted in Fig. 7 to the C-RLAB algorithm, to have an extended C-RLAB algorithm:

- Merge this step with the Steps 4, 6.3 and 6.4 of the C-RLAB algorithm
  After drawing each room, say $\mathrm{R_i}$, $i > 1$, if $\mathrm{L_{i-1}} > \ell_\mathrm{i}$, then draw an extra space $\mathrm{E_i}$ to the right of $\mathrm{R_i}$ with width $\mathrm{L_{i-1}} - \ell_\mathrm{i}$ and height $\mathrm{h_i}$ such that $\mathrm{E_i}$ is adjacent to $\mathrm{R_i}$ and its upper left vertex is the upper right vertex of $\mathrm{R_i}$. If $\mathrm{L_{i-1}} < \ell_\mathrm{i}$, draw an extra space $\mathrm{E_i}$ to the right of $R^F_{BC}(i-1)$ with width $\ell_\mathrm{i} - \mathrm{L_{i-1}}$ and height $\mathrm{H_{i-1}}$ such that $\mathrm{E_i}$ is adjacent to $R^F_{BC}(i-1)$ and its upper left vertex is the upper right vertex of $R^F_{BC}(i-1)$.
- Merge this step with the Steps 6.1 and 6.2 of the C-RLAB algorithm
  After drawing each room, say $\mathrm{R_i}$, if $\mathrm{H_{i-1}} > \mathrm{h_i}$, then draw an extra space $\mathrm{E_i}$ below $\mathrm{R_i}$ with width ℓ_i and height $\mathrm{H_{i-1}} - \mathrm{h_i}$ such that $\mathrm{E_i}$ is adjacent to $\mathrm{R_i}$ and its upper left vertex is the lower left vertex of $\mathrm{R_i}$. If $\mathrm{H_{i-1}} < \mathrm{h_i}$, draw an extra space $\mathrm{E_i}$ below $R^F_{BC}(i-1)$ with width $\mathrm{L_{i-1}}$ and height $\mathrm{h_i} - \mathrm{H_{i-1}}$ such that $\mathrm{E_i}$ is adjacent to $R^F_{BC}(i-1)$ and its upper left vertex is the lower left vertex of $R^F_{BC}(i-1)$.

It should be clear from the above steps that after positioning each room, our aim is to obtain a rectangular composition containing all positioned rooms; and to have this composition, sometimes an extra space is required. The idea of extra spaces guarantees that, we will have an automatically generated R^F_{BC} in the end, irrespective of the size of rooms.

For better understanding, consider an example, say Example 1, where we have been given four rooms with their widths and heights as follows (it should be noted that we are demonstrating an example for a very small value of *n* but this algorithm can generate floor plans for any value of *n*):

$$R_1{:}\,(10\times 15),\quad R_2{:}\,(12\times 18),\quad R_3{:}\,(12\times 16),\quad R_4{:}\,(16\times 20)$$

Through this example, we discuss the construction of a best connected rectangular floor plan, denoted by $R^F_{BC}(4)$, as illustrated in Fig. 7e. The steps for the construction of the required $R^F_{BC}(4)$ are as follows (for constructing $R^F_{BC}(4)$ using the extended C-RLAB algorithm, we are considering k = 2):

- Allocate R_1 with width 10 and height 15 at a position say (x, y) (see Fig. 7a).
- Draw R_2 above and adjacent to R_1 with width 12 and height 18 (see Fig. 7b). Since $\ell_2 > L_1$, to get a rectangular composition, draw an extra space to the right of R_1 (see Fig. 7c).
- Draw R_3 to the right of $R^F(2)$ and draw an extra space below R_3 (see Fig. 7d).
- Draw R_4 to the left of $R^F(3)$ and draw an extra space below R_4 (see Fig. 7e).
- Figure 7e is the required $R^F_{BC}(4)$ with area 1320.

Reducing the Area of Extra Spaces

From above example, we can see the presence of extra spaces inside the obtained $R^F_{BC}(4)$. To guarantee a minimum area in any $R^F_{BC}(n)$, we use the following two operations:

- Order of allocation
  The order in which rooms are arranged to obtain a rectangular floor plan using the extended C-RLAB algorithm is called the order of allocation. Clearly, for n rooms and for a fixed value of k, by considering each order of allocation, n! $R^F_{BC}(n)$ can be obtained and each $R^F_{BC}(n)$ may have a different area because of the presence of extra spaces.
- Swapping of width and height
  It is obvious to see that, we can position a room R_i with width ℓ_i and height h_i or with width h_i and height ℓ_i, i.e., we can swap the width and height of each room without changing the area of the room, which in turn produces two $R^F_{BC}(n)$ with different areas. If all the combinations of the rooms whose width and height can be swapped are taken into account, then we can obtain 2^n best connected rectangular floor plans using the extended C-RLAB algorithm for a fixed value of *k*.
- From the above two steps, we can obtain $n! \times 2^n$ $R^F_{BC}(n)$ using the extended C-RLAB algorithm for a fixed value of *k*. For the input of Example 1, we can generate $24 \times 16 = 384$ $R^F_{BC}(4)$ where each $R^F_{BC}(4)$ may have a different area. Now, to reduce the areas of extra spaces or to have a $R^F_{BC}(n)$ with minimum area among $n! \times 2^n$ $R^F_{BC}(n)$, we can pick the one with minimum area. By computing the areas of all $R^F_{BC}(4)$ for the input of Example 1, we found that the minimum and maximum areas are 968 and 1786 respectively. For the minimum area

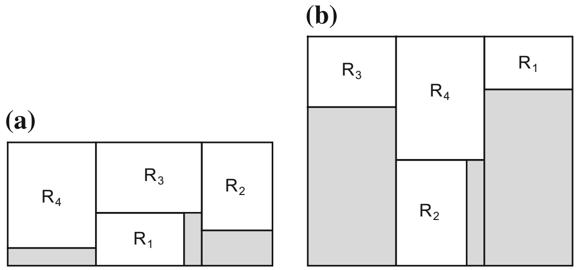


Fig. 8 The minimum and maximum area best connected rectangular floor plans with four rooms respectively

$R^F_{BC}(4)$, the order of allocation is R_1, R_3, R_2, R_4 where the width and height of R_1 and R_3 are swapped (see Fig. 8a). For the maximum area $R^F_{BC}(4)$, the order of allocation is R_2, R_4, R_1, R_3 where the width and height of R_1 and R_3 are swapped (see Fig. 8b).

Remark Using the definition of adjacency between rooms via extra spaces, provided in Shekhawat (2015b, Section 2.1), we can verify that the rectangular floor plans constructed using the extended C-RLAB algorithm are best connected.

Conclusion and Discussion

The current work is part of a larger work aimed at exploring exhaustively the use of rectangular arrangements in the design of layouts. We chose to work with rectangular arrangements because most architectural solutions still use this type of compositional schema. According to Steadman (2006) they do so because rectangular packing offers the best flexibility of dimensioning. Flexibility allows for any configurations of rectangles irrespective of their sizes; it also allows for the assignment of different dimensions to such configurations, while preserving their rectangularity. Our ultimate goal is to provide architects with design aids, that is, algorithms that can generate good candidate solutions, taking dimensional and topological requirements into account and that can be further improved and adjusted by them, to provide better solutions to the user. These aids would be particularly useful in design of large buildings with complex and specialized programs like hospitals, courthouses, and so on.

We know that the Fibonacci rectangle and the golden rectangle are among the best connected rectangular arrangements and that they have been used by many architects in their designs throughout the history. In this work, we proposed an algorithm called C-RLAB, which can generate $n - 3$ best connected rectangular arrangements. Each of these arrangements is geometrically different from the others in terms of the relative position of rooms and, consequently, in terms of the adjacency relations of each room with other rooms.

Since, the Fibonacci rectangle and the golden rectangle are best connected and they were used by many architects in their works, the arrangements generated after the proposed C-RLAB algorithm can also be used by architects for a similar reason, that is, to guarantee the generation of best connected floor plans. However, the kind of geometric diversity referred to above enables greater connectivity flexibility, which can be perceived as an advantage over the Golden- and the Fibonacci-based algorithms.

As a way of illustration, in Fig. 9 we present two new designs using the proposed C-RLAB algorithm based on the plan for Villa Rotonda. By considering the arrangement of rooms in Fig. 3, we can see that this plan was constructed using four Fibonacci rectangles (for details, refer to Shekhawat 2015a). Both the new designs are also constructed using four best connected rectangular arrangements, generated from the C-RLAB algorithm.

Here, we are not comparing our solutions with floor plans designed using the Fibonacci and the golden rectangles. For instance, we have not shown that floor plans generated by the C-RLAB algorithm are perceived as beautiful, the other reason pointed out by researchers for the use of the golden section by architects. In this work, we only tried to provide an additional algorithm to generate candidate best connected floor plans that can be modified by architects according to other design criteria they perceive as important, such as aesthetics, views, comfort, and so on.

In architectural design a multitude of aspects with different nature need to be considered. In this paper, we are dealing with functional requirements in the strict

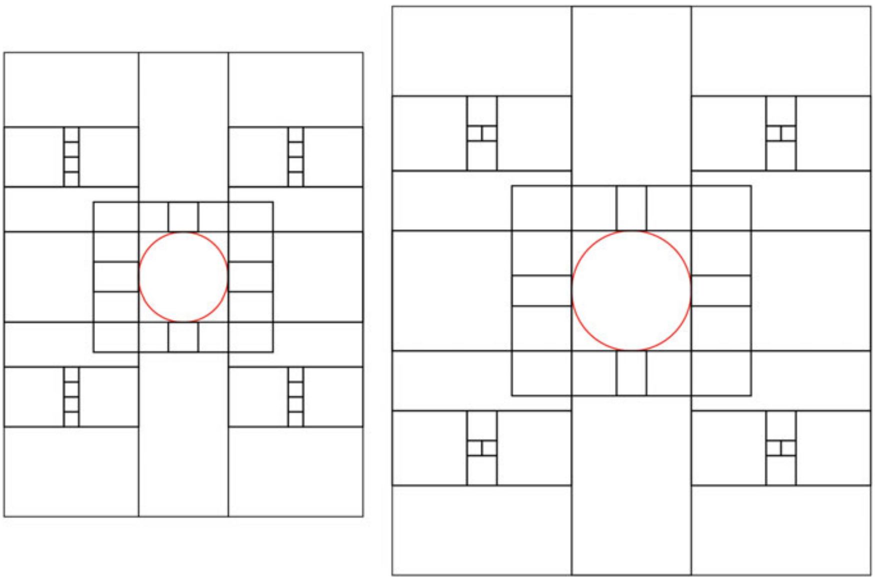


Fig. 9 New architectural designs with best connected rectangular arrangements constructed after the proposed C-RLAB algorithm

sense only, i.e., we are trying to provide an optimal solution while satisfying topological and dimensional requirements only. In the future, we will aim to cover other aspects.

Acknowledgements The research described in this paper evolved as part of the research project TECTON 3D funded by the Portuguese Foundation for Science and Technology (FCT) with grants PTDC/EEI-SII/3154/2012 and SFRH/BPD/102738/2014. This work also was partially funded by the Stuckeman Center for Design Computting (SCDC), Penn State University, United States and BITS Pilani, Pilani Campus, India.

References

Bartoli MT (2004) The sequence of Fibonacci and the Palazzo della Signoria in Florence. In: Williams K, Cepeda FD (eds) Nexus V: architecture and mathematics. Kim Williams Books, Fucecchio, pp 31–42

Baybars I (1982) The generation of floor plans with circulation spaces. Environ Plann B 9:445–456

Baybars I, Eastman CM (1980) Enumerating architectural arrangements by generating their underlying graphs. Environ Plan B 7:289–310

Bhasker J, Sahni S (1988) A Linear algorithm to find a rectangular dual of a planar triangulated graph. Algorithmica 3:247–278

Bloch CJ (1979) Catalogue of small rectangular plans. Environ Plann B 6(2):155–190

Bloch CJ, Krishnamurti R (1978) The counting of rectangular dissections. Environ Plan B 5 (2):207–214

Corinna R, Christopher AT (2002) Were the Fibonacci series and the golden section known in ancient Egypt? Hist Math 29:101–113

Earl CF (1977) A note on the generation of rectangular dissections. Environ Plann B 4(2):241–246

Fernandez-Llebrez J, Fran JM (2013) The church in the Hague by Aldo van Eyck: the presence of the Fibonacci numbers and the golden rectangle in the compositional scheme of the plan. Nexus Netw J 15(2):303–323

Flemming U (1978a) Wall representations of rectangular dissections and their use in automated space allocation. Environ Plann B 5(2):215–232

Flemming U (1978b) Representation and generation of rectangular dissections. In: IEEE 15th design automation conference, pp 138–144

Flemming U (1980) Wall representation of rectangular dissections: additional results. Environ Plann B 7(1978):247–251

Garson J (1970) A dual linear graph representation for space-filling location problems of the floor-planning type. In: Moore GT (ed) Emerging methods in environmental design and planning. MIT Press, Cambridge, pp 170–178

Green CD (1995) All that glitters: a review of psychological research on the aesthetics of the golden section. J Percept 24:937–968

Kalay YE (2004) Architecture's new media (Principles, theories, and methods of computer-aided design). The MIT Press, Cambridge

Kozminski K, Kinnen E (1984) An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits. In: 21st design automation conference, pp 655–656

Krishnamurti R, Roe PHON (1979) On the generation and enumeration of tessellation designs. Environ Plann B 6:191–260

Lai YT, Leinwand SM (1988) Algorithms for floorplan design via rectangular dualization. IEEE Trans Computer Aided Des 7(12):1278–1289

Levin P (1964) Use of graphs to decide the optimum layout of buildings. Archit J 140:809–815

March L, Steadman P (1971) The geometry of environment (An introduction to spatial organization in design). Methuen&Co Ltd, London

Mitchell WJ, Steadman JP, Liggett RS (1976) Synthesis and optimization of small rectangular floor plans. Environ Plann B 3(1):37–70

Park J, Lee HK (2009) The proportional design in Rudolph M. Schindler's Braxton-Shore house of 1930. J Asian Archit Build Eng 8:33–39

Rinsma I. (1987a) Existence theorems for floorplans. Ph.D. thesis University of Canterbury

Rinsma I (1987b) Nonexistence of a certain rectangular floorplan with specified areas and adjacency. Environ Plann B 14:163–166

Rinsma I (1988) Rectangular and orthogonal floorplans with required room areas and tree adjacency. Environ Plan 15:111–118

Robinson DF, Janjic I (1985) The constructability of floorplans with certain given outerplanar adjacency graph and room areas. In: Proceedings Xth British combinatorics conference Ars Combinatoria, vol 20B, pp 133–142

Roth J, Hashimshony R, Wachman A (1982) Turning a graph into a rectangular floor plan. Build Environ 17(3):163–173

Shekhawat K (2015a) Why golden rectangle is used so often by architects: a mathematical approach. Alexandria Eng J 54:213–222

Shekhawat K (2015b) Mathematical propositions associated with the connectivity of architectural designs. Ain Shams Eng J 6:795–802

Steadman P (2006) Why are most buildings rectangular? Arq Mag 10(2):119–130

Terzidis K (2007–08) AutoPLAN: a stochastic generator of architectural plans from a building program. Z Joint Study J 84–87

Walser H (2001) The golden section. The Mathematical Association of America, Washington

Wong SSY, Chan KCC (2009) EvoArch: an evolutionary algorithm for architectural layout design. Comput Aided Des 41(9):649–667

Individual Coffee Maker Design Using Graph-Based Design Languages

Claudia Tonhäuser and Stephan Rudolph

Abstract Graph-based design languages are used in this work to implement individualized mass customization. Using coffee machines as examples, the individualization of the product architecture and geometry is demonstrated. Combined with a user interface, a "*coffee maker language*" is shown to automate the design process including topological and parametric product variations by user inputs. The automatically generated processing result is an "individualized" creation of a digital coffee maker design which satisfies all implemented geometrical, physical and functional constraints.

Introduction

Market studies have identified a trend away from the "*same for all*" towards an increasing consumer product individualization. Manufacturing companies are thus faced with the task to respond to two conflicting goals: leaving mass the production originally introduced to reduce cost, they have now to figure out new ways of building much more individualized products at even lesser cost. New innovative methods of mass customization need therefore to combine the benefits of mass production with the single-item production ("*lot size 1*") customized to each customer individually.

Configuration Versus Re-design

Several kinds of implementations of mass customization are already in the market, e.g. web-based product configurators in the automotive industry. They allow to select additional functionalities from an interactive catalog of options. This means

C. Tonhäuser (✉) · S. Rudolph
University of Stuttgart, Stuttgart, Germany
e-mail: tonhaeuser@isd.uni-stuttgart.de

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_28

that the customer selects a single configuration out of a (potentially huge) number of allowable product permutations (Feldhausen and Grote 2013).

Truly individual product design according to individual customer needs is so far only known for rapid prototyping technologies, where customers upload their designs which are then manufactured and sent back to their home.[1] The current restriction to (simple) product configuration in the consumer industry has a good reason, since significant modifications of the topology or parameters may cause an avalanche of changes in the design. In full generality the design process has to be completely re-executed if significant topological or parametrical design changes occur (Kröplin and Rudolph 2005).

Configurators which offer already many benefits avoid this problem and ensure the manufacturability by only allowing product variants which have been pre-computed to stay within the feasible space. Evident examples of such product configurations are in the automotive industry e.g. choice of car colors, kind of seat cover and the type of radio or navigation system.

However, the customer's wish for product individualization doesn't care about the internal consequences of the requested design changes and may therefore affect both the product architecture and the product shape in its entirety. Mass customization needs therefore a complete digital model of the design process to manage the interplay of parametrical and topological design constraints and needs to provide a design representation which ensures the product realization within manufacturing and cost limits.

Design Synthesis Methods

Formal design synthesis methods for design automation have been investigated for at least two decades (Antonsson and Cagan 2005). In general, a classification can be made based on three different kinds of *string-based*, *shape-based* and *graph-based* representations. L-Systems (Lindenmayer 1967) from mathematical biology are an example of a string-based representation. L-Systems are a formal language consisting of a rule-based grammar defined by an alphabet, an axiom and (at least one single or) several rules which form a production system.

L-Systems belong to the class of so-called *rewriting systems* in the field of computer science (Alber and Rudolph 2002). After execution, the resulting symbol string is translated into geometry using a "*turtle graphic*" (Prusinkiewicz and Lindenmayer 1996). L-Systems have been successfully used to model biological structures with tree-like architectures such as plants, trees or sea shells (Prusinkiewicz and Lindenmayer 1996; Meinhardt 2009). This inbuilt tree-like product architecture is however also the main disadvantage, since the limitation to

[1]www.shapeways.com.

cycle-free product architectures is not really compatible with the needs of modern product architectures in complex engineering system designs.

Shape grammars are also a rule-based approach, but use a direct shape-based representation to generate geometry. They have been used to generate 2-dimensional (Stiny and Gips 1972) or 3-dimensional (Agarwal and Cagan 1998; Hoisl and Shea 2010) objects and have been applied in architecture (Stiny and Gips 1972) and engineering (Agrawal and Cagan 1997). While direct geometry representation may be an advantage to generate geometry, it may turn out as a severe disadvantage for modelling any other non-geometric design aspect.

Graph-based design languages inherit many benefits of string-based and shape-based grammars without most of their drawbacks (Kröplin and Rudolph 2005). Graph-based representations enable the encoding of arbitrary relations between nodes and allow the creation of topological ring structures (Kröplin and Rudolph 2005). Especially for explicit topological modeling tasks, graph-based techniques seem thus to possess a natural advantage over a string-based or shape-based grammar approach.

Graph-grammars have been used for the synthesis of function structures (Bryant et al. 2005; Kurtoglu and Campbell 2009) and the addition of functional and mathematical constraints to the product geometry (Kröplin and Rudolph 2005; Schmidt and Cagan 1997; Schmidt et al. 2000). Finally, the combination of the abstract graph-based design representation together with a graphical rule-based transformation scheme has led to the development of the design compiler 43 (IILS mbH 2015). Design languages have been used to design space stations (Irani and Rudolph 2003, 2005), satellites (Schaefer and Rudolph 2005a, b), aircrafts (Kormeier et al. 2003), airships (Rudolph and Boelling 2004) and space frames (Haq and Rudolph 2004). Design languages itself are represented in the UML standard, which guarantees an open and vendor-independent knowledge representation (Reichwein 2011). Recent works cover digital factory integration (Arnold and Rudolph 2012), wire harness of aircraft cabins (Landes and Rudolph 2011; Landes 2013) and exhaust treatment systems (Vogel et al. 2010, 2012).

Focus and Paper Outline

Taking up the classic coffee maker of Agarwal and Cagan (1998) as reference, the subject of the paper is to demonstrate a graph-based design language for the individualized generation of coffee machines. This design language encodes the complete design process and stores all relevant product information in one single abstract meta-model in form of the design graph. This meta-model ensures model consistency and enables incremental alteration of the product architecture and product parameters along the design process. The design language includes the mapping of the product requirements into a function structure and the transformation of the function structure via solution principles into components forming the product structure according to the "systematic (German) design methodology" by Pahl and Beitz (Feldhausen and Grote 2013).

This graph-based "*coffee maker language*" is intended to push the limits towards a better coverage of the aforementioned problems in the generation of new "*individualized*" mass product customization by serving as adequate digital model for the re-execution of the underlying complex product design process. Letting the user influence the design requires to prevent him from doing "*wrong things*". Therefore, a set of "*hidden rules*" to ensure consistency is needed (and presented in detail in "Graphical User Interface (GUI)" section).

To examine our method, an insight into the graph-based design languages approach is given first (see "Graph-Based Design Languages" section) and the design methodology of Pahl and Beitz is briefly reviewed (see "Pahl–Beitz Design Methodology" section). Afterwards the coffee maker is investigated as an example for the digital design process (see "A Graph-Based Design Language for Coffee Makers" section) and a graph-based coffee maker language is presented including all its different aspects (see "Implementation of the "Coffee Maker Language"" section). In a last step, the user interface is described (see "Graphical User Interface (GUI)" section) and finally a variety of generated coffee machines is shown before concluding (see "Results" section).

Graph-Based Design Languages

Graph-based design languages are inspired by natural languages, since both consist of vocabulary and rules that build up a grammar. For representation purposes the Unified Modelling Language (UML) (2004) provides a worked out formal syntax. Graph-based design languages use mainly two types of UML diagrams. The UML class diagram represents the domain background knowledge in form of an ontology of the used domain vocabulary. The UML activity diagram represents the sequence of design rules which encodes the complete decision making during the design process.

In UML class diagrams, the engineering design vocabulary is modeled by classes which may have attributes or may be linked to each other to express inheritance (reads as "*is_a*") and various types of associations (reads as "*has_a*"). The class diagram supports an in- or decreasing degree of abstraction, improves modularity and fosters design concept reusability.

The UML activity diagram contains all design rules, their sequence and other program control structures such as decision nodes for modeling all the conditional design decisions and program control structures known from string-based programming languages (e.g. *if…then*, *for*, *while*) for loops and branches.

A rule consists of a two-quadrant scheme with a conditional "*if*"-part (i.e. the left hand side, LHS) to leave empty or to search a specific instance to be modified and a generative "*then*"-part (i.e. the right hand side, RHS) to create new instances and add links or to modify existing ones.

The creation of a design language is a non-trivial task which may, on the one hand, require upfront a quite high time invest. On the other hand, the execution of a design language is achieved downstream by using a design compiler such as the "*design compiler 43*" (IILS mbH 2015) at a Giga-Hertz-rate, thus reducing the actual execution time of a full product design cycle down to the theoretical limit of the addition of the algorithm runtimes (Groß 2014). These relative advantages and disadvantages of "*design compilers*" have long been discussed controversially (Antonsson and Whitney 1997; Whitney 1996) since their first appearance in the field of mechanical design (Ward 1988; Ward and Seering 1989, 1993) in the late eighties.

During execution of the production system, the "*design compiler 43*" (IILS mbH 2015) generates a design graph. This design graph represents the complete digital model of the created system. In this way, all relevant product information is stored in one single meta-model, thus guaranteeing model consistency for subsequent model-to-text (M2T) transformations which map the abstract UML design representation to external analysis tools (i.e. CAD-kernels for geometry generation, CFD-, MBS- and FEM-solvers for computational fluid dynamics, multi-body and finite element analysis, etc.). The simulation results can be fed back into the design model in order "*to close the design loop*", thus supporting an iterative design style. The design language information and processing workflow is shown in Fig. 1.

Pahl–Beitz Design Methodology

The "*systematic (German) design methodology*" by Pahl and Beitz (Feldhausen and Grote 2013) aims at providing a generally applicable methodology for conceptual and detailed design of technical systems and comprises a collection of various design methods. These include experience-based [e.g. creativity or design patterns (Alexander et al. 1977)] and systematic approaches (e.g. switching between synthesis and analysis of objectives and solution) and has become a german standard

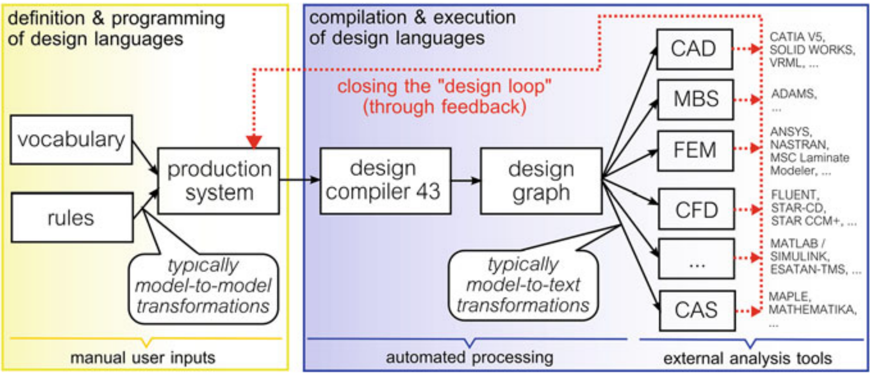


Fig. 1 Schematic workflow of design languages

(Verein Dt. Ingenieure 2221 1993). It summarizes the product design process into several steps (Feldhausen and Grote 2013; Verein Dt. Ingenieure 2221 1993):

1. Clarifying and specifying the task inclusive the requirements.
2. Determining the functions and their structures.
3. Search for solution principles and their structures.
4. Dividing into realizable modules.
5. Design of these modules.
6. Design of the entire product.

In the early phases [points (1–4)], it is decided on the creation of a product architecture including all preliminary considerations. This results in a rough conceptual overview of the product. In order to achieve this, requirements have to be identified and mapped into abstract product functions forming a function structure. Subsequently, this function structure is transformed via solution principles into individual components building up a product structure. Since functional product decomposition can cause new requirements or additional functions, product design is an iterative process which is segmented into several spaces with mapping steps in between.

The resulting product structure depends on "*individual*" decisions and compromises concerning requirements, functions, solution principles and components (Feldhausen and Grote 2013). Figure 2 shows the product architecture of a coffee maker.

Requirements which affect the product shape require special treatment since they directly translate into geometry. Shape information on product and parts must therefore be taken into account in the subsequent phases of the design process (5, 6). In our approach, the final product structure and shape depend on the user inputs and find their expression in the choice of the casing topology and parameters as discussed in Sect. 2.

A Graph-Based Design Language for Coffee Makers

The coffee maker (Agarwal and Cagan 1998) design example seems to be based always on the very same functional principles (at least for drip coffee machines). With its basic, almost "*invariant*" functionality, it is implemented into a graph-based design language which is used here as a simple example. Design languages may however also handle design tasks with higher complexity. The satellite design language of Gross (Gross and Rudolph 2015) covers most of the satellite design cycle as documented in the *SMAD book* (Wertz 1999) of about *1000 pages* which illustrates thus successful complexity handling.

In coffee makers, water has to be poured firstly into a container where it is stored until the coffee maker is turned on. Secondly, the water flows through an inlet hose with a back pressure valve into the heating element of the hotplate where it turns to

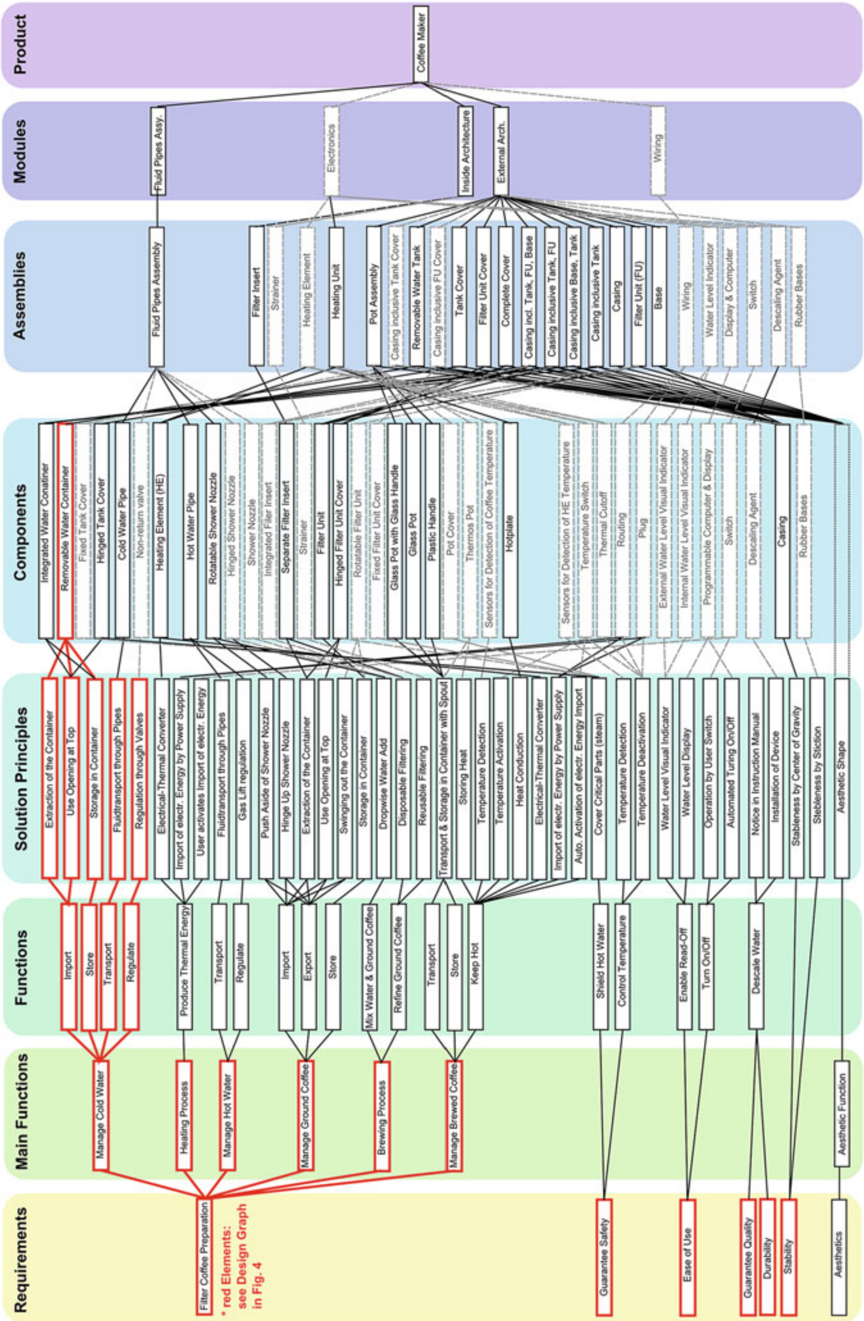


Fig. 2 Product architecture of a drip coffee maker

steam, building up pressure which closes the back pressure valve. Thirdly, the steam rises through a tube upwards, expands at the top, condenses and drips into a filter, where the hot water mixes with the coffee grounds and trickles as coffee into a pot below due to gravity. Meanwhile the pressure in the heating element is falling until the valve opens again and more water can flow through it. Finally, when all cold water is used up, the increasing temperature in the heating element is detected by a sensor which either turns off the coffee machine automatically or regulates the temperature of the hotplate to keep the coffee warm.

Agarwal and Cagan (1998) already stated that the design functionality drives some requirements for topology and shape. Firstly, the heating element must be placed below the water storage container (in order to flow). Secondly, the filter has to be above the heater (but low enough to prevent the water to condense prematurely). Thirdly, the coffee pot has to be below the filter, and, fourthly, the heating element has to be located below the pot.

The functional constraints involve fluid pipes which have to be positioned between the water storage container, the heating element and a shower nozzle above the filter. Aside from functional requirements, the physical constraints include the volume of the pot, which has to be nearly the same as the volume of the water container. Similarly, the size of the filter has to be compatible with the required amount of ground coffee resulting from the water volume. The capacity of the hotplate must match the dimensions of the pot bottom to keep the brewed coffee warm. It is obvious that physical constraints merge into the geometric ones, thus affecting the volume or dimensions of all components which have to fit into the product casing.

Lastly, the outer product shape has to be designed around the functional components. Individualized mass customization as interpreted in this work focuses on the product appearance. Due to the fact that possible user inputs are intercoupled with the initial geometric, functional and physical requirements, the design of a coffee maker is governed by both of the two design paradigms: the "*form follows function*" and the opposite "*function follows form*" (Denzin and Lincoln 2003). The following points should be considered:

- The topological arrangement of functional components (water container, heating element, fluid pipes, shower nozzle, filter and pot) has to be predefined excluding the exact dimensions or positions.
- Since the product architecture possibly affects the initial product requirements it should be selected in a first step by the customer.
- The processing of customer inputs requires a user interface.
- The restricting shape requirements, which are driven by the basic functionality and architectural changes, have to be secured during the whole processing of the user inputs.
- The exact dimensions and positions of the basic functional components must be automatically adapted during the processing of the user inputs.

Implementation of the "Coffee Maker Language"

The coffee maker language in this work consists of two parts. The first part empowers a user to choose between several product architectures and to manipulate the shape of the product. A simple user interface accepts inputs and ensures the compliance with some restricting constraints running in the background. The second part is the design language for coffee makers which encodes the whole design process and can be (re-)executed again to generate a complete digital product model out of the user inputs. In the next two sections, the vocabulary and rules this coffee maker design language are discussed along with some of the background design constraints.

Vocabulary

The first mental step to build up a graph-based design language is a finite conceptual product decomposition into a vocabulary. This decomposition is later the basis for the aggregation of the vocabulary into a product architecture according to the "systematic (German) design methodology" (Feldhausen and Grote 2013).

In the coffee maker case it is useful to differentiate between functional components and casing, which in turn is partitioned into different topological regions. These regions are governed by the basic functional components and are partitioned into a base, a water storage unit, a filter unit, a pot and covers for the water storage and the filter unit. Some of the topological casing elements may be combined, such as the base or the filter that can merge with the water tank unit into a "lower" or "upper" section. It is also possible to convert the base, the water tank and the filter into one "*complete casing*" or to make one "*complete cover*" out of two. This casing partition specifies whether parts are topologically independent or possibly combined with each other and aims at creating a certain variety of the aesthetic issue since it is a key aspect of individualization. Furthermore, a removable water container is optional.

One of the implemented functional components is a fluid pipe unit consisting of a pipe between the water storage container and the heating element, a pipe between the heating element and the shower nozzle and the shower nozzle above the filter. Other components are: the filter holder and the heating unit containing a water heating element and a hotplate to keep the brewed coffee warm. Since all possible product architectures have to be represented by the vocabulary, the class diagram includes all variants of the casing and functional components.

For the representation of geometry, a design language offers a large variety of geometric elements and operations. All parts that need a geometric representation must have a generalization to the *Component* class from the geometry class diagram. This is represented in the class diagram via a shortcut. It means that all these coffee maker classes are *Components* and have therefore a geometric representation. In Fig. 3, the class diagram of our coffee maker vocabulary is shown including all

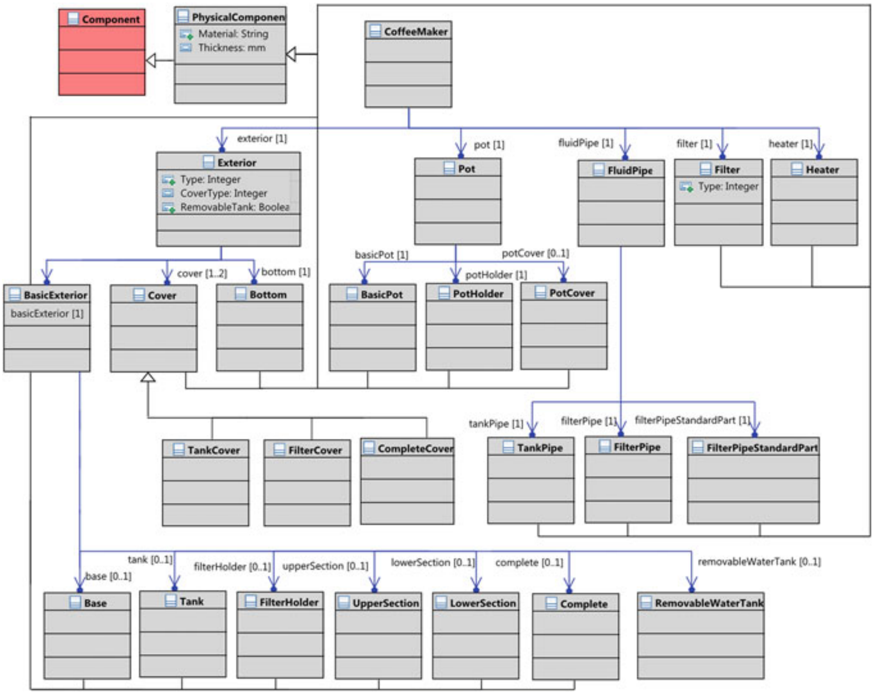


Fig. 3 The class diagram of the coffee maker vocabulary

connections and the *Component* shortcut. For the vocabulary of the product architecture, all requirements, functions, solution principles and functional components have to be modeled as classes in a product architecture class diagram.

Grammar

To represent the design process activities, a graph-based design language may consist of graphical rules and rules written in Java.[2] The rule execution sequence can be static or can be modified during compilation by means of control structures such as the aforementioned decision nodes.

Since the creation of a product architecture comes first in the design process, the rules which map requirements into functions and transform them via solution principles into functional components have to be modeled first. Executing this rule

[2]The design compiler 43 (IILS mbH 2015) for graph-based design languages offers an editor for graph-based M2M-transformations and accepts also the use of string-based programming languages such as Java (Java SE from ORACLE. www.oracle.com) and OCL (Object Management Group, Object Constraint Language. www.omg.org) constraints.

sequence generates the product architecture. By means of a dedicated model view[3] the design graph may display a similar structure than the intended product architecture.

Figure 4 exemplarily illustrates the transformation of one product architecture. At the top, an extract from the program implementing the rules is given. The rules are depicted below in more detail. On the left hand side, the conditional part is pictured, whereas the generative part can be seen in the middle. On the right hand side the expanding design graph is presented. The first rule (Rule: *Requirements*) instantiates six requirements shaded in gray. In the following rule (Rule: *Main Functions*) the requirement object "*Filter Coffee Preparation*" is searched and mapped onto the right hand side (colored in yellow) and the six related main function objects are instantiated and attached to it. The last rule (Rule: *Connect To Components*) shows the coupling of three solution principles to one resulting component. For further clarity all the elements contained in the illustrated path are additionally highlighted in Fig. 2.

The rules generating the front part of the product architecture (requirements, functions, solution principles) are implemented as static, since this part is elaborated more extensively resulting in some functions that are not transformed into components which may be completed in a later design stage. The dynamic adaption of the product architecture based on the user is applied on the rear part modifying the subsequent levels (components, modules etc.).

The digital product design process of the coffee maker geometry starts with an axiom (implemented as a rule with an empty conditional part and instantiating objects in the generative part) generating the first casing instances selected by the user by means of a simplified interface described in a later stage in "Graphical User Interface (GUI)" section.

Focusing on the geometric arrangement and parameters, the subsequent rules gradually build up the geometric representation of all casing parts. Since the abstract geometric operations are related to the specific ones in the CAD program, the process is similar to the procedure engineers follow when using CAD tools manually. First, the points of each casing element have to be generated and then connected with splines forming profiles and guidelines for each shape, thus resulting finally in a wireframe model.

The number of points and their positions, as well as the shape of the splines are defined by the user during the user input procedure explained in more detail in Sect. 3.3. The user thereby specifies for each topological element of the casing successively the top profile, the bottom profile, the side profile in the *XZ*-plane and the side profile in the *YZ*-plane. Subsequently, a shell is generated which is thickened to become a volume. The volumes then are trimmed, so that their

[3]The design compiler 43 (IILS mbH 2015) for graph-based design languages offers means for customized model views and massive debugging features for the analysis of the internal states of design language attributes during compilation.

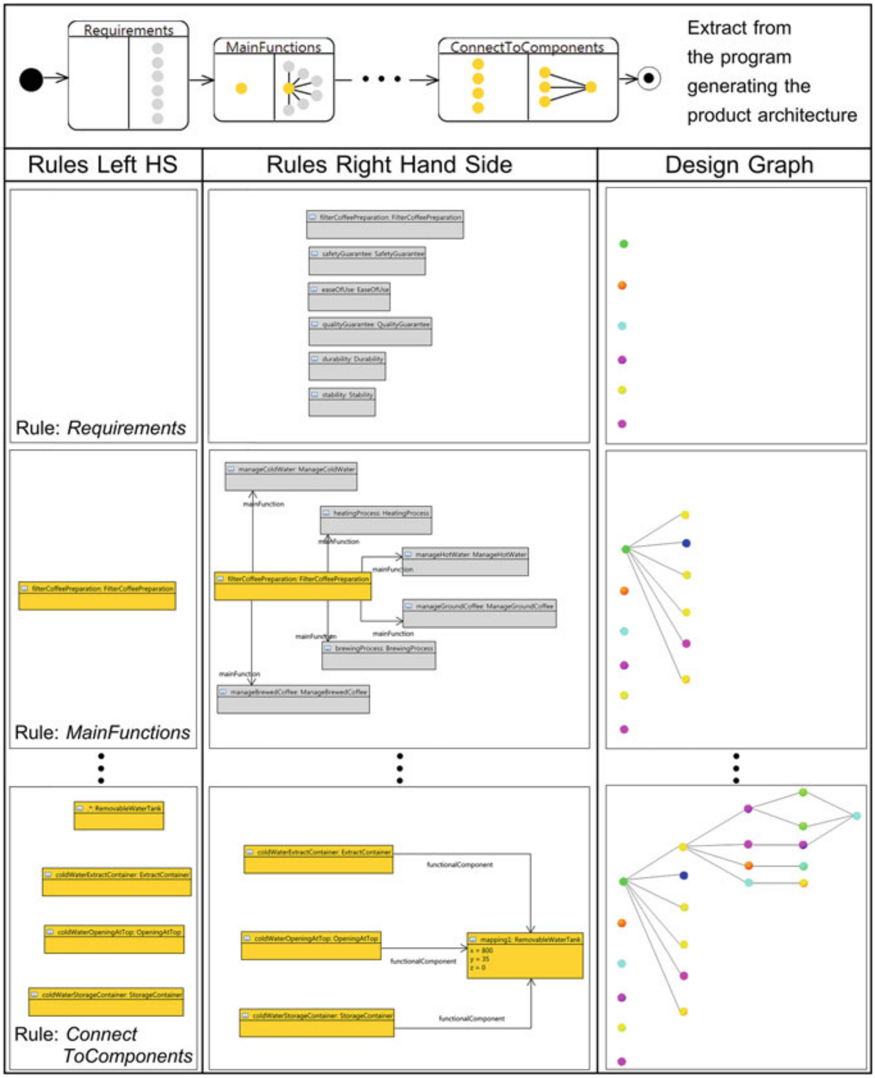


Fig. 4 Requirements transformation process of some exemplary details

assembly builds up an entire coffee maker casing. The outcome of the trimming procedure herein depends on the chosen casing architecture.

To comply with all constraints, a sequence of rules exists for every (implemented) possible product structure, resulting in a program with various branches. Decision nodes provide the design compiler with the required information for choosing the correct branch based on the preceeding user choices. Figure 5 illustrates the automated decision making during the program execution concerning the cover topology.

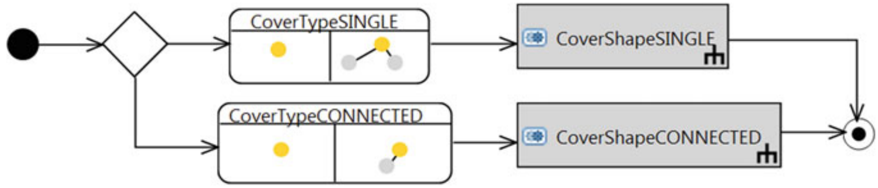


Fig. 5 Decision nodes affecting the cover topology

While the design compiler executes the production system, the coffee maker design graph is generated and iteratively modified. Since the whole digital design process consists of 96 vocabularies, 204 rules and 30 hierarchically nested production sequences, only a small part of the procedure is presented to illustrate the method. Figure 6 shows the generation of the filter geometry whereas Fig. 7 shows some exemplary extracts of the corresponding rule sequence (only generative part, yellow color indicates the searching and mapping of already existing objects). The corresponding process steps are marked with the same numbers.

Of course, the interface for the user input has to be called in the beginning. Partly this interface is implemented using Java code and Java rules which are integrated into the graphical design rule sequence execution.

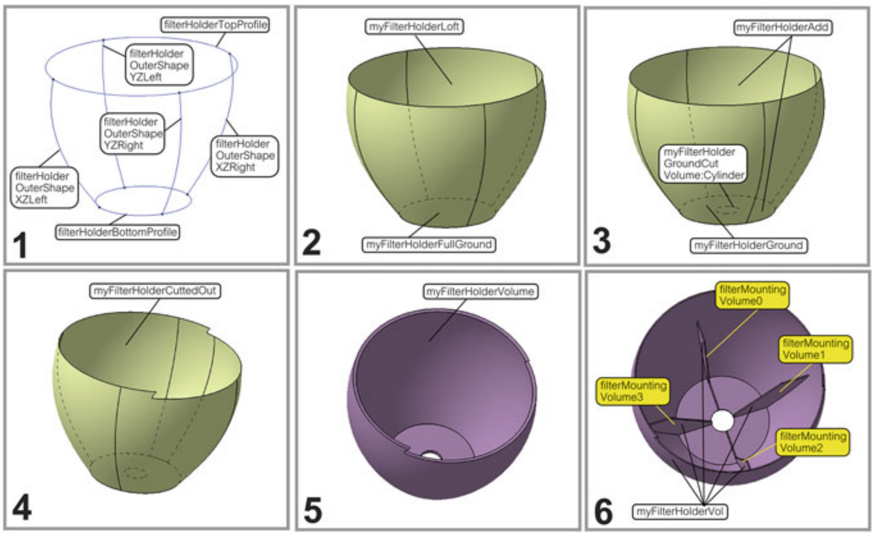


Fig. 6 Generation steps of the filter unit geometry. *1* Generating the wireframe model. *2* Generating the shapes: loft and plane. *3* Generating the outlet and joining the plane with the loft. *4* Creating the cutting (for the shower nozzle). *5* Surface thickening and volume creation. *6* Addition of filter mountings to filter

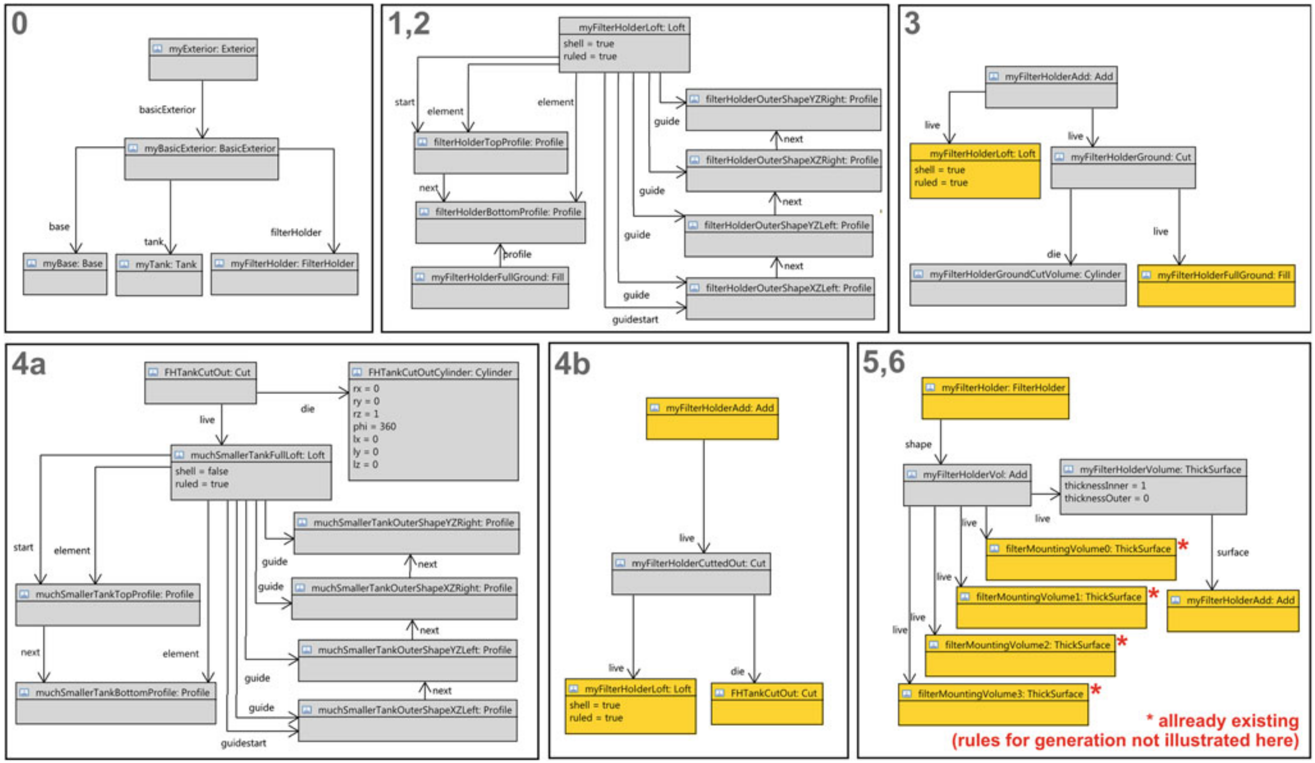


Fig. 7 Rules generating the filter unit. Functions of rules *1* to *6* see Fig. 6. Rule *0* creates whole casing structure

Graphical User Interface (GUI)

The user interface is partitioned into three subroutines, each intended to process one main topic of the product design. Initially the desired topology is queried followed by the shape input and simultaneous calculation of all functional components, finally terminated by an input window for material selection not elaborated in more detail in this work.

Variety of Product Architecture

The variety of the topology is closely linked to the partition of the casing but includes also the integration of a removable water container. In consequence, users choose between four different casing partitions, two cover topologies and an option for a detachable water tank (see Fig. 8).

Shape and Parametrization of Components

The shape of the casing is defined by several 2-dimensional views modifying the profiles of the casing's subparts (see Fig. 9). Pot, base, filter, tank, covers and pot handle are specified going through the top, bottom and side profile (in the XZ- and YZ-plane). All profiles consist of points and splines connecting them. The user can add, remove or move points and can alter the shape of the splines by modifying their tangent direction or stiffness (shown as red points connected by dashed lines). Predefined points are colored in grey and determine the minimal amount of points while additional points are colored in blue. In the view the user only sees the proportions and relative arrangements of the casing parts, the real dimensions are calculated automatically. In order to do this, the user is asked to select the pot volume (by choosing a number of cups), which is used as sizing factor.

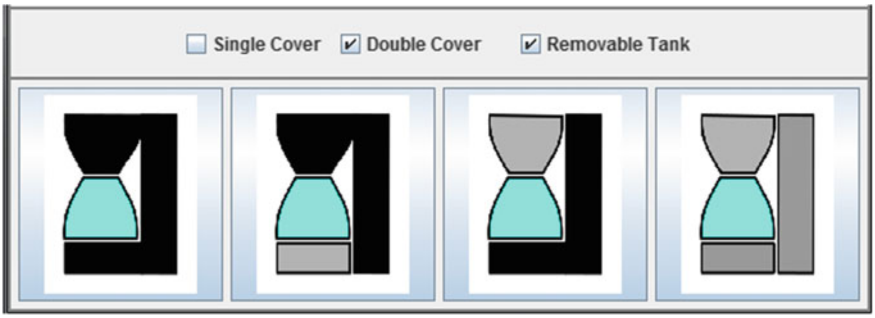


Fig. 8 Pop-up window for product topology selection

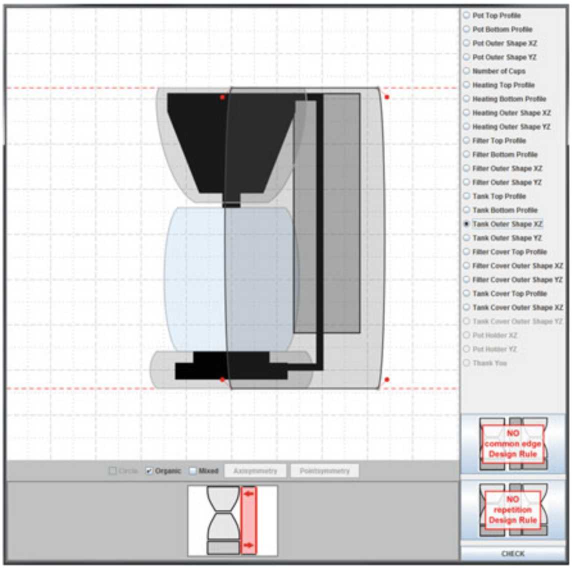


Fig. 9 Water tank: XZ-profile

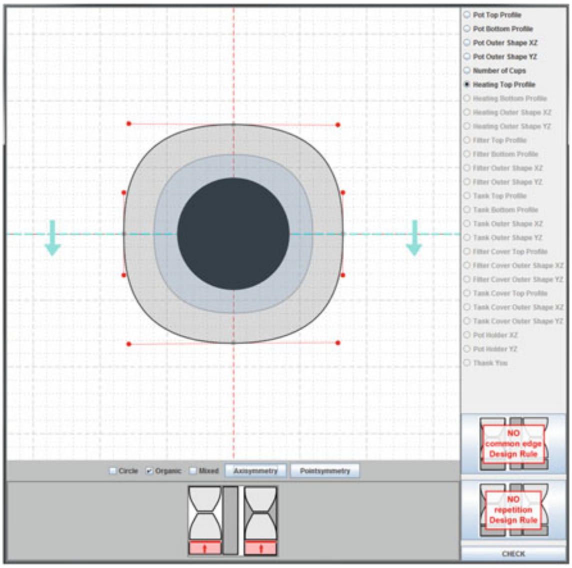


Fig. 10 Axisymmetric Profile

Once the shape of the pot and its volume are determined, the remaining parts are built around the pot's shape including all constraints. To make operation easier, points can be mirrored around the X-axis or around the X- and the Y-/Z-axis to create axisymmetric or point-symmetric profiles (see Fig. 10). Moreover, the user is

Fig. 11 Generated CAD-model including all functional components

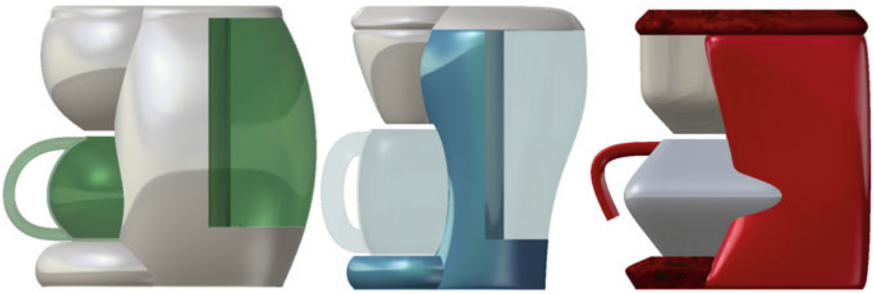


Fig. 12 Variety of generated CAD-models

guided through the steps by means of additional icons which highlight the current profile that is processed.

Functional components which include constraints for the shape are displayed as black figures. The contours have to be arranged around these components (heating unit, filter, water tank) while other functional components (fluid pipes, the shower nozzle) are completely adjusted to the silhouette of the casing. To demonstrate the use of standard parts, the filter and shower nozzle are selected from a catalog. The calculation methods used for the functional components are described briefly in the following:

- Heating Unit (incl. Hot Plate and Heating Element): It is represented as cylinder in the GUI. The radius and height are computed calculating the radius of the maximum circle contained in the bottom profile of the pot.
- Filter: Based on standard parts three different filter dimensions are predefined. Depending on the chosen volume the filter is computed.
- Water Tank: The indicated water container restricts the minimal casing dimensions. Its general shape is predetermined, but the width is calculated from the pot's volume. The height is computed by the predefined space for the electronics assembly planned to be mounted on the bottom.

- Shower Nozzle: The shower nozzle is computed with discrete lengths in steps of 10 mm. This method represents the use of standard parts.
- Fluid Pipes: The top and bottom attachment positions are calculated from the heating unit und filter positions. The spine is further computed from the maximum dimension in X-direction of the pot, base or filter.

Results

This section shows some examples of generated CAD models of the coffee maker language in Fig. 11. In Fig. 12 examples of the possible diversity in the generated models are shown. They include parametric and topological alterations such as the integration of the removable water container.

Discussion

Due to the representation of the whole design process and the automated execution, graph-based design languages tolerate topological as well as parametrical changes. Through offering a choice of four casings and two cover topologies and an optional removable tank, the robustness of the encoded design process is shown. The topological modifications are limited to the implemented topological switches (kind of housings, covers, tank), but as the model can be expanded further by additional topological rules, the number of feasible topologies may be increased.

The reference to the shape grammar approach by Agarwal and Cagan (1998) does not aim at comparing the performance of both methods in the design of coffee makers. Our intention is to illustrate an alternative approach which overcomes the disadvantage of shape grammars in the handling of non-geometric requirements, functions and solution principles.

Each decision during the design process has consequences on the further procedure. Initially all options are available, but through progressive decision making, the set of possible solutions is decreasing. In the coffee maker example the choice of the pot volume affects the shaping result and the size of the remaining casing parts. This applies for the automated choice of the filter size and the resulting restrictions on the shape of the filter unit as well. The compliance with the shrinking solution space is ensured by creating a suitable set of "*hidden rules*" for the user. In this respect, the potential of design languages for the development of user-interfaces becomes apparent. Apart the geometric and physical functions, the aesthetic aspects are left completely up to the user. Such a strategy is based on the premise for the intended "*true individualization*". The paper therefore implicitly and explicitly avoids any definition or claim concerning "*aesthetics*".

The approach offers possibilities for further features. For instance, the price of the product could be calculated straightforward from the choice of the material and the required amount of it (calculated from the wall thickness and the user-defined shape of the housing). Moreover, specifying a desired heating time, the required power of the heating element could be computed and a suitable heater could be found in a catalogue. The derived costs could also be taken into account. In this respect, one further benefit of design languages becomes apparent, that is the possibility to implement almost any arbitrary non-geometric information processing algorithm.

Constraints for manufacturing aren't included yet but can be considered in two ways. Either the presented approach could be combined with rapid manufacturing techniques, or missing manufacturing constraints could be explicitly incorporated. Especially the latter option may affect the user interface, since the set of hidden rules has to be complemented or extended.

In terms of "*modeling effort*" the total project duration was 6 months at experienced expert level. This includes the definition and programming of the coffee maker design language (the yellow part in Fig. 1) in the "*design compiler 43*" (IILS mbH 2015) framework. The design compiler itself includes all the necessary software interfaces (the blue part in Fig. 1).

Conclusion

The successful generation of a wide range of coffee maker CAD models demonstrates that the graph-based design language approach can serve as a generic model for the design process of coffee makers. Combined with the individual user inputs, a single-item production with "*lot size 1*" using rapid manufacturing techniques could be completely automated.

Acknowledgements The authors thank the three anonymous reviewers for their thoughtful comments which helped to improve the quality of the paper. Part of this work of the first author was supported by a grant of the state of *Baden-Württemberg* in the context of the *ZAFH*-Project (grant no. 43031423).

References

Agrawal M, Cagan J (1997) Shape grammars and their languages—a methodology for product design and product representation. In: Proceedings of ASME design engineering technology conference, Sacramento

Agarwal M, Cagan J (1998) A blend of different tastes: the language of coffee makers. Environ Plan 25(2):205–226

Alber R, Rudolph S (2002) On a grammar-based design language that supports automated design generation and creativity. In: 5th IFIP WG5.2 workshop on knowledge intensive CAD, Malta

Alexander C, Ishikawa S, Silverstein M (1977) A pattern language. Oxford University Press, New York

Antonsson EK, Cagan J (2005) Formal engineering design synthesis. Cambridge University Press, Cambridge

Antonsson EK, Whitney DL (1997) Correspondence on the potential for mechanical design compilation. Res Eng Des 9(4):246–247

Arnold P, Rudolph S (2012) Bridging the gap between product design and product manufacturing by means of graph-based design languages. In: 9th international symposium on tools and methods of competitive engineerings, TMCE 2012, Karlsruhe

Bryant C, Stone R, McAdams D, Kurtoglu T, Campbell M (2005) A computational technique for concept generation. In: Proceedings of ASME design engineering technology conference

Denzin NK, Lincoln YS (2003) The landscape of qualitative research: theories and issues, 2nd edn. Sage, Thousand Oaks

Feldhausen J, Grote KH (2013) Pahl/Beitz Konstruktionslehre—Methoden und Anwendung erfolgreicher Produktentwicklung. Springer, Berlin

Groß J (2014) Aufbau und Einsatz von Entwurfssprachen zur Auslegung von Satelliten, Dissertation, University of Stuttgart

Gross J, Rudolph S (2015) Modeling graph-based satellite design languages, Aerosp Sci Technol

Haq M, Rudolph S (2004) EWS-Car: a design language for conceptual car design, VDI-Berichte 1846. In: Conference on numerical analysis and simulation in vehicle engineering, Würzburg, Germany, pp 213–237

Hoisl C, Shea K (2010) Interactive, visual 3D spatial grammars, design cognition and computing (DCC)

IILS mbH (2015) The Design Compiler 43. www.iils.de, last visited: 30.11.15, Ingenieurgesellschaft für intelligente Lösungen und Systeme mbH

Irani M, Rudolph S (2003) Design grammars for conceptual designs of space stations. In: Proceedings of IAC international astronautical congress, IAC-03-T.P.02, Bremen

Irani M, Rudolph S (2005) Space station design rules. SAE Aerosp Eng Mag 10(2005):43–45

Java SE from ORACLE. www.oracle.com

Kormeier T, Alber R, Rudolph S (2003) Topological and parametrical design of aircraft surfaces using design grammars. In: Proceedings of DGLR symposium, München

Kröplin BH, Rudolph S (2005) Entwurfsgrammatiken—Ein Paradigmenwechsel? Der Prüfingenieur, pp 34–43

Kurtoglu T, Campbell M (2009) Automated synthesis of electromechanical design configurations from empirical analysis of function to form mapping. J Eng Des 20(1):83–104

Landes B (2013) On physical aspects of cabin architectures using tolerancing methods, Dissertation, University of Stuttgart

Landes B, Rudolph S (2011) Aircraft cabin architectures including tolerancing using a graph-based design language in UML. In: Proceedings of DGLR symposium, Bremen

Lindenmayer A (1967) Mathematical models for cellular interaction in development, Parts 1 and 2. J Theor Biol 18(1968):280–315

Meinhardt H (2009) The algorithmic beauty of sea shells. Springer, Berlin

Object Management Group, Object Constraint Language. www.omg.org

Prusinkiewicz P, Lindenmayer A (1996) The algorithmic beauty of plants. Springer, Berlin

Reichwein A (2011) Application-specific UML profiles for multidisciplinary product data integration. Dissertation, University of Stuttgart

Rudolph S, Boelling M (2004) Constraint-based conceptual design and automated sensitivity analysis for airship concepts. Aerosp Sci Technol 8, 333 ff

Schaefer J, Rudolph S (2005a) Satellite design by design grammars. Aerosp Sci Technol 9 (2005):81–91

Schaefer J, Rudolph S (2005b) Spacecraft thermal model generation by design compilation. Deutscher Luft-und Raumfahrtkongress, Friedrichshafen

Schmidt L, Cagan J (1997) GGREADA: a graph grammar-based machine design algorithm. Res Eng Des 9(1997):195–213

Schmidt L, Shetty H, Chase S (2000) A graph grammar approach to mechanism synthesis. J Mech Des 122(2000):371–376

Stiny G, Gips J (1972) Shape grammars and the generative specification of painting and sculpture. In: Proceedings of IFIP congress 1971, Amsterdam

Unified Modeling Language (2004) Superstructure, version 2.0, of the Object Management Group (OMG) at http://www.omg.org/cgi-bin/doc?formal/05-07-04

Verein Dt. Ingenieure, VDI 2221 (1993), VDI 2222 (1997). www.vdi.de

Vogel S, Danckert B, Rudolph S (2010) Entwicklung von Abgasnachbehandlungssystemen auf Basis einer graphenbasierten Entwurfssprache. ATZ-MTZ Konferenz Heavy-Duty-On-und Off-Highway-Motoren, Mannheim

Vogel S, Danckert B, Rudolph S (2012) Wissensbasierte Entwicklung von SCR-Systemen mit graphenbasierten Entwurfssprachen. MTZ 73:702–708

Ward AC (1988) A theory of quantitative inference applied to a mechanical design compiler. Ph.D. Dissertation, MIT, Cambridge

Ward AC, Seering W (1989) The performance of a mechanical design compiler, AI Memo No. 1084. MIT, Cambridge

Ward AC, Seering W (1993) Quantitative inference in a mechanical design compiler. ASME J Mech Des 115(1):29–35

Wertz J (1999) Space mission analysis and design. Microcosm Press, Portland

Whitney D (1996) Why mechanical design cannot be like VLSI design. Res Eng Design 8:124–138

Part VIII
Design Activity

Translating Analytical Descriptions of Cities into Planning and Simulation Models

Kinda Al-Sayed and Alan Penn

Abstract With the increase in urban complexity, plausible analytical and design models became highly valued as the way to decode and reconstruct the organization that makes urban systems. What they lacked is a mechanism by which an analytical description of urban complexity could be translated into a design description. An attempt to define such a mechanism is presented in this paper, where knowledge is retrieved from the natural organization that cities settle into, and devised in a procedural model to support urban planning at the problem definition stage. The model comprises two automated modules, giving preference to street accessibility. The first module implements plausible spatial laws to generate street structures. The performance criteria of these structures are measured against accessibility scores and clustering patterns of street segments. In the second module, an Artificial Neural Networks model (ANNs) is trained on Barcelona's data, outlining how street width, building height, block density and retail land use might be dependent on street accessibility. The ANNs is tested on Manhattan's data. The application of the two computational modules is explored at the problem definition stage of urban planning in order to verify how far deterministic knowledge-based models are in the transition from analysis to design. Our findings suggest that the computational framework proposed could be instrumental at generating simplified representation of an urban grid, whilst being effective at forecasting form-related and functional attributes within a minimum resolution of 200 m. It is finally concluded that as design progresses, knowledge-based models may serve as to minimize uncertainty about complex urban planning problems.

Introduction

Over the last decades, urban studies were witnessing a divide between the analytical sciences and the applied sciences of cities. Analytical sciences embraced many attempts to decode urban complexity by means of explanatory models

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

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_29

(Hillier and Hanson 1984; Hillier 1996; Ratti and Richens 1999; Bettencourt et al. 2007; Batty et al. 2008; Barthélemy 2011). A complementary effort was made in applied urban sciences, where more emphasis was laid on assumption-based simulation models on the scale of cities and regions (Wu and Silva 2009; Marshall 2012). Any attempts to bridge the divide between the analytical and applied sciences of cities were faced by non-trivial challenges, perhaps for the very reason that backed critics against Alexander's work (1964); that is the inherent distinction between analysis and design. To bridge between analysis and design, there needed to be some intuition into the type of mechanism required to devise an explanatory description of urban phenomena into planning and simulation models. How far can these explanatory descriptions be used in reconstructing urban complexity is a question that needs further investigation in the realm of design and computation. In response to this question, this paper embraces an attempt to encode a computational description of the organization that couples street structures with form-function attributes of cities. Learning from Barcelona, Manhattan, and London, a knowledge-based model is devised to aid urban planning.

In line with observed self-referential processes in street networks (Al-Sayed et al. 2012), and the premise that form-function attributes are dependent on the spatial accessibility of road infrastructure (Hillier 1996; Banister et al. 1998), the model proposed here outlines a prioritized structure of urban planning, comprising two automated modules to enable the generation and evaluation of street networks, and the prediction of form-function attributes of urban structures. The methodological framework for the two modules is explained in detail. The generative module is to utilize plausible rules and empirically-validated criteria for assessing the urbanity of the generated street structures. The forecasting module is to devise an Artificial Neural Network model to forecast form-function features of the generated street structure. Both modules are then applied to generate simplified descriptions of an urban grid and predict its attributes. The application of these two modules serves as to explore the extent to which knowledge-based models might determine some features of an urban grid.

From Analytical Descriptions to Planning Prescriptions

Bridging the gap between complexity and planning, significant contributions were made within the framework of urban modelling and simulations (Alexiou et al. 2010; Portugali et al. 2012). For the most part, research in these domains came short of high resolution structural descriptions of urban form. To adapt such descriptions into the linear course of design, a comprehensive framework was required to decode, encode and reconstruct the architecture of cities. For the purpose of developing such a framework, there is a need to frame the problem definition of cities before tackling the problem of design.

One of the first calls to define urban problems was that of Jacobs (1961), where she called for understanding cities as problems of organized complexity. Any translations of this understanding into quantitative descriptions were subject to representation. In general, we could recognize two types of approaches; that of Space Syntax and that of complexity science. Space Syntax is a theoretical framework that builds on a hypothetical relationship between street structures, natural movement and socioeconomic processes (Hillier and Hanson 1984; Hillier 1996). This theoretical proposition is debated in the context of complexity science. Complexity scientists and geographers often questioned the overreliance on linear models used to describe urban relationships (Wu 2002; Batty 2010), whilst questioning the validity of two dimensional representations of urban phenomena (Ratti 2004). This is in view of the argument that reductionist models that relied on simple causal relationships between two variables or more were not immune to erroneous interpretations. Similar skepticism was posed against urban simulation models (Portugali et al. 2012), mainly questioning the over-reliance on assumptions in simulation models, especially when no clear explanation was given on how a plausible knowledge about cities could inform planning and simulations. With some exceptions (Teeling 1996; Duarte et al. 2007), the majority of computational urban planning models were not directly based on tangible knowledge about the mechanisms that drive growth and differentiation in cities. Recently, there has been significant development on this front. Duarte et al. (2012) have developed a computational urban design model called "City Induction"; which incorporated three sub-models; the first sub-model generated context-specific solutions based on the ontologies introduced in "Pattern Language" by Alexander et al. (1977). The second sub-model was based on Stiny's shape grammar and description of designs (Stiny 1981), and was used to generate designs. The third sub-model was building on Hillier's theory on space syntax (Hillier 1996) as an evaluative tool of urban design. There was no conceptualization, however, about how space syntax itself might be used to generate city-like structures.

Attempts to inform urban design theory by virtue of empirical knowledge on spatiotemporal patterns of urban growth were more focused on the regional scale (Dietzel et al. 2005). Methodologies varied depending on the computational models and the elementary seeds used in growth simulations. City blocks were often considered as the elementary components in these simulations (Batty 2005). Despite early attempts to combine structural and shape descriptions (Brown and Johnson 1985), there was generally less emphasis on simulating street structures. Some studies implemented L-Systems in procedural models utilizing discursive rules of addition and subdivision in streets (Parish and Müller 2001), whilst other studies used accessibility scores to assess street patterns generated by means of agent-based models (Derix et al. 2012). These studies, however, made no reference to empirical data on historical urban growth (Al-Sayed et al. 2012).

Forecasting models were also needed to cover a wide range of variables that represent urban complexity without looking at one or two variables in isolation of others. A modelling description of land use transformations in isolation of street infrastructure might pose serious challenges (Stanilov and Batty 2011), particularly when measuring on a hypothetical dependency between street accessibility and urban form and function (Hillier 1996; Banister et al. 1998; Marcus 2010; Porta et al. 2012). Similarly a separation between urban blocks and street patterns

(Lämmer et al. 2006), might be questionable if we consider block agglomerations as the inverted representation of street spaces (Hillier 2002). In the same way, a separation between street width and spatial accessibility (Zheng et al. 2010) needs to be reconsidered when regarding street width as the supply for the demands of street network accessibility (Banister et al. 1998). Urban design models might also benefit from a more comprehensive account of the relationships that couple transformations on blocks and changes on land uses, land values and building height as well as street spacing (Siksna 1997). In reviewing research in this domain, the questions that persist are; how to simulate the growth of street structures in such a way as to build on temporal descriptions of urban transformations? And how to forecast form-function attributes of cities in such a way as to build on empirical models of urban structures and their dependencies?

A Prioritised Structure Model for Urban Planning

In response to the challenges presented in previous sections, this paper presents an attempt to build a computational description of space syntax at the problem definition stage of urban planning, using a scheme that prioritizes urban structures. Our proposed model is based on two procedural modules. The first was a more defined version of the "generic function" (Hillier 1996); that is a street network that is derived from local rules to generate a permeable street structure. The second procedural modules was based on modelling the relationship between street accessibility and form-function variables (block density, street width, building height and land uses).

Methods for Decoding and Mapping Spatial Variables

In the next sections, we will briefly explain the methods developed in Space Syntax theory to measure spatial accessibility in streets. We will also describe our mapping methodology that enabled the construction of empirical models for the purpose of forecasting.

Network Analysis of Street Spaces

In syntactic analysis, street structures are represented by topological and topo-geometric network representations, namely; axial maps and segment maps. An axial map is a network representation of the longest and fewest lines of sight that cover all street spaces. The segment map is a broken description of the axial representation where each segment element between two street inter-junctions is considered as a "node" in a street network, where intersections are considered to be links. In this network representation, nodes are spatially distributed and links are associated with a cost of turning from one street to the other (Turner 2000). Space Syntax research incorporates different measures of network distance; topological, metric and angular (Hillier and Iida 2005). For the purpose of this paper, we were mostly concerned with angular depth and connectivity (degree) of street lines. Angular measures were proven to be powerful at capturing vehicular and pedestrian movement potentials as well as at highlighting catchment areas for active economic centers (Hillier et al. 2012). In segment analysis, integration (closeness) and choice (betweenness) measures can be used to capture the angular geometric properties of street networks. The angular network measures were not normalized until very recently (Hillier et al. 2012), and are still under testing.

Mapping and Aggregation Techniques

In order to map street network measures against other continuous and ordinal variables, a special technique was developed for aggregating spatial data in a separate layer; called the pixelmapper (Al-Sayed 2012). Using this method, indices of street accessibility and form-function attributes of urban areas were aggregated within square polygons (Fig. 1). Larger polygons imply that relationships were captured within a lower resolution. Data was binned in two overlapping polygon grid layers. The second polygon grid layer was shifted diagonally so that the end point of a polygon in the second grid layer was placed on the center of a polygon in the first grid layer. The highest values of both layers were filtered in a third polygon layer with double the resolution of the previous two grid layers. We applied this method on Manhattan to capture the linear correlation coefficient between street accessibility (average NAIN per grid square, or total segment connectivity per grid square) and density of blocks, commercial land uses, high-rise development, and street width. We used different grid resolutions (200, 400, 600, 800, 1000, 1200, 1400, 1600 m). The analysis yields 1000 m as the ideal grid resolution for capturing the highest correlations between street accessibility and form-function variables. It was recognized, however, that lower resolutions will increase the risk of error and will misrepresent the local properties of urban structure (Ozbil et al. 2011). For this reason, data was binned at both the (1000–500 m), and (500–250 m) resolution scales.

A Semi-automated Model for Urban Planning

This paper will expand on the first two procedural modules through automating them in two separate modules; a generative one and a forecasting one. In the generative module, knowledge is utilized at two stages; during the implementation of the generative algorithm and at the evaluation stage. To simulate growth

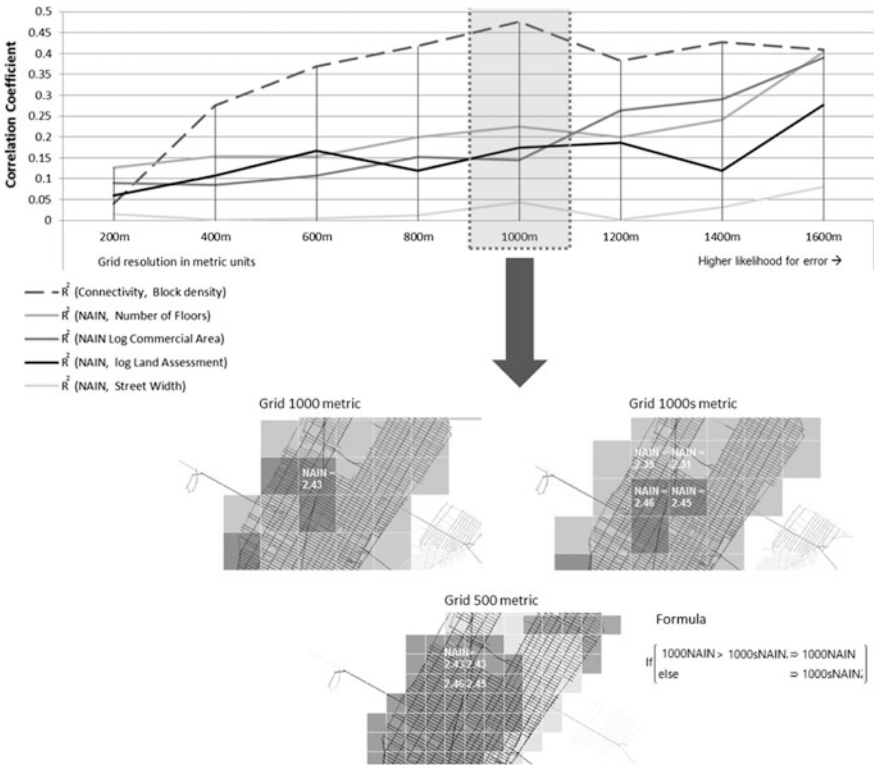


Fig. 1 Binning data to correlate street accessibility to street width and density of blocks, retail landuse and high-rise buildings. Different grid resolutions were used (200, 400, 600, 800, 1000, 1200, 1400, 1600 m), of which 1000 m was proving to score higher correlations. The method implies; storing data and spatial configurations in two overlapping grid reference layers and selecting the highest values in a third higher-resolution reference layer

mechanisms, a set of spatial laws were applied to govern the length and angularity of street spaces. The generated outcome is then evaluated against certain spatial scores.

For the forecasting module, a nonparametric Artificial Neural Networks (ANNs) model is devised. The model relies on empirical data that define the relationship between street structures and form-related and functional attributes, including; street width, building height, block density and land uses in Barcelona and Manhattan.

A Generative Design Module

Early Space Syntax experiments (Hillier and Hanson 1984) presented a generative pattern of organization on the local scale of an urban area. As to reflect on the emergent nature of the resultant grid structures and block alignments, Hillier (1996) recognized the tendency of longer lines to continue straight and shorter lines to be blocked forming near-right angles. By identifying that process as the "centrality and extension" rule he made the assumption that global patterns of urban structures are an emergent product of local rules. Whether a centrality and extension rule on its own can lead to the generation of city structures is something that needs to be questioned, provided evidence on different feedback mechanisms that govern urban growth behavior (Al-Sayed et al. 2012). At this stage, it is difficult to rule out the sequence in which these laws generate urban structures. We therefore take their overall features as criteria for urban pattern recognition.

Rules for Generating Street Structures

In the first simulation module, we generated a number of growth iterations for hypothetical urban structures using Hillier's "centrality and extension" rules whilst allowing for a margin of randomness. Longer lines were encouraged to continue and intersect with other lines forming semi-continuous patterns. Shorter lines were more likely to stop at the first line they intersected with forming near-right angles where possible. The pseudo code follows the following logic;

- Draw three lines starting from random points within the screen area and following random directions;
- Divide each of these lines to 20 segments, and choose randomly one of the points of division as a seed for a new line;
- For 2% of the cases, if the original line happened to be longer than 400, and the new generated line started close to one of the end points of the original line, direct the new line in an angle that is within (0, $\pm$ 12.8) degrees range.
- For another 2% of the cases, let the new line go in a direction that is within (0, $\pm$ 42.8) degrees range.
- If the original line is shorter than 400, for 52% of the cases let the new line go in a direction that is close to 90 $\pm$ 2.2°.

The resultant structures presented varying syntactic properties. In order to recognize structural patterns that match those of cities we compared these iterations to real urban structures.

Assessing the Urbanity of the Growth Iterations

To establish criteria for evaluating generative growth iterations, we compared the generated structures to a random structure and to London's street structure and an existing sample taken from Barcelona's deformed grid. The random structure is regarded as hypothesis null; marking the lowest performance of a street structure.

The hypothesis is that structures generated by virtue of Hillier's simple rules will be more similar to real cities than to a random structure.

Through conducting research on 50 US cities, research by Bin Jiang confirmed that connectivity (degree) of street networks follows a power-law distribution (Jiang 2007). This was also observed in the historical growth of Barcelona, which revealed preferential attachment dynamics (Al-Sayed et al. 2012). This observation is used here to evaluate the structures of the generated iterations. From Table 1, it is clear that the correlation coefficient of power-law distributions is not a strong discriminator of real street networks compared to random networks. Iterations 3 and 1 presented closer values of correlation coefficient to both London and Barcelona, but these values were also close to a random network. When measuring on the parameters of power-law distributions; *a* and *k*, the distinctions between random and urban systems became more visible. The values of *a* yielded iteration 4 as the closest iteration to Barcelona's deformed grid, whilst iteration 3 came second. The values of *k* yielded iteration 1 as the closest iteration to Barcelona and London, and again iteration 3 came second.

As established in Al-Sayed et al. (2010), in a grid that presents a differentiated structure integration values tend to follow lognormal distribution. The distribution differs from that of random networks in that it shows a higher degree of skewness (asymmetry). On aggregate, the closeness centrality of a random network will be characterized by a normal distribution with minimal skewness. This constitutes the

Table 1 Evaluating the four growth iterations against the spatial properties of Barcelona and a randomly generated structure

		Iteration 1	Iteration 2	Iteration 3	Iteration 4	Barcelona	London (all)	Random structure
Spatial structure								
Power-law distribution of connectivity	R^2	0.92	0.93	0.92	0.926	0.865	0.832	0.87
	a	130.2	178.0	179.2	249.5	234.62	1251.7	1505.7
	k	-0.7	-0.72	-0.71	-0.73	-0.61	-0.579	-1
Lognormal Distribution	KSL	0.09	0.15	0.088	0.1	0.035		0.04
	Skewness	-1.32	-1.15	-1.46	-1.5	0.28		-0.18
Intelligibility (R^2)		0.12	0.13	0.17	0.1	0.33	0.061	0.56

The generative code is written in Processing (Java). Spatial Structures are analysed using UCL Depthmap (Turner 2011) second criterion for evaluating the urbanity of generative structures. The lognormal distribution can be evaluated through measuring the goodness-of-fit D representing the distance between the cumulative distribution and a cumulative fraction plot for the data sample. The goodness-of-fit is measured by running the empirical distribution function KSL test (Lilliefors 1967). Skewness is also added as an indicator to the degree of asymmetry in the structure as a whole in comparison with a random structure.

Measuring on the cumulative structural properties of depth in the network, aggregate integration values did not seem to fit very well to a lognormal distribution compared to the randomised map and Barcelona. Barcelona's structure prevailed over a randomised map in its fitness to a lognormal distribution, and it showed higher degree of skewness.

Judging on KSL test, iteration 3 presented a better fit with lognormal distribution as well as a differentiated structure (Skewness = −1.46).

Given that distributions do not interpret structural properties, another criterion is added to compare the relationship between axial connectivity and axial integration as a measure of intelligibility. Urban systems exhibit relatively high intelligibility between the local and global axial structures (Conroy 2001).

Considering intelligibility as a measure of the part-whole structural unity, the structure of iteration 3 was found to be more intelligible compared to other iterations. Yet, it is difficult to foresee how intelligibility might act as a law for recognising the urbanity of street structures, since our observations indicate that random networks are more intelligible than both Barcelona's grid, and London's street network. It is worth mentioning here that intelligibility is largely influenced by the system's size. To verify these results, we may need to use different rules and seeds for the randomised networks.

Considering these findings, iteration 3 prevailed as it presented an optimum foreground structure that conserved physical distance and angular turn costs. It also presented a higher level of structural differentiation that made a better match with real cities. On aggregate, angular depth values in iteration 3 followed a lognormal distribution. The structure of iteration 3 was also more intelligible than other structures. Despite the relative success of iteration 3, it failed to be fully compatible with real urban structures. Additionally, it was difficult to identify an optimum performance for the growth iterations, a performance that might fully comply with how urban structures are configured in real cities. However, for the purpose of our experiment; we proceeded by applying the ANNs model on iteration 3 to further define the design features of the generated structure.

A City-to-City Learning Approach

In this section, a supervised machine learning model will be applied using a soft computing technique based on ANNs. The use of ANNs in modeling would enable empirical encoding of data on space, form and function. The ANNs allow for minimizing assumptions about the input and output data distribution and the type of data used, whether continuous, categorical, or binary. They are particularly useful in cases where complexity in the system relationships and imprecision in observations are issues that threaten the credibility of simpler models. ANNs are also fault tolerant towards redundant information coding, where there are hidden relationships between spatial measures or between socioeconomic variables.

ANNs consist of layers and neurons that simulate human learning. The training of ANNs can help storing embedded functions that are then used to categorize information and provide projections given new situations. With such functionality, ANNs can be used to answer *what if* questions and generalize complex relationships on presumably similar situations to the situation used in the training. ANNs are used in many fields; including medical sciences, engineering, A.I. and many others. They are also known to be successful in the nonlinear mapping and modeling in geography and planning (Openshaw and Openshaw 1997). The downside in using ANNs is in the difficulty to describe the relationship between the input variables and the output variables. All the training takes place within a *black box*. Neural Networks comprise a large class of different model architectures. Traditionally, ANNs are used to classify a set of observations. In most cases, the issue is in approximating a static nonlinear, mapping $f(\mathrm{x})$ with a neural network $f(\mathrm{x})_{\mathrm{NN}}$, where $\mathrm{x}\in\mathrm{R}^{\mathrm{K}}$. The ANNs model to be used in training space and form-function data in this section will consist of three layers, the input, output and a layer with hidden nodes in-between. The different layers are encoded in the *multilayer-perceptron* (MLP) model[1] (Rumelhart and McClelland 1986) illustrated in (Fig. 2). Three hidden nodes are considered in the middle layer, where activation functions that store weights and biases are embedded. The ANNs model will be *fully connected* and will use a *feed-forward* mechanism. The network is *fully connected* since the output from each input and hidden neuron is distributed to all of the neurons in the following layer. The *Feed-forward* mechanism of the model entails that the values would only move in the forward direction from input to hidden to output layers; so that no values are fed backwards to input or hidden layers. Due to the limited number of inputs (3) and outputs (4) and a fair amount of redundancy (correlation) between two spatial measures in the input layer, we chose simple network architecture for the model.

The ANNs is fitted using standard nonlinear Least-Squares Regression methods. The inputs x_n, $n = 1,\ldots, n$ to the neuron in the hidden layer are multiplied by weights w_{ni} and summed up together with the constant bias term Q_i. The resulting n_i is the input to the activation function y. The activation function used here is the hyperbolic tangent function which is a sigmoid function. It transforms values to be between −1 and 1, and is the centered and scaled version of the logistic function. The hyperbolic tangent function is:

[1]MLP consists of multiple layers of simple, two state, sigmoid processing nodes/neurons that interact using weighted connections.

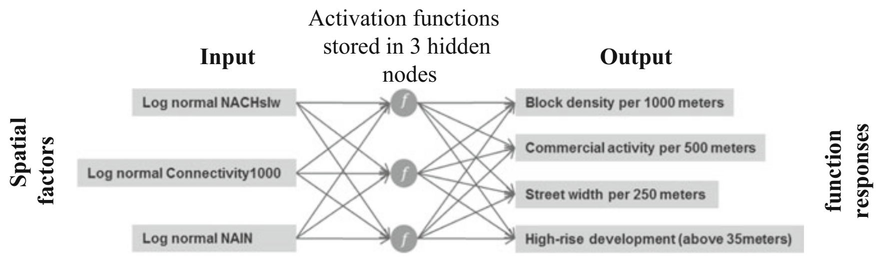


Fig. 2 An ANNs model applied to Barcelona and Manhattan, considering normalized spatial measures of choice, integration and connectivity as factors and form-function attributes as responses

$$\frac{f(x)}{\tan h(x)} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{1}$$

where x is a linear combination of the X variables.

The output of node i is then defined as the following;

$$y_i = f\left(\sum_{i=1}^{k} w_{ik}x_j + Q_i\right) \tag{2}$$

To explore the application of ANNs in urban planning, we encoded empirical data from Barcelona, and tested the model against data from Manhattan. Geometric measures of street network configurations were used as inputs. The output was a combination of Form attributes (building height and density, street width), and functional (overall commercial zoning).

The ANNs were trained and validated against Barcelona's. In instructive training, the error information was propagated backwards through the network using a backward propagation algorithm. The algorithm iteratively minimizes an error function over the network outputs and desired outputs (Rumelhart et al. 1986; Foody 1996). We used the KFold method to validate the operative mechanism of the model by recursively selecting one subset out of five. The subset that best validated the model was then chosen. The validation helped detecting if the model overfits the data.

For the input layer, we used normalised choice [Segment length weighted] (NACHslw), normalised integration (NAIN) and aggregate connectivity per 1000 square unit (Connectivity 1000). All indices were computed using Depthmap (Turner 2011). NACHslw is an angular measure of graph betweenness that is normalised and weighted by street segment length (Turner 2000). NAIN is a normalised and angular-weighted measure of graph's closeness (Hillier et al. 2012). Connectivity is equivalent to degree in graph theory. It is here aggregated per 1000 m square unit. Both NACHslw and NAIN were calculated for the whole system (radius n). Before using the continuous variables as input in the ANNs model, we normalised their values using a lognormal probability function to fit in the range [0, 1]. The dependent responses were a mix of continuous variables running in *regression* mode (Block density per 1000 metric square) and ordinal variables running in *machine* mode (commercial activity, street width above 30 m, high rise above 35 m). The positive presence of the ordinal response variables was marked as 1 and the negative presence is 0.

The performance of ANNs on Barcelona was evaluated using a Linear Regression for block density. Both *accuracy* and AUC measures were applied to evaluate the predictive power of the ANNs running in *machine* mode. For *accuracy*, we calculated the rate of classified scores against total scores from the confusion matrix.[2] The Receiver Operating Characteristic (ROC) curve plotted the true positive rate (sensitivity)[3] on the vertical axis and false positive rate (specificity)[4] on the horizontal axis. For ROC, we calculated the area under the ROC curves (AUC). We then observed the cross-validated estimates of *accuracy* and AUC. In addition, the Root Mean Square Error (RMSE) between vaildated and training data was examined to check for overfitting.

Training Artificial Neural Networks on Barcelona's Data

Measuring on *accuracy* and AUC, the ANNs were successfully fitted between the input (indices of accessibility) and the output (form-function) data. The difference in Root Mean Square Error (RMSE) between trained and validated data was minimal (0.01, 0.03, 0.01 for ordinal variables) showing no signs of overfitting. The AUC recorded values above 0.8, 0.81, 0.79 in predicting High-rise development, Commercial activity and Street width respectively (Fig. 3). Measures of *accuracy* were recording 0.71, 0.82, 0.74 for classifying the presence of High-rise buildings, Commercial activity and Street width respectively. The correlation between actual and predicted block density was also high $R^2 = 0.61$. The evaluation rates indicated that spatial accessibility can classify the positive/negative presence of ordinal responses and correspond to block density.

Testing the Artificial Neural Network Model on Manhattan

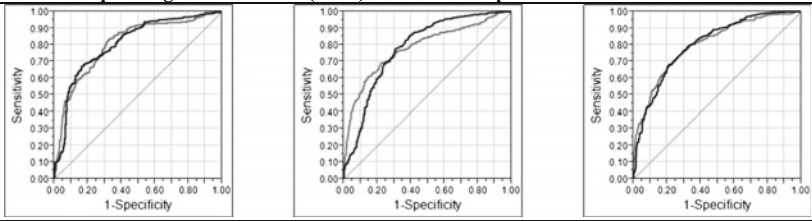
In this section, ANNs was tested against Manhattan's indices of accessibility. The three indices of accessibility were devised again as independent factors (explanatory). The input spatial data was scaled into the range [0, 1] for both Barcelona and Manhattan (Gong 1996). This scaling made these variables compatible with the

[2]Accuracy can be calculated from the contingency table as follows; ((True Positives) + (True Negatives))/((True Positives) + (True Negatives) + (False Positives) + (False Negatives)).

[3]Sensitivity = True Positives/((True Positives) + (False Negatives)).

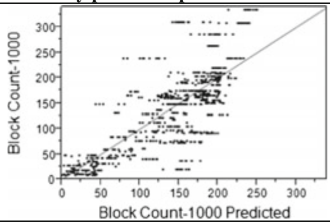
[4]Specificity = False Positives/((False Positives) + (True Negatives)).

Receiver Operating Characteristic (ROC) for ordinal responses



	Commercial activity	Street width	High-rise above 35m
— 0 — 1			
Area	0.81	0.79	0.80
Accuracy	0.82	0.74	0.71

Actual by predicted plot for continuous response



$\mathbf{R^2}$ = 0.61

Block density per 1000metric square

Fig. 3 ROC and scattergram plots evaluating the performance of the Neural Networks on Barcelona's data. Spatial configurations were used as factors. Form-related and functional attributes were considered as responses

sigmoid activation function. For this reason, we normalized indices of accessibility using a *lognormal* probability distribution to fall within the range [0, 1]. The *lognormal* distribution function was chosen because it fits well with the distribution of the three indices (Lilliefors 1967). For evaluation, we used the correlation coefficient R^2 to plot block density predictions against actual block density in Manhattan, and we used contingency tables to calculate the ratio of successful scores[5] against misses[6] and false alarms.[7]

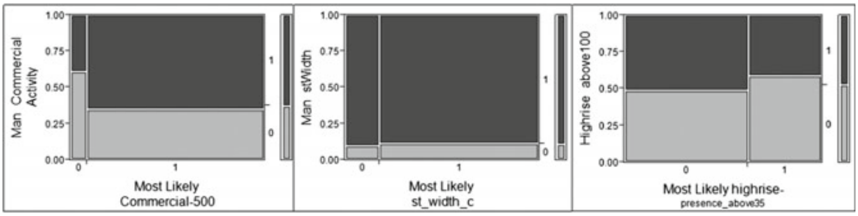
The comparison (Fig. 4) showed correspondence in Manhattan. The mosaic and scattergram plots in Fig. 4 showed how response variables correspond to predicted likelihoods. The matching scores between actual data and predicted responses are significant (0.64, 0.74) for commercial activity and street width, but less so for high rise detection. The R^2 value showed a correlation of 0.52 between actual and predicted block density.

[5]Successful scores are where there is an agreement between predicted change and true change.

[6]Misses are where there are no change predicted but change actually occurred.

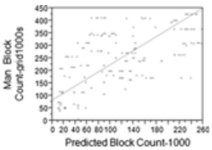
[7]False alarms are where there is change predicted but no change actually occurred.

Contingency tables for actual parameters by predicted ordinal responses



	Commercial activity	Street width	High-rise above 35m
Matching rate[8]	0.65	0.74	0.51

Actual Block count in Manhattan by predicted continuous responses



$\mathbf{R^2} = 0.52$

Block density per 1000metric square

Fig. 4 Different contingency and scattergram plots elucidating how Manhattan's data corresponds to predictions enabled by ANNs that was initially trained on Barcelona's data. [a]The rate of true positive and true negative to all scores

Forecasting Form-Function Attributes for a Hypothetical Grid

The validation and testing qualified the ANNs model to be used in forecasting form-function variables for a given spatial structure. This time, the *pixelmapper* method was used to define the approximate features of the urban space. The attributes of the solution space were then defined within that resolution level (Fig. 5). The street width response was estimated directly from the NACHslw values and further informed by the ANNs predictions. The rest of the estimated attributes were fully automated assuming a full correspondence between the spatial measures of the winning iteration (iteration 3) and the response variables. The automation was subject to the accuracy of the ANNs model and the scale of representation. Scale might be identified as the metric resolution of the square units in the *pixelmapper* grid. To produce a smooth representation of the target spaces, positive values (1) for ordinal responses were replaced by their correspondent probabilities. Further elaborations on how the pixelated target spaces for the response variables might be translated into 3D descriptions of design solutions was explored in Al-Sayed (2014a).

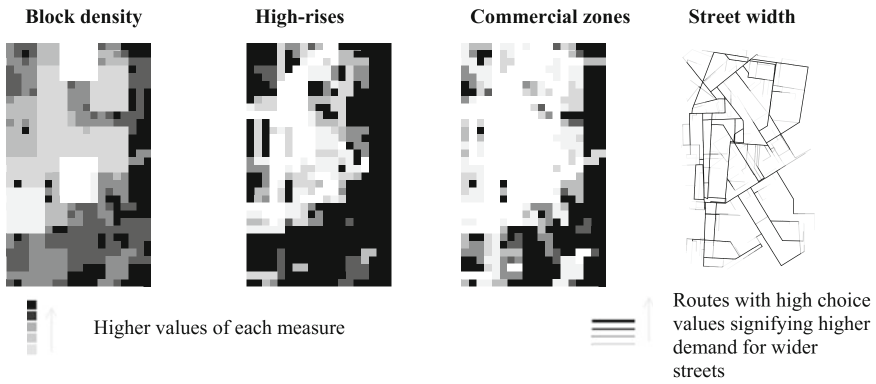


Fig. 5 Responses for form-function estimated by applying the trained and validated ANNs model. The spatial network measures of iteration 3 were used as factors in the ANNs

Conclusions

The design approach presented here builds and extends on a theoretical urban planning model that prioritises the structure of street networks in generating and predicting other features of urban form and function (Al-Sayed 2012, 2014a, b). The theoretical urban planning model; namely the prioritised-structure model involved a procedural development starting from the universal; that is a permeable street structure, to define the variables that are thought to be dependent upon street accessibility, based on the assumption that street accessibility has a preference in defining the demand for high rise development, wider streets, dense urban fabric and retail land uses. To encode all these variables we used ANNs.

The design experiment presented in this paper comprised two automated modules. As part of the first module, a generative algorithm was implemented. Four growth iterations were evaluated and compared to a random system and a section of Barcelona's grid structure. The evaluation helped selecting a growth iteration that successfully reproduced the spatial properties witnessed in real cities. The generative process was fully automated. Yet, the evaluation revealed few shortcomings that were either related to the inadequacy of certain measures or to the directional growth mechanisms implemented. Some shortcomings stemmed out of the difficulty to automate a recognition system for certain spatial measures, particularly those related to the definition of street clusters, which might be recognised through spectral analysis (Hanna 2012).

In the forecasting module, ANNs were devised to encode the different topological and geometric measures of street configurations as factors and the different form-function attributes of cities as responses. The model was trained, validated on empirical data from Barcelona, and then tested against data from Manhattan. Data was mapped using an aggregation technique called the *pixelmapper*. The method was introduced in (Al-Sayed 2012), although similar methods were explored in GIS

(Ye and van Nes 2012). The *pixelmapper* technique helped binning different types of spatial, binary and continuous data into pixelated square units; hence it was possible to look for invariant relationships in-between different variables within the metric limits of each pixel. Accordingly, a system-based planning model was devised using the ANNs activation functions that defined a nonlinear relationship between street network measures and data on form-function in Barcelona. When applied to Manhattan's data, the applicability of ANNs was returned positive. This finding yields with the possibility of applying the functionality of the model on predicting form-function attributes for hypothetical grid structures, hence as a tool to aid urban planning. There might be issues, though, that has to do with the computational cost of training ANNs on big data, which might limit the applicability of our proposed model on the regional scale. This effect was trivial in our study, since the largest set of data used in the training was 44,093 street segments.

The work presented here encompasses a plausible model to support urban planning at the problem definition stage. For a more comprehensive account of the variables that shape urban form, the model needs to be incorporated as part of a broader model description, considering environmental parameters, and qualitative properties of the urban environment. We only accounted here for variables that might be estimated from street network geometric and topological configurations, where street space acts as a proxy of other urban features. In Space Syntax (Hillier 1996), the affordances of street networks for movement were thought to shape the economic development in cities. This notion was recently recalled in urban morphology (Oliveira 2013). Hence, spatial accessibility is likely to have a preferential role in urban planning. Up to this date, space syntax description as an evaluation tool in urban design (Duarte et al. 2012; Karimi 2012). In our approach, however, we emphasize that the direct adaptation of analyses into planning applications would help supporting policy-making practices with empirical evidence through the use of plausible computational models.

References

Al-Sayed K (2012) Space Syntax as a parametric model. In: Proceedings of the Fall 2011 PUARL international conference, Oregon

Al-Sayed K (2014a) Thinking systems in urban design: a prioritised structure model. In: Carmona M (ed) Explorations in urban design. Ashgate, Farnham, pp 169–181

Al-Sayed K (2014b) A systematic approach towards creative urban design. In: Gero JS (ed) Design computing and cognition DCC'12. © Springer 2014, pp 133–150

Al-Sayed K, Turner A, Hanna S (2010) Modelling the spatial morphogenesis in cities: the dynamics of spatial change in Manhattan. In: Timmermans HJP (ed) Proceedings of the 10th international conference on design and decision support systems in architecture and urban planning, Eindhoven

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 de Chile, PUC

Alexander C (1964) Notes on the synthesis of form. Harvard University Press, Cambridge
Alexander C, Ishikawa S, Silverstein M (1977) A pattern language: towns, buildings, construction, vol 2. Oxford University Press, Oxford
Alexiou K, Johnson J, Zamenopoulos T (eds) (2010) Embracing complexity in design. Routledge, London
Banister D, Penn A, Hillier B, Xu J (1998) Configurational modelling of urban movement networks. Environ Plan 25(1):59–84
Barthélemy M (2011) Spatial networks. Phys Rep 499(1):1–101
Batty M (2005) Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals. MIT Press, Cambridge
Batty M (2010) Networks, flows, and geometry in cities: a challenge to space syntax. J Space Syntax 1(2):366
Batty M, Carvalho R, Hudson-Smith A, Milton R, Smith D, Steadman P (2008) Scaling and allometry in the building geometries of Greater London. Eur Phys J B 63(3):303–314
Bettencourt LM, Lobo J, Helbing D, Kühnert C, West GB (2007) Growth, innovation, scaling, and the pace of life in cities. Proc Natl Acad Sci 104(17):7301–7306
Brown FE, Johnson JH (1985) An interactive computer model of urban development: the rules governing the morphology of mediaeval London. Environ Plan 12(4):377–400
Conroy R (2001) Spatial navigation in immersive virtual environments. Doctoral dissertation, University College London
Derix C, Gamlesæter Å, Miranda P, Helme L, Kropf K (2012) Simulation heuristics for urban design. In: Digital urban modeling and simulation. Springer, Berlin, pp 159–180
Dietzel C, Herold M, Hemphill JJ, Clarke KC (2005) Spatio-temporal dynamics in California's central valley: empirical links to urban theory. Int J Geogr Inf Sci 19(2):175–195
Duarte JP, Rocha JM, Soares GD (2007) Unveiling the structure of the Marrakech medina: a shape grammar and an interpreter for generating urban form. AI EDAM 21(4):317–349
Duarte JP, Beirão JN, Montenegro N, Gil J (2012) City Induction: a model for formulating, generating, and evaluating urban designs. In: Digital urban modeling and simulation. Springer, Berlin, pp 73–98
Foody GM (1996) Relating the land cover composition of mixed pixels to artificial neural network classification output. Photogram Eng Remote Sensing 62:491–499
Gong P (1996) Integrated analysis of spatial data from multiple sources using evidential reasoning and artificial neural-network techniques for geological mapping. Photogramm Eng Remote Sensing 62:513–523
Hanna S (2012) A representational scheme for the extraction of urban genotypes. In: Gero JS (ed) Design computing and cognition DCC'12. © Springer
Hillier B (1996) Space is the machine. Cambridge University Press, Cambridge
Hillier B (2002) A theory of the city as object: or, how spatial laws mediate the social construction of urban space. Urban Design Int 7:153–179
Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press, Cambridge
Hillier B, Iida S (2005) Network and psychological effects in urban movement. In: Cohn AG, Mark DM (eds) Proceedings of spatial information theory: international conference 2005. COSIT, Ellicottsville, pp 468–476
Hillier B, Yang T, Turner A (2012) Normalising least angle choice in Depthmap—and how it opens up new perspectives on the global and local analysis of city space. J Space Syntax 3 (2):155–193
Jacobs J (1961) Death and life of Great American Cities—the failure of town planning. Penguin Books, Harmondsworth
Jiang B (2007) A topological pattern of urban street networks: universality and peculiarity. Phys A 384(2):647–655
Karimi K (2012) A configurational approach to analytical urban design: 'Space Syntax' methodology. Urban Design Int 17(4):297–318
Lämmer S, Gehlsen B, Helbing D (2006) Scaling laws in the spatial structure of urban road networks. Phys A 363(1):89–95

Lilliefors HW (1967) On the Kolmogorov-Smirnov test for normality with mean and variance unknown. J Am Stat Assoc 64:399–402

Marcus L (2010) Spatial capital. J Space Syntax 1(1):30–40

Marshall S (2012) Science, pseudo-science and urban design. Urban Design International 17 (4):257–271

Oliveira V (2013) Morpho, a methodology for assessing urban form. Urban Morphol 17(1):149–161

Openshaw S, Openshaw C (1997) Artificial intelligence in geography. Wiley, Chichester

Ozbil A, Peponis J, Stone B (2011) Understanding the link between street connectivity, land use and pedestrian flows. Urban Design Int 16:125–141

Parish YI, Müller P (2001) Procedural modeling of cities. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques. ACM, New York, pp 301–308

Porta S, Latora V, Wang F, Rueda S, Strano E, Scellato S, Latora L (2012) Street centrality and the location of economic activities in Barcelona. Urban Stud 49(7):1471–1488

Portugali J, Meyer H, Stolk E, Tan E (eds) (2012) Complexity theories of cities have come of age: an overview with implications to urban planning and design. Springer, New York

Ratti C (2004) Urban texture and space syntax: some inconsistencies. Environ Plan 31(4):487–499

Ratti C, Richens P (1999) Urban texture analysis with image processing techniques. In: Computers in building. Springer US, New York, pp 49–64

Rumelhart DE, McClelland JL (eds) (1986) Parallel distributed processing, vol 1. MIT Press, Cambridge

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323:533–536

Siksna A (1997) The effects of block size and form in North American and Australian city centres. Urban Morphol 1(1):19–33

Stanilov K, Batty M (2011) Exploring the historical determinants of urban growth patterns through cellular automata. Trans GIS 15(3):253–271

Stiny G (1981) A note on the description of designs. Environ Plan 8:257–267

Teeling C (1996) Algorithmic design: generating urban form, vol 2, pp 89–100

Turner A (2000) Angular analysis: a method for the quantification of space. Working paper 23, Centre for Advanced Spatial Analysis. UCL, London

Turner A (2011) UCL Depthmap: Spatial network analysis software, version 10. University College London, VR Centre of the Built Environment, London

Wu F (2002) Complexity and urban simulation: towards a computational laboratory. In: Geography Research Forum. Ben-Gurion University of the Negev Press, pp 22–40

Wu N, Silva EA (2009) Artificial intelligence and 'waves of complexity' for urban dynamics. In: Proceedings of 8th WSEAS international conference on artificial intelligence, knowledge engineering and data bases (AIKED '09), pp 459–464

Ye Y, Van Nes A (2012) New ways of spatial modelling: making spatial diagnosis in combining space syntax, spacematrix and MXI with GIS of New and Old Towns. In: New urban configurations: EAAE/ISUF international conference, Delft, October, pp 16–19

Zheng X, Zhao L, Su Y Yan G, Wang S (2010) An extension method of space syntax and application, advancing computing, communication, control and management. Lecture Notes Electr Eng 56:7–14

Exploring the Cognitive Dynamics of Product Appreciation

Niccolò Becattini, Gaetano Cascini and Francesca Montagna

Abstract Understanding users' choices is the key to both plan and design a product having higher chances of market success. Designers and marketing professional adopt different approaches for such investigation and the lack of a shared perspective between them can represent, by itself, one of the causes of no success in the market arena. This paper presents an approach that aims at focusing on how customers/users interpret product features and link them to their own needs. The theoretical framework behind the proposed approach is based on the latest updates of the situated FBS framework. An illustrative application in the field of sport shoes clarifies strengths and weaknesses of the proposal.

Introduction

From a traditional engineering perspective, the design process consists in defining solutions that fully address the project requirements. However, looking backwards in time, there exists a large set of perfectly working technical products that turned out to be a commercial fiasco (Becattini et al. 2013), simply because they did not meet the customer appreciation. To this purpose, in the last decades, both scholars and practitioners have posed a higher attention to the interpretation of users' needs, up to including the users in the design loop. Literature shows that this attempt has been carried out through different approaches. One consists in the definition of success factors starting from the analysis of successful products (e.g., Borgianni et al. 2013; Cooper and Kleinschmidt 1987; Cooper 1999). Others aim at capturing customer's preferences (Griffin and Hauser 1993). The most recent and radical thread is the shift from design "for the user" to inclusive design "with the user" (Lee 2008).

N. Becattini · G. Cascini (✉)
Politecnico Di Milano, Milan, Italy
e-mail: gaetano.cascini@polimi.it

F. Montagna
Politecnico Di Torino, Turin, Italy

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_30

In general, these methods share the assumption that customers have a set of more or less tacit needs and that—given a product that adequately fulfills these needs—those customers will decide to purchase the product, thus ensuring product success in the market. When moving from B2C (Business-To-Consumer) to B2B (Business-to-Business) markets, and/or when dealing with highly innovative products, really this kind of translation becomes more complex. In these cases, products' attractiveness and customer needs are connected to product use, but products must be conceived considering that customers may be embodied by different actors, with different roles and expectations. For this reason, the design community is increasingly investigating the adoption process and the stakeholders there involved (Cantamessa et al. 2016), so to generate solutions that can be more easily accepted, thus resulting into a concrete innovation. Therefore, the identification of mechanisms behind the customer adoption of a given product represents a paramount research goal for those who aim at "designing for innovation".

This desire to investigate the process by which products are adopted, neglecting for a moment the influences exerted by the other stakeholders involved in the process, has its starting point in the analysis of the customer' adoption decision. This issue is something already studied in many streams of literature, at least because marketing contributions have focused on interpreting customer behavior for many decades. However, the perspective adopted in this paper is different. The paper does not focus on the interpretation of customer needs or on the psychological processes that induce customers to adopt, but aims at looking at the customer through the eyes of the designer, shedding light on the user's interpretation of the product features and the cognitive dynamics through which users link them to their needs.

The authors believe that an analysis from this original perspective can be useful for two main purposes. On the one hand, it can support the different actors that plan the development of new user-centered products to appoint directions for the improvement. On the other hand, it can facilitate product designers in the communication of goals and objectives thanks to a better understanding of what happens in the mind of the user when interacting with the product. Designers, in turn, are also better supported in defining the best ideas among the alternatives they explore when creating new products.

Customer needs are object of investigation both for marketing professionals and for designers; the former should interpret what emerges from market in order to provide useful insights to designers; the latter, instead, from needs should derive requirements and specifications that will drive the design activities. Therefore, the investigation of needs and the requirement definition represent the thorny interface between two worlds, marketing and design. Moreover, the major relevance of this activity is due to the eventual common goal that these two business units have at the end: the product planning that is one of the most strategic steps in product development.

Despite of this peculiarity and relevance, if one looks at the usual industrial practice, the identification and interpretation of needs are carried out separately and the two business units use completely different approaches, as well as different methods and tools. That evidently represents a quite problematic issue, if one thinks that activities such as "identification" and "interpretation" represent the main tasks of both the groups, and possible misalignments are recognized in literature as one of the factors that affect the success of the product on the market (Cooper 1999; Calantone et al. 1993). The need for integrating the two business units is well known in literature (Slater and Mohr 2006; Rein 2004); however, none of the current contributions proposes tools that can be practically deployed for supporting the product development process. This paper, hence, aims at defining approaches fostering synergies between marketing and design.

The paper is mainly structured in three parts: the first part focuses on the methods usually adopted in industry; one central methodological section that, from the reference models in literature, states the hypothesis for the applicability and tries to propose the methodological approach; and a third section that proposes an illustrative application of the proposed method. Some conclusive remarks are discussed at the end.

Interpreting Needs in the Marketing Literature

In the marketing literature, "needs" represent the basic motivation for pushing people to change their situation (Maslow 1987), and hence they are at the basis of customer adoption decisions. Really, the marketing literature even explicitly considers consumer needs as an output of consumers' decision-making processes, as well as recognizes the importance of widening the perspective toward the customer behavior (Loudon 1988; Sheth and Mittal 2004).

The tacit nature of consumers' purchasing behavior has induced marketing scholars to use results from cognitive psychology in order to gain a better understanding of this phenomenon. Human beings are known to be unable to cope with multidimensional problems (such as purchasing a product) and therefore daily use perception to aggregate tacit elements and "hard" raw data in a hierarchical way, until the problem becomes manageable (Brunswik 1952). According to this view, consumers "perceive" the mass of information attached to a product by abstracting it in a space of limited dimensionality (Tapp 1984), defined by high-level needs and features, in which they will make their purchasing decision. This perceptual process does not occur as a single event in time, but evolves along time, is tacit and—to some extent—unconscious.

The most common approach for capturing the perceptual process consists in performing market research actions, so to recreate the hierarchical process followed by the customer (Griffin and Hauser 1993). The aim becomes modeling a hierarchy of customer needs, being this classification related to different life aspects (Maslow 1987), or to the dimensional attributes of a product (Kano 1984). Focus groups, scenarios of use, user trials, customer diaries, etc. are used to gain the tertiary needs arising from consumers, then these needs are aggregated into secondary needs to feed a deeper analysis toward the primary dimensions of the perceptual space. Perceptual maps (Dolan 1990) are powerful representations of the primary needs arising from the market and of the way with which the industry is fulfilling them.

It is quite clear that this process, in addition to being extremely expensive, requires to be able to organize the information in an extremely rigid and structured way, as well as it requires knowing the pool of potential customers the product is for. There are cases for which it is difficult to start from an existing offering or a comprehensive view of the customers, especially when products are new and highly innovative. In this context, some more unstructured and qualitative techniques (e.g. empathy maps, Gray et al. 2010) have been developed to generate insights not only on the needs customer has, but even on the features that characterize possible groups of customers. In this sense, the potential customer does not exist before the product is conceived, and, therefore, customer and needs are identified and generated together with the business model.

It is quite evident that all these methods focus on the motivations and intentions of an adoption process. However, they capture these aspects in a too generic way for designing. If marketing wants to provide really useful information to designers, this should be provided according to cognitive categories designers usually deal with during their activities. On the other hand, if designers want to develop products that have higher chances of market appraisal, they have to clarify their own vision about design targets and objectives through the knowledge of both customers' cognition and adoption process. The generic marketing category of "needs" becomes not enough to this aim and "needs" instead should be detailed so that the structural features of the product, the desired product behaviour, the intentions for the purchasing and the actual native needs that eventually push users to decision are distinguished.

This paper, therefore, aims at building a practical methodology to elicit user's needs and to understand motivations and mechanisms behind the adoption of a product, so that both marketing professionals and designers can use the same variables for the analysis and infer more precisely customers' desires and intentions.

Methodological Approach and Reference Models

This section explains the research methodology and the reference models adopted by the authors to address the issues proposed in the introduction, also according to the considerations about the state of the art presented above. After resuming some essential concepts about the situated FBS framework (Gero and Kannengiesser 2004) and its extension to represent user needs (Cascini et al. 2013), the following paragraphs detail the assumptions and hypotheses of the present work and the experimental approach proposed to check their validity.

The Situated FBS Framework Integrating Needs Identification

The Function-Behaviour-Structure framework was originally presented by Gero (1990) in order to represent the main patterns characterizing cognition in design. That reference model has been applied in different studies and further updated in Gero and Kannengiesser (2004) to integrate a situated perspective, i.e. shifting the perspective from the sole mind of the designer to the designer immersed in a specific context. This allows a finer description of the mechanisms through which a designer interprets reality through its sensory experiences, builds meaning out of such experiences and updates its knowledge according to the way those experiences have been interpreted. More recently, the situated FBS framework has been considered as a relevant ontology also to represent user needs, in the perspective of extending the design process to tasks such as needs identification and requirements formulation (Cascini et al. 2013).

Table 1 summarizes the variables presented in the FBS framework, including the latest extensions to needs and requirements. The ways the above variables are transformed into each other define the set of cognitive activities that a designer carries out to generate a design proposal. Gero (1990) defined them as a set of five elementary actions (+1 referring to the documentation of what was designed):

- the *formulation* of a design goal (F → Be)
- the *synthesis* of a solution (Be → S)
- the *analysis* of its outcomes (S → Bs);
- the *evaluation* of its suitability (Bs ↔ Be); and
- the possible *reformulation* of a design variable (S → [F; Be; S]).

Table 1 Variables considered in the FBS framework and its extensions

Name of the variable	Label	Definition of the variable
Function (Gero 1990)	F	It describes the aim of the object, what the object is for, its purpose
Structure (Gero 1990)	S	It describes the object's components and their relationships, i.e., what the object is
Behaviour (Gero 1990)	B	It describes what the object is expected to do (Be) or what the object does (Bs) according to the structure variables (S) characterizing it
Need (Cascini et al. 2013)	N	It is an expression of an undesirable situation to be avoided or a desirable situation to be attained. This situation can be perceived by any of the actors involved in the product life from the purchasing phase to each stage of use and disposal. Needs can be explicitly stated to the designer or perceived by the designer because of being extracted (or even postulated) by the observation of users' behavior
Requirement (Cascini et al. 2013)	R	It is a measurable property related to one or more Needs. They 'are structured and formalized information about a product' and 'consist of a metric and a value'

The situated perspective considers cognition as it occurs during the interaction the designer has with the surrounding environment/context. To this purpose, Gero and Kannengiesser (2004) introduced three different layers (worlds) through and within which cognitive processes take place.

The designer, indeed, experiences something in the *external world* (made of the representation outside the designer, variables = $[F, B, S]^e$) through its senses and creates an interpreted representation of what he has perceived. Those interpreted representations constitute the *interpreted world* of the designer (variables = $[F, B, S]^i$). Among what the designer interprets, there are (mental) representations that are built on the designer's previous knowledge as defined targets, i.e. as design goals to be attained or expectations to be satisfied. These last representations are, then, a subset of what is interpreted and that as a whole constitutes the content of the *expected world* (Variables = $[Fe, Be, Se]^i$).

Changes between different worlds occur through a small set of cognitive processes the designer carries out and that can be summarized as follows:

- *Interpretation:* what is sensed in the external world is transformed into a concept rebuilt on the basis of what is perceived;
- *Focusing:* what is conceptualized is then used to create a specific set of goals to concentrate the attention on and thus nurture the emergence of strategies to attain them;
- *Action* or *Transformation:* the generation of effects that aim at perturbing the external world, consistently with the above strategies, in order to attain the desired goals.

Knowledge, to both interpret the world and create expectations out of it, is to be better referred as knowing in situatedness, since memory is seen as a dynamic process and not a static state. Constructive memory, thus, is seen as a process that is close to what was above called interpretation. One builds up meaning about what is sensed and perceived by digging its own knowledge in a push-pull process whose detailed dynamics is explained in Kelly and Gero (2014). The push process is initially driven by the data sensed in the external world and the pull process is driven by the previous or updated expectations that determine the way one interpreted what is sensed. This process, indeed, occurs between the external and the interpreted world, as well as in the interpreted world itself, as a result of knowledge retrieval from previously conceived expectations. This means that the sensory perceptions of the designer can be just interpreted by means of the knowledge possessed so far, by browsing relevant concepts where the knowledge is "stored", namely constructive memory.

With the extension of the situated FBS framework to needs and requirements (Cascini et al. 2013), the set of cognitive activities characterizing the design process also includes two important elements of the product planning phase in the development process: the identification of needs and the definition of requirements. These activities, as well, show that both need and requirement variables have their correspondents in the three worlds. Needs and requirements exist outside the designer in the external world (variables = [N^e, R^e]). Inside the mind of the designer, in turn, they can be interpreted according to what is perceived and experienced (variables = [N^i, R^i]). Moreover, these interpreted representations also allow the designer to focus on specific objectives or goals to be attained in order to satisfy user needs as well as other requirements (variables = [Ne^i, Re^i]).

Hypotheses About the Applicability of the Situated FBS to Represent User's Cognitive Processes

According to the aim of the present work, it is interesting to investigate the reasoning patterns occurring in the mind of the user before and after the purchase of a product, in order to understand possible reasons for adoption or rejection. A proper interpretation of these reasons can be of vital importance to improve the efficacy of promotional campaigns by marketing professionals, and to suitably orient the design of innovative features in product planning and development.

In this perspective, it is useful to change the reference system of the three worlds the FBS variables are situated, i.e. switching from the point of view of the designer to the user's one. It is expected that this change of perspective allows modeling: the interpretation processes made by the user while interacting with a product; the conceptualization paths that brings to the creation of true or false knowledge; and the emergence of decision elements that finally bring to the adoption or rejection of a product.

Several scholars in Engineering Design have studied the interpretation of products in the user's perspective. These studies range from investigations upon product usability and the necessity to represent use plans as in Vermaas (2006), to the rich literature on product affordances, e.g. Brown and Blessing (2005). Some relevant insights for the objectives of this work can be derived from the studies about perceived affordances: in the words of Norman (1988 p. 219) "affordances result from the mental interpretations of things, based on our past knowledge and experience applied to our perception of the things about us". In turn, user's knowledge acts as a "filter" in the interpretation of the information coming from a product: some information is not perceived, understood, processed.

However, none of these studies explicitly addresses the representation of the processes that, consciously or unconsciously, occur for matching user' own perceived needs with its interpretation of a product. Furthermore, in the perspective of supporting marketing and product planning initiatives, it seems relevant to distinguish between user's sensation of the structural characteristics of a product, his interpretation of product behaviour, the intentions he/she has by purchasing the product and eventually the needs that push him/her to this decision. In other terms, the authors believe that distinguishing between FBS variables can be important not only when describing the cognitive processes occurring in a design activity, but also when analysing symmetrical cognitive processes occurring during purchasing.

The essential assumption of this paper is that the variables F, B, S and N are representative of the objects of thinking in the user's mind while he is reflecting on a product (either while deciding whether to purchase it, or while assessing after a period of usage). Furthermore, those variables can be situated in the same three worlds, despite they must be reconsidered in the user's perspective; the following definitions are adapted from Gero and Kannengiesser (2004):

- the *external world* is the world composed of the reality outside the user;
- the *interpreted world* is the world that is built up in the user's mind in terms of sensory experiences, percepts and concepts; it is the internal, interpreted representation of that part of the external world that the user interacts with; and
- the *expected world* is the world that the imagined actions of the user will produce.

From the lessons learned in the literature on affordances, users recognize affordances from past experience, learn them through interaction with a product and also infer them by analogy. It can be stated that all FBS variables can be recognized, learned and inferred with similar processes. Moreover, users reflecting on a product happen to apply any kind of logical inference, i.e. deduction, induction and abduction. Some evidences of abductive reasoning by a user are showed in the next section.

Towards a Methodology to Recognize User's Needs, Related Expectations and Mismatches

Under the assumption that variables, worlds and processes of the situated FBS framework, suitably adapted to the user's perspective, can represent the cognitive processes in the user's mind when reflecting on a product, the authors intend to build a practical methodology to elicit user's needs and to understand motivations and mechanisms behind the adoption or rejection of a product. The methodology aims at allowing marketing professionals and designers to more precisely infer customers' cognition and clarify their own vision about design targets and objectives, with the purpose of developing products that have higher chances of market appraisal and adoption.

The methodology is still under construction, but its essential elements are worth sharing, as well as their prototypical application to a real product. The basic elements are the following:

- a protocol analysis is applied to user's comments on a product (e.g. derived from an interview within a field test);
- the user's discourse is segmented and coded by identifying the variables of the mentioned objects of thinking;

- the cognitive processes that link the identified variables are recognized with reference to the situated FBS framework also extended to the representation of needs;
- in case of lack of coherence between the recognized processes made by the user and the reference framework (e.g. logical jumps), some inferences can be made on the relevant features of a products that determine dislike and potential lack of adoption of a product; and
- further inferences on product appreciation can derive from mismatches between interpreted and expected values of product variables.

The above steps are not mature enough to be structured in an operational methodology, but are sufficient to describe the logic of the proposed approach. In the perspective of gaining more sensibility on the issues behind the objectives of this research and with the aim of testing the validity of the proposed approach, the authors decided to start some preliminary experimental application of the model.

As the one described in the next section, the tests are conducted through the following steps:

- selection of a prototype or a product with some innovative features (e.g. claimed by the producer) to be assessed by users;
- collection of feedbacks by a sample of users (e.g. through interviews or reviews on the internet);
- protocol analysis of the collected users' responses;
- elaboration of the emerging evidences and discussion with product experts about the relevance and soundness of those results.

An Illustrative Application of the Proposed Model

Through the analysis of interviews carried out with customers and adopters of specific products, the authors intend exploring strengths and weaknesses of the proposed approach. To this purpose, this section presents short excerpts from interviews with customers and customer reviews that are investigated with a "protocol analysis-like" approach. They were found in freely accessible websites, such as www.onlineshoes.xyz, running.competitor.com and www.toeshoes.xyz/. The authors did not apply any modification to the collected text, which results in plenty of jargon and informal expressions.

Customers are expressing their opinion about a pair of sneakers: Nike Free 3.0 v3 (Fig. 1). Nike currently advertises these shoes on its website as "NIKE'S MOST NATURAL RIDE—Our lowest-profile Nike Free running shoe allows your foot to move any way it wants, while still offering the cushioning it needs. Best for those with extensive natural-running experience." Moreover, Nike also suggests using those shoes after you "build your muscles" since "…making the Free your

Fig. 1 Nike Free 3.0 v3

primary training shoe will likely require months of gradual adaptation to gain the strength…”.

The selected interviewee are not occasional runners, but (semi-)professional ones that got in touch with the shoes and experienced them for a limited but not negligible time (thus including the experience after the purchase). These profiles have been chosen so that they can consider the shoes under investigation from two different angles (both as a fashion and as a technical item); to grab different needs they can be able to satisfy.

The following part of the section follows this pattern: a dialogue excerpt from the interview or the customer review; the analysis and a critical discussion of the excerpt according to the codification carried out by means of the variables and the worlds characterizing the situated NRFBS. It is worth recalling that the main goal of this approach aims at grabbing cognition underlying product appreciation or dislike. Interpretations of product features matching or mismatching the pre-conceived expectations represents the key point to start understanding the underlying needs that the customers want the proposed solutions to satisfy. The identification of (mis)matches allows exploring customers’ thoughts to catch the tacit needs there hidden.

First Respondent—Interview to Tim Wunderlich—Decathlete

Interviewer: And how did you find the feel of the shoe? Did you notice anything? What are the differences to a normal running shoe would you say?
Interviewee: I think that’s what sets the Nike Frees apart (from other running shoes) is that they have a very comfortable feel to them. (…) Take as example the converse all stars which (…) don’t have an ergonomic feel to them as in you feel that (they have) just a very plain sole. If you compare that to the Nike Frees you’ll notice that, especially when walking for a long distance, there is a noticeable difference in that they are very comfortable and you don’t have a lot of stress on the foot in that sense. (..)

This speech segment shows that the interviewee has worn the shoes and that it resulted in a comfortable feel (B^i). On the contrary, the “very plain sole” of the Converse All Stars (S^i) does not allow for an analogous ergonomic feel (Be^i). This can be considered as a comparison [Evaluation in the situated FBS framework (Gero 1990)] between two different kinds of shoes. The Nike free have been directly tested and interpreted; the described behavior of the Converse All Stars, in turn, is something that has been not experienced, but it is rather inferred (and thus an expectation is behind this inference) from the interpreted structure of the sole, as the interviewee saw it. Still with reference to the original situated FBS framework, one can notice that there is an implicit reference to the interpreted behavior of shoes having plain soles (B^i), that results in the building of the expectation that all the shoes having a plain sole do not allow for an ergonomic feel.

The comparison, then, gets further clarified in the rest of the excerpt, where the interviewee clearly says that the Nike Free are very comfortable (B^i) when one walks for a long distance with them, which should result in less stressed feet, i.e. one of the reasons to wear them (Fe^i). This final part of the speech allows to shed light on one of the reasons behind the adoption of the shoes, which can be linked to not explicitly formulated needs concerning safety and health.

> Interviewee: Yeah, well the thing with using it as an athletic shoe is that it is made to be very comfortable and light. And (…) because I ran a lot and regularly, and because it is not per-say designed for people of my weight, I weigh 95 kilograms, or not designed to be used in such a way (…), I ran on a treadmill. My brother advised me that using it on a hard surface like that, (the shoe) will not provide the dampening you need and after a month or two and that is something that is pretty you know constant from then on until I got new running shoes with a lot more dampening.

This part of speech, differently from the previous one, deals with what does not meet the initial expectations of the customer that bought the shoes. Beyond what reprised from the previous discourse about comfort and weight, the customer clearly recognizes that the shoe is not designed for someone of his weight, and more precisely for someone weighing 95 kg (R^i). The user clearly expresses a requirement in an explicit way in this peculiar case and, immediately after that, he links the requirement to the function he expects the shoes are capable to carry out (Fe^i: to be used regularly and for long runs). In order to reinforce his opinion, he moves from the purpose of the shoes to the way they work, clarifying that there is a clear mismatch between what he would like to have (Be^i) and what he actually gets (B^i), according to his interpretation (“the shoe will not provide the dampening you need”). The last part of the selected speech, if seen under the light of the first part previously commented, implicitly suggests that the next pair of shoes will be bought looking for a match between an expected behavior and the behavior they promise. This can be just done on the basis of the structure he interprets (S^i) from the structure he perceives (S^e) (e.g. by looking at the sole thickness).

With reference to the whole part of speech here discussed, it is worth noticing how the user builds up his own knowledge: he progressively interprets his feelings according to the feedback the shoes give him and postulates a consequent behavior through his knowledge. This is, in other terms, a potential evidence that the user, when choosing a product, practically behaves like a designer, since he hypothesizes some behavior with a process that is neither inductive, nor deductive, but rather abductive. He has no knowledge about the pair of shoes he is taking into consideration, but he interprets their external aspect and projects new expectations that can be just created after the update of its constructive memory.

> Interviewer: How did you find the sizing of the shoe? Did you find the size? Did your size fit you well?
> Interviewee: (…) my brother bought them for my dad, and I just took them from my dad. So it's not like I fitted them. But that being said, they were (size) 12.5 and from my experience and what I know about the nike sizes from my brother is that once you know what size you wear with nike you will be able to buy the same size with other products they are very consistent.

This speech fragment, even if short, clearly describes the process of creating expectations on the basis of the previous experiences that have been already filtered and stored by the customer's constructive memory. Stemming from the size of the shoes (S^e), he shifts the speech to something that is both, even if a little implicitly, referred to the behavior of the shoes and the need they have to satisfy. The expected behavior refers to the wearing properties of the shoe (Be^i). The logic the authors infer from this speech is that the external structure of the shoe (the size, S^e), if properly interpreted (S^i), should lead the user to hypothesize the behavior (B^i, which is not directly interpreted from its external representation, but inferred from the structure with a transformation process). Whenever the user will try the new pair of Nike shoes, he will try to match the above expectations due to its current knowledge with the situation he is experiencing. This comparison will support the evaluation of the new pair of shoes: if the interpreted behavior is consistent with the expected one, the user can more strongly be convinced that the shoes will satisfy his own requests of comfort (Fe^i) that, in turn, hinders the need of avoiding foot pain (Ne^i).

> Interviewee: Um, yeah. I think that is something Nike managed to do very well. It is a, I'd say a sport shoe but (Nike) managed to give it a look that allows people to wear it as a regular sneaker as well. (…) You will spot it as a sport shoe right away, but it is very acceptable to wear it in another context, and that's something that opens up a bit of a market segment for Nike in my opinion.

In this excerpt, there emerges a lot of different information that can be more easily linked to the product appreciation and adoption. Considering the external look of the shoes (S^e), the related interpretation of the structure ("sport shoe", S^i) leads him to consider two different behaviors for the shoes. On the one hand, he interprets them as a couple of shoes to be used in challenging or sporty contexts (B^i), but at the same time they can be suitable for occasions in which one does not need to use sportswear (still B^i). It is not clear if these interpreted behaviors can suggest their transformation into functions and then requirements; however, the above descriptions clearly point out that each of those interpreted behaviors has the chance to satisfy a specific perceived need (Ne^i). Respectively, the need of feeling no pain while running and having the feet covered, the latter related to two basic needs such as protecting oneself from cold and being integrated in the society (with its customs and habits).

In the end, it is worth noticing that all the questions here reported and posed by the interviewer aim at extracting specific elements to capture the voice of the customer. The answers, indeed, can provide both the marketing and design departments with knowledge elements to support product development, so as to get higher chances of appraisal and adoption.

Second Respondent—Review Grabbed from a Shoe-Selling Web Platform

> The only real cons I found in this shoe was that after wearing it maybe ten times, I started to notice that dirt and grime was collecting on the front from wear. This is hard to clean because of the type of material the shoe is made out of.

This customer presents the drawbacks he encountered in using the shoes. The first sentence clearly points out that progressively (and quickly-*after wearing it ten times*-) the shoes get dirty (B^e). This is carried out through the observation of the front part of the shoes that collects dirt and grime (B^e). Moreover, trying to turn them back to clean, he had trouble. He expresses this problem by saying they are hard to clean (B^i) and this is due to the material the shoes are made of (S^i). It is possible to recognize the inference by the user who appoints the responsibility of the difficulty of cleaning (B^i) to the material of the shoes (S^i). In other terms, the user does not realize that the superposed layers of grid-like hollow textiles (S^e) would require a different cleaning tool (e.g. a brush) or process (e.g. suitable to be washed in a laundry machine). Besides, the analysis clearly highlights a mismatch with the user's expectations: he would like to have shoes that do not get dirty so easily (Be^i) or that, in turn, are designed so that one can easily clean them (Fe^i). Overall, this can lead to a double interpretation: it can stand under the strand of hygiene or aesthetical needs (Ne^i).

> Another con is the grip. Although when wearing this, your foot can move in any direction and has a large range of motion, the grip suffers. I feel that it is good, but not good enough. Overall, this shoe offers a lot to the average customer, and if you are willing to fork out the extra money over a more modestly priced shoe, it is worth it!

Here pros and cons are mentioned together in terms of the shoes behavior. The customer positively evaluates the freedom of movement (B^i), meaning that this matches his expectations (Be^i). On the contrary, when the customer says "*the grip suffers… is good, but not good enough"* there is a mismatch between the expected behavior (Be^i) and what he interprets by wearing the shoes (B^i). The next fragment shows a clear reference to the profile of potential customers for this product: those expressing non-professional needs (N^i) might be satisfied (Ne^i), while professional runners might find these shoes not suitable for addressing their expectations (Be^i), especially for what concerns the feeling they have upon running on technical grounds (B^i).

Third Respondent—Review Grabbed from a Shoe-Selling Web Platform

> Granted I'm not the athlete either of these two are, I'd like to think I am doing as much as I can to be as close their level. So one day while shopping, I come across Nike shoes that looked more suited to be rolled up than to support my feet. Upon trying them on however, I was quickly finding out something that our species has forgotten over the last century. We were built to run barefoot, or at least as close to barefoot as possible. I liked what I felt, quite a bit actually. After purchase I quickly went to break them in. Ten minutes later I was panting on the couch.

The first part of the above review clearly shows that the user has the need to be more athletic or in a better fit (Ne^i).

After that, he mentions the Nike free apparently delivers a different function (*to be rolled up* → F^i) than the usual one (support the feet → Fe^i) and this represents a clear mismatch between what is interpreted and what one expects. Despite this kind of mismatch, the respondent decided to give them a try. This shows that mismatches are not necessarily triggers of rejection. This is consistent with recent studies on surprise emergence (Becattini et al. 2015) and with other investigation linking surprise-like reactions with product appraisal (Im et al. 2015). Thereafter, he expresses the realization of a need (*running barefoot* → Ne^i) and the capability of these shoes to satisfy it.

> After slowly building my stamina up to be able to run a mile without stopping, I began to focus on new goals. Being able to do more strenuous exercise without fear of cardiac arrest, the gym was no longer a curse word. Almost 2 years later, I am in the best shape of my life, and still focused on building an even better fitness portfolio. This was all obtainable because of that goofy looking shoe, from the company that started with a waffle maker. The feel of them just made the bend of my foot feel natural, compared to a stiff, locked sole I had been accustomed to. Although there was very little padding, the flow of energy was transferred so easily and efficiently that I barely notice concrete to grass.

The initial part of this fragment refers to a self-defined objective for the respondent that, time after time, managed to improve his athletic condition. Moreover, he also says that he shifts his goal to an even improved situation (*…an even better portfolio*). This sentence implicitly mentions a need that he expects to match in the future (Ne^i). After that, a big leap appears if one just refers to his reported words: two structural elements ("*goofy looking shoes*" → S^i and "*started with a waffle maker*" → S^i, recalling the shape of the sole) are considered capable of satisfying the need. These two fragments also appear to be quite unrelated to each other, but the user's thought becomes clearer with the rest of the selected review. The natural foot bending (B^i that matches with the expected behavior Be^i he has when running) is attributed to the waffle maker structure of the sole (S^i), similarly to what happens with a transformation process in the situated models about designing. In addition, there is also a clear reference to what the customer perceives as a limiting structural factor for obtaining the expected behavior: he expects that a different sole structure (Se^i) can generate exactly the opposite, unsatisfying, behavior. According to the authors' understanding, this exactly represents a clear example of the mechanism behind building expectations as the constructive memory generates new knowledge/knowing with novel experiences. Previous shoes having a uniform and non-segmented sole always behave differently from the Nike Free, resulting in an overall increase of the shoe stiffness (Be^i). Differently from the previous respondent, here it seems that the shoe have a nice grip. So good that one barely notices the difference when running on an artificial ground or on the grass. Probably here a "positive" mismatch appears with what one expects: running on the grass or on concrete should provide different feedbacks (e.g. on the grip) according to what was previously experienced. On the contrary the shoes (and their sole specifically) appear to be particularly effective to reduce these differences between different grounds. Despite it was something not expected, the overall outcome on the customer's appraisal is positive.

> The green grass print insoles striked me odd at first. After coming to understand the barefoot ideals, I had to smile at the notion of running on "grass IN my barefoot" shoes.

This part of speech begins with a clear mention to the observed structure of the shoes (S^e) in terms of color and (apparent) texture. The mismatch here is not explicitly stated, but the customer says that he felt some surprise, suggesting that it is uncommon or completely novel to him to find shoes having an insole like that (Se^i). The interpreted structure, here apparently missing, is somehow made explicit in the already examined part of speech. The customer, in fact, interprets the green insole as something that recall grass (S^i) and the barefoot running he got in mind from the beginning of the analysis.

Discussion on the Emerging Evidences

The above descriptions, as emerged from the analysis of speech fragments or web reviews, can be then characterized in terms of the Needs, Requirements, Functions, Behaviors and Structures as the one proposed firstly by Gero and Kannengiesser (2004) and later on recalled by Cascini et al. (2013). Moreover, clear connections between the classified variables and the worlds that are typical of situatedness, can be more easily inferred by considering the speeches as a whole nucleus of contents. The whole speech fragment, indeed, better shows if the considered sentence refers to the external world or to the interpreted or expected one. Moreover, missing elements emerge in several cases, meaning that it is not always possible to find a direct correspondence with what happens in the situated FBS framework (for designing) in terms of elementary cognitive processes. These elements, however, highlight the topics designers and marketing professional should focus on, in order to better support the development of products having a higher chance of market appraisal. For instance, the considerations about dirt collected on top of the shoes—2nd respondent—might lead to multiple interpretations (hygiene or aesthetical needs?): the proposed approach is expected to provide the greatest benefits precisely in these situations, where the experts involved in the design process needs to find critical issues requiring further clarifications. Moreover, lacks and missing elements in the evaluation/ assessment discourse might help the marketing professionals ask more targeted and precise questions. Furthermore, they should stimulate designers to improve their products, to communicate better their real features and behavior.

The application of the model also confirmed that the satisfaction of expectations is a key for the product acceptability. At the same time, it is not possible to state the contrary. Indeed, the analysis of the above presented parts of speech shows that mismatches between the interpretations and the expectations are critical to both the acceptability and the rejection to products. The customer, indeed, positively feels this mismatch if the interpreted NRFBS variable overcomes its initial expectations. On the contrary, the mismatch creates rejection to the product if it contradicts the expectations.

For what concerns the kind of considered variables, the provided examples show that the users more easily interpret what they directly perceive with their senses, i.e. structure variables, or use-related product feedbacks, i.e. perceivable behavior variables. Moreover, the example also shows that the customers carry out the evaluation by, even naively, transforming the variables they interpret with mechanisms that belong to inference; rather than, deduction and, in some cases, abduction. In fact, they guess that something they don't know can happen if some variables get different values from the ones they have already experienced and interpreted from the external world.

Examples providing the customers' opinion, after a certain amount of time they got in touch with the product they evaluate, show that the use of the product itself allows them to build up their knowledge consistently with the processes characterizing constructive memory. What is not properly interpreted or creates a mismatch between what is known (expected) and what is experienced (interpreted) might trigger a process of knowledge acquisition in that the existing expectations can be confirmed or denied. When a mismatch occurs, new knowledge routes can be hypothesized to find meaning in what is experienced and on this basis new expectations can emerge, as the memory gets enriched with novel concepts.

Conclusions

The user-centered perspective in design is gaining a stronger consideration as a strategy to develop products having a higher chance of appraisal and adoption. To this purpose, the paper critically reviews the main approaches to consider the customer/user's viewpoint during the product planning stage of the product development process. In order to address the identified lacks, the paper proposes a framework whose aim is to allow the different profiles involved in product planning (marketing professionals and designers) to share a common viewpoint on customers' cognition, when dealing with product features in a situated context. The proposed framework relies on the main constructs characterizing the situated FBS framework, considering its extension to Need and Requirements variables. Differently from the situated FBS and its NRFBS extension, it considers the viewpoint of the customers/users instead of designer's one. Hence, regardless of the original intent of the framework to characterize cognition in design, the situated NRFBS here describes the user's cognitive processes behind the interpretation and the potential appreciation/dislike of market solutions. The applicability of the proposed framework has been preliminarily checked by segmenting interviews carried out with people experiencing products so to grab their cognition. Specifically, the interviews refer to the experience that different user had when interacting with a pair of shoes by Nike (Nike Free 3.0 v3).

The results emerged from the application of the proposed framework to investigate user/customers' cognition show that both the constructs of the situated FBS model and its extension to Needs and Requirements allow for a more precise understanding of the dynamics characterizing users' interpretation and the capability of products to match the related expectations. More precisely, the NRFBS variables can be used to characterize what the customer expresses in terms of appreciation or dislike and the main processes described in the situated framework can be used, as well, to describe the main links between variables also from the perspective of a user that evaluates a product. According to the analyzed excerpts of interview and from the web reviews about the Nike Free, it clearly appears that some connections (between variables or world, especially with reference to needs) often remain implicit. Despite this might be seen at first as a limitation of the proposed approach, it represent a significant opportunity. In fact, by the analysis of those missing elements, both designers and marketing professionals can better focus their questions aiming at the capture of the customers' opinion. The proposed approach allows the exploration of unexpressed/implicit needs in a more structured and repeatable way, so that designer and marketing professionals do not just rely on their intuition and talent to understand needs to be satisfied.

The application of the proposed approach, in turn, showed that the characterization of specific part of speech in terms of NRFBS variables and worlds characterizing the situated context still depends on the subjective interpretation of part of speeches retrieved from interviews or reviews. Nevertheless, this appear to be a minor limitation of the approach because it does not hinder the goal of clarifying which aspects deserve to be explored or clarified with higher priority. This preliminary application, using product reviews already available on the internet, can be seen as a proof of concept for the proposed approach and the promising results here presented allow the authors to plan a more intensive deployment. In the next future, field test with directly interviewed potential customers will provide more evidences about the applicability of the approach and the robustness of the characterization of NRFBS design variables.

References

Becattini N, Borgianni Y, Cascini G, Rotini F (2013) Does experience in design and innovation affect the perception of creativity? In: DS 75-7: proceedings of the 19th international conference on engineering design (ICED13), design for harmonies, vol 7: human behaviour in design, Seoul, Korea, 19–22 Aug 2013

Becattini N, Borgianni Y, Cascini G, Rotini F (2015) Surprise as a situated phenomenon. In: DS 80-11 proceedings of the 20th international conference on engineering design (ICED 15), vol 11: human behaviour in design, design education; Milan, Italy, pp 71–80

Borgianni Y, Cascini G, Pucillo F, Rotini F (2013) Supporting product design by anticipating the success chances of new value profiles. Comput Ind 64(4):421–435

Brown DC, Blessing L (2005) The relationship between function and affordance. DETC2005-85017, 24–28 Sept 2005, Long Beach, California, USA

Brunswik E (1952) The conceptual framework of psychology. (International Encyclopedia of Unified Science, Volume 1, Number 10). The University of Chicago Press, Chicago

Calantone RJ, Di Benedetto CA, Divine R (1993) Organizational, technical and marketing antecedents for successful new product development. R&D Manag 23:337–351

Cantamessa M, Montagna F, Cascini G (2016) Design for innovation: a methodology to engineer the innovation diffusion into the development process. Comput Ind 75:46–57

Cascini G, Fantoni G, Montagna F (2013) Situating needs and requirements in the FBS framework. Des Stud 34(5):636–662

Cooper RG (1999) From experience: the invisible success factors in product innovation. J Prod Innov Manag 16(2):115–133

Cooper RG, Kleinschmidt EJ (1987) New products: what separates the winners from the losers. J Prod Innov Manag 4(3):169–184

Dolan RJ (1990) Perceptual mapping: a manager's guide. Harvard Business School Background Note 590-121, July 1990

Gero JS (1990) Design prototypes: a knowledge representation schema for design. AI Mag 11(4):26–36

Gero JS, Kannengiesser U (2004) The situated function-behaviour-structure framework. Des Stud 25(4):373–391

Gray D, Brown S, Macanufo J (2010) Gamestorming: a playbook for innovators, rulebreakers, and changemakers. O' Reilly, Sebastopol

Griffin A, Hauser JR (1993) The voice of the customer. Market Sci 12(1):1–27

Im S, Bhat S, Lee Y (2015) Consumer perceptions of product creativity, coolness, value and attitude. J Bus Res 68(1):166–172

Kano N (1984) Attractive quality and must be quality. Hinshitsu (Quality) 14(2):147–156

Kelly N, Gero JS (2014) Situated interpretation in computational creativity. J Knowl-Based Syst 80(3):48–57

Lee Y (2008) Design participation tactics: the challenges and new roles for designers in the co-design process. Co-Des 4(1):31–50

Loudon DL (1988) Consumer behaviour: concepts and applications. McGraw Hill, London

Maslow AH (1987) Motivation and personality, 3rd edn. Harper & Row Publishers Inc, NY

Norman DA (1988) The psychology of everyday things. Basic Books, Inc

Rein GL (2004) From experience: creating synergy between marketing and research and development. J Prod Innov Manag 21:33–43

Sheth JN, Mittal B (2004) Customer behavior: a managerial perspective, 2nd ed. Thomson/Southwestern Publishing

Slater SF, Mohr JJ (2006) Successful development and commercialization of technological innovation: insights based on strategy type. J Prod Innov Manag 23:26–33

Tapp SR (1984) Brunswik's lens model: a review and extension to consumer research. In: Kinnear TC (ed) NA—advances in consumer research vol 11, Ass for Consumer Research, Provo, UT, pp 103–108

Vermaas PE (2006) The physical connection: engineering function ascriptions to technical artefacts and their components. Stud Hist Philos Sci 37(1):62–75

A Means to Characterise and Measure the Information Processes of Engineers Using Eye Tracking

Duncan Boa and Ben Hicks

Abstract How engineers use and process information during design has primarily been investigated using "think-aloud" studies. However, self-reporting thought process affects a task and introduces several associated biases that lead to a general lack of commensurability. Some of these issues can be addressed by using passive observation techniques, such as eye tracking, and standardised information sources to investigate information processing behaviour. Eye tracking is a powerful research tool, from which inferences about information processing can be made based on someone's gaze. In this paper a series of fundamental information processes based on gaze, *Information Operations*, are characterised and evaluated in an eye tracking experiment with 42 trainee engineers. It has been demonstrated that Information Operations are distinguishable using gaze and can be used to characterise information processes of engineers. The findings and corresponding operations offer a potentially novel means for real-time support of information activities, compliance checking, and characterising information use.

Aim

The aim of this paper is to characterise a series of information processes used by engineers using eye tracking and their gaze.

Significance

Engineering design is commonly viewed as an information processing activity during which the information management skills of engineers heavily influences the finished product's quality (Pahl et al. 2007; Baya and Leifer 1997, p. 151).

D. Boa (✉) · B. Hicks
University of Bristol, Bristol, UK
e-mail: dr.boa@bristol.ac.uk

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_31

Designers act as information processors in this process, solving problems that contribute to the development of a product. To assist engineers during design and improve product quality, information management tools and techniques have been developed. However, broad information requirements and poor understanding of how engineers fundamentally process information hinders attempts to support them in this critical activity (Dieter and Schmidt 2012, p. 151).

The information processing behaviour of engineers has primarily been investigated using "think-aloud" studies. Inferences about an engineer's information processing, including information assimilation and evaluation, and knowledge structuring, are made from transcripts of their thought process. The commensurability of these results can be problematic for a number of reasons (e.g. Baykan 1997; Cross 2011; Ullman et al. 1988):

- Protocol analysis relies on self-reporting, potentially interfering with the task
- Processes are specific to the design task undertaken in the experiment instead of the design information
- Difficulty consistently coding data.

In addition to these issues specific to protocol analysis there is also an over-reliance on case studies and small sample sizes to characterise information processes of engineers. This reduces overall confidence in the findings, as there is a lack of discussion as to how variable information processes are between individuals.

Eye tracking is an example of a passive observation technique from which inferences about information processes can be made (Duchowski 2007; Holmqvist et al. 2011). The technique offers several advantages over protocol analysis in that it interferes with a person minimally, produces accurate and precise data that can be analysed readily, and is based on the established link between gaze and cognition i.e. where someone is looking is genuinely attributable to what they are thinking about (during constrained tasks). Some of the disadvantages of eye-tracking is that equipment is relatively expensive, gaze is inherently idiosyncratic, and isolating causal effect can be challenging.

The view of Newell and Simon that (in constrained problems) humans are representable as information processing systems is adopted in this paper (Newell and Simon 1972). A series of fundamental information processes based on gaze, *Information Operations,* are proposed and evaluated. Five fundamental operations that are regarded as relevant to the design process are characterised in this study: *familiarisation*, *finding*, *comprehension*, *review selection* and *review critique* (Boa and Hicks 2014).

Measuring how engineers process information in a passive, quantifiable and repeatable manner can be used in a number of situations. Gaze-based real-time support of information activities, such as checking orthographic drawings, could be used to determine adaptive interfaces.

Method

It is the hypothesis of this paper that the gaze responses to a series of stimuli tasks can be used to characterise Information Operations. To test this an experiment is described in the remainder of this paper involving trainee engineers and standardised engineering information sources. The participants are required to interact with engineering information on a computer and answers questions designed to stimulate each Information Operation whilst their gaze is recorded. The results are analysed to determine if Information Operations can be distinguished using gaze data.

Isolating causal effects of gaze is challenging as a wide range of factors influences eye movements. To isolate the causal effect of gaze the following must be considered: the stimuli over which gaze occurred, the task the gaze is a function of, the means of measurement (eye tracking equipment), and the individual participating. These factors are discussed in this section.

Stimuli Information

To isolate an effect on gaze associated with a stimuli task, stimulus information needs to be adequately controlled for information form, quantity, and structure, as these are all known determinants of gaze. For the information processes to be relevant to engineering the stimuli information must also be suitably representative of those found in engineering design.

One of the largest determinants of gaze is information form, which describes how information is encoded in an information source. For example, geometry of a component can be encoded as a series of dimensions in a table or as an orthographic drawing. The information is fundamentally the same but the form in which it is represented, as text or as a drawing, significantly affects the eye-movements used to process it. This is important, as the information process of an individual examining the geometry may be the same (e.g. comprehension) for both the text or drawing, but the corresponding gaze behaviour will be substantially different. Designing information sources to include only a single type of information form will isolate this causal contribution to gaze.

Experimental information sources have been developed to include only a single type of information form. Hubka and Eder's classification of design representations is used to classify the information forms and determine what types of engineering information corresponds to each (Hubka and Eder 1982, p. 76). They are:

1. Iconic representations that record the visualisation of the original in true form, as sketches, drawings or physical models.
2. Symbolic representations using assumed or conventional symbols i.e. language, mathematics, and analogues.
3. Diagrammatic representations, such as graphs, schematics or diagrams for representing relationships.

Diagrammatic forms of information lie on a spectrum somewhere between symbolic and iconic, as graphs and other *diagrams* often use a combination of symbols and visual elements. However, within engineering it is considered that diagrammatic information is closer to iconic. Iconic and diagrammatic representations are accordingly combined to include example representations from both and are referred to from here on as iconic.

A total of 22 stimuli information sheets were used in the experiment, 11 for symbolic and 11 for iconic. The stimuli information sheets are highly controlled for information contents, quantity and structure as these are all known gaze determinants. The symbolic information sheets are based on bearing selection and include a combination of text, calculations, and tables (with one diagram). The iconic information sheets are based on shaft design and contain a combination of orthographic drawings, graphs, and free body diagrams. Development of the information sheets are described in full in Boa and Hicks (2014) with examples of stimuli symbolic and iconic information sheets provided in Fig. 1.

Information Operations and Stimuli Tasks

Within the literature there are a number of information process identified that engineers perform during design. However, the information processes are usually derived from analysing "think-aloud" protocols with little explanation as to how to apply the coding schema to different design tasks.

Synthesised from existing literature on design information processes, and potentially measureable with eye-tracking, are *Information Operations*. The operations are primitive information processes considered applicable to general engineering information use but do not include generation or creation operations (a limitation of using eye-tracking to measure them). Quantifying the operations using gaze measures addresses the issue of how to distinguish them without relying on an observer's interpretation of a thought transcript. The operations are: *familiarisation*

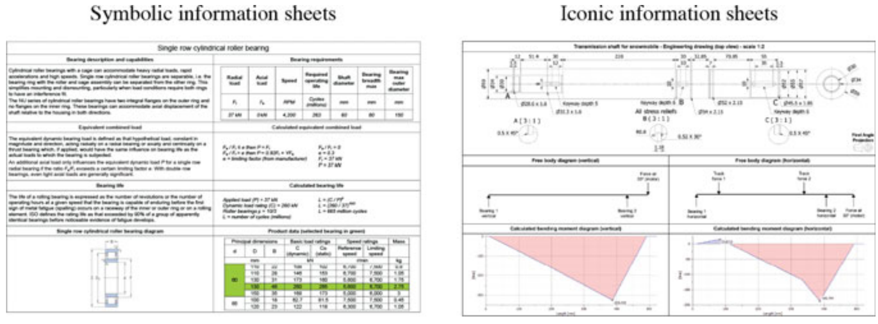


Fig. 1 Example symbolic and iconic information sheets used in the experiment

(FAMI)*, finding* (FIND)*, comprehension* (COMP), *review critique* (REVC), *and review selection* (REVS).

To promote behaviour in line with each Information Operation and generate corresponding gaze responses, a series of stimuli tasks have been developed (Boa and Hicks 2014). The tasks require participants to answer a series of basic questions regarding the aforementioned information sheets. Stimulus of each Information Operation is achieved by the question, and the manner in which the question is asked. Questions relate to a specific information sheet, with all the necessary information to answer the question contained within that sheet. Participants input their answer via multiple choice on a computer, selecting the correct response from a possible six. The multiple choice responses are designed for varying levels of plausibility and an option to indicate they do not know. Two questions for each symbolic and each iconic information sheets have been formulated. Participants have their gaze recorded as they view the information sheets and attempt to answer the corresponding questions. The order in which participants view the sheets is randomised to control for potential ordering and learning effects.

Information Operation tasks are undertaken twice (except for Review selection) with separate information sheets and different questions for each time. This increases the size of the data set and allows for testing that gaze behaviour is robust when the same stimuli Information Operation task is repeated. Repeating a task has the potential for introducing a minor learning effect but this is deemed necessary to validate the findings. A summary of the Information Operations and stimuli tasks is given in the remainder of this section.

Familiarisation is the acquisition of information on a subject area or topic where a specific goal has not, or has yet to be set/developed (Simon 1996, p. 127; Günther et al. 1997, p. 122; Dorst and Dijkhuis 1997, p. 266; Visser 2006, p. 56). To stimulate the operation participants are informed that they will be asked a series of questions after viewing an information sheet, without clarification as to the nature of the question. After 30 s viewing the sheet a simple question is asked e.g. the type of bearing described by the sheet, or the number of transmission elements on the shaft.

Finding is the process of targeted information acquisition for a specific goal (Çukur et al. 2013; Dieter and Schmidt 2012, p. 158; Günther et al. 1997, p. 122; Hurst 1999, p. 142). To stimulate the operation participants are presented with a question prior to viewing the sheet (for 30 s) requiring them to locate an element of information e.g. the temperature limit of the bearing, or the length of the keyway on the shaft.

Comprehension is the learning of information with understanding (Simon 1996, p. 101; Pirozzolo and Wittrock 1981). To stimulate the operation participants are informed that they will be asked a series of questions after viewing the sheet that requires detailed understanding of the sheet contents e.g. what is the risk of exceeding the bearing rated speed. The trial duration is 100 s for symbolic and 60 s for iconic.

Review critique is when information beyond that which is presented in the external environment is assessed for its for its quality, validity or appropriateness

(Pahl et al. 2007, p. 58; Birmingham et al. 1997, p. 109; Hubka and Eder 1982, p. 32; Baya and Leifer 1997, p. 155; Ullman et al. 1988). To stimulate the operation participants are asked to determine the appropriateness of the component on the sheet for which the requirements are presented within the same stimulus. Gaze is recorded during stimulus viewing for which the trial duration is the same as *Comprehension*.

Review selection is when information in the external environment is assessed for its quality, validity or appropriateness against explicitly stated requirements or constraints (Pahl et al. 2007, p. 58; Birmingham et al. 1997, p. 109; Hubka and Eder 1982, p. 32; Baya and Leifer 1997, p. 155; Ullman et al. 1988). To stimulate the operation participants are asked to determine which out of a possible three candidate components is the most appropriate. Each candidate component is presented on a separate stimulus with the identical requirements repeated across each stimulus. Gaze is recorded during stimulus viewing for which the trial duration is the same as Comprehension.

Gaze Measure—Saccade Amplitude Fixation Duration Ratio

The range and types of movements exhibited by the eye are classified into gaze events, with fixations and saccades making up the majority by duration. A fixation is a period of time over which the eye remains relatively still and typically lasts for 200–300 ms (Holmqvist et al. 2011, p. 21). A saccade is the fast ballistic movement of the eye to relocate gaze and is measured in degrees of amplitude attenuated in the scene (Gilchrist 2011). Typical ranges of amplitude are between 2° and 20°.

In this paper the ratio between saccade amplitude and fixation duration (the SF ratio) is used to distinguish Information Operations. Using the SF ratio combines the two core measures of fixations and saccades and approximates how long someone is looking at somewhere and the distance their eyes have moved since the previous fixation. This is beneficial as reading text generates a distinct SF ratio pattern that is relatively consistent.

Baseline Behaviour—Reading Control

Gaze is affected by a number of determinants (as discussed in section "Stimuli information") that makes generalising values of gaze measures difficult. To reconcile this a common information processing activity that can be used as a baseline behaviour is sought.

Reading is a well-understood information processing behaviour with a substantial corpus of knowledge on the topic. By relating Information Operation gaze values back to the baseline the relation between the two can be characterised. Reading of a piece of technical engineering text (on bearing installation) is used to compare symbolic information interaction against. An equally suitable control activity for iconic information is not so clear. Generic "scene-viewing" is generally used to refer to interacting with iconic information, but the definition of what constitutes the activity is varied. In the absence of an equivalent baseline for iconic information, reading will be used again to compare differences in gaze measures.

Participants

Participants were recruited from the University of Bath Mechanical Engineering department with a total of 42 individuals taking part in the experiment. The group was comprised of 39 males, and 3 females with a mean age of 22.4 years (2.5 SD). Of the participants 28 reported minor issues with their vision (predominantly the use of contact lenses), 40 spoke English as a first language, and 28 had at least one year of continuous industrial experience.

The participating students have a common first 2 years of education for all degree specialisations. Accordingly, the information sheets and stimuli questions have been suitably designed to take this into consideration.

Equipment

The experiment has been undertaken with a Tobii TX300 eye-tracker and Tobii Studio version 3.2. The Tobii TX300 is a remote binocular eye-tracker with an integrated 23" high definition display (1080p). Eye-movements are sampled at 300 Hz and the positional accuracy is 0.4° under ideal conditions. The 'dark pupil' response method is employed to measure eye movements. A recording sample threshold of 93% has been used for participant's gaze data to be included in the analysis, which represents a balance between data quality and quantity. Gaze events have been classified using the default IV-T fixation detection algorithm, as implemented by Tobii Studio version 3.2.

Results

Eye-movements are affected by a wide range of factors, including some that correlate to a task (e.g. presentation order) and others that may vary between individuals (e.g. corrective eyewear). In investigating the relationship between Information Operations and eye-movements, failure to control for these factors will bias any estimates of the gaze measures. In practice, this would mean the net-effect of all gaze-affecting factors being estimated rather than an isolated Information Operation effect. Any subsequent characterisation of the Information Operation model would therefore also be biased.

Gaze-affecting factors correlated with Information Operations are controlled for in the experimental design. For example, stimuli information sheets are similar in information quantity, content and structure, and the order in which tasks are viewed is randomised to remove their effect on gaze. In addition to gaze determinants (e.g. corrective eyewear), residual factors not controlled for may still bias any estimates and thus need to be accounted for in the analysis.

Regression modelling is used for this reason because of its ability to isolate the effect and avoid potential for bias in the estimate of that relationship. Consequently, the contribution to the mean of the outcome variable can be estimated for each Information Operation and used as the basis of Information Operation discrimination. Additionally, gaze measures are highly variable and regression models can be used to improve the ability to detect an effect given the participant sample size.

In this section the gaze behaviour of participants is explored to highlight potential issues with the data set prior to formulating a statistical model.

Data Description, Distribution and Trends

The SF ratio for symbolic information Operations is between 15 and 35% lower than the SF ratios for iconic information interaction (statistically significant at $p < 0.05$) (Table 1). The standard deviation for between persons is an order of magnitude lower than for within person for symbolic and iconic information interaction (Table 1). This suggests that the nature of the SF ratio “randomness” is generalizable between people. If a model of an individual participant’s behaviour can be sufficiently fitted, which is not a trivial task, then the model should be applicable to other individuals too.

The distribution of the SF ratio is estimated using the Epanechnikov kernel function (as implemented in Stata 13.1). Time trends of the SF ratio are estimated using a locally weighted Lowess regression.

Table 1 Summary statistics of SF ratio for symbolic (S) and iconic (I) information operations (IO)

IO	Mean		SD		Within person SD		Between person SD	
	S	I	S	I	S	I	S	I
CTRL	0.0099	–	0.0142	–	0.0141	–	0.0018	–
FAMI	0.0205	0.0274	0.0259	0.0315	0.0257	0.0313	0.0027	0.0039
FIND	0.0189	0.0217	0.0221	0.0246	0.0220	0.0244	0.0028	0.0040
COMP	0.0182	0.0244	0.0247	0.0276	0.0246	0.0274	0.0025	0.0032
REVC	0.02	0.0275	0.0270	0.0329	0.0268	0.0327	0.0030	0.0038
REVS	0.0188	0.0276	0.0260	0.0332	0.0259	0.0330	0.0026	0.0037

CTRL reading, *FAMI* familiarisation, *FIND* finding, *COMP* comprehension, *REVC* review critique, *REVS* review selection

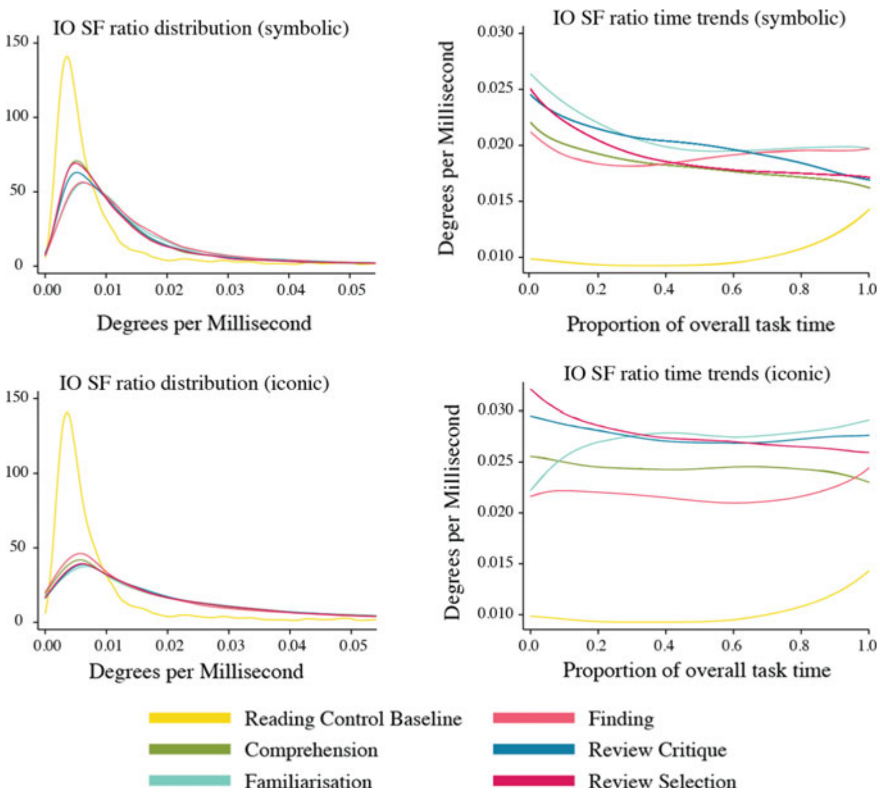


Fig. 2 Comparison of SF ratio between information operations

Overall distributions of SF ratio values for individual Information Operations are clearly distinguishable from the baseline activity of reading but are less so from each other (Fig. 2). Time trends for symbolic and iconic information interaction differ from one another for Information Operations with apparent first and second order time effects (the rate, and rate of change, differ over time for the mean SF ratio).

Regression Model

The conditional expectation function for the SF ratio for a given participant, time, and information sheet is given in Eq. 1. The corresponding coefficients for each explanatory variable for the SF Ratio are estimated using OLS linear regression. The model is fitted separately for symbolic and iconic information interaction samples and is formulated based on the data exploration in the previous section and known gaze determinants.

$$Y_{i,t,s} = \beta_1 i + \beta_2 T_{i,t,s} + \beta_3 t + \beta_4 t^2 + \beta_5 T \cdot t + \beta_6 T \cdot t^2 + \beta_7 R \cdot t + \beta_8 R \cdot t^2 + \varepsilon_{i,t,s} \tag{1}$$

where, $Y_{i,t,s}$ is the SF ratio, for a given participant, time, and information sheet, i is the participant, t is the percentage time elapsed, T is the Information Operation task, s is the information sheet, R is the repeat of an Information Operation task, $\varepsilon_{i,t,s}$ is the error, for a given participant, time, and information sheet.

The SF ratio is approximated as a normal distribution by taking the data in natural logs. Outliers have been removed by trimming the data set to the 99th percentile (median 0.01° ms^{-1}, 95th percentile 0.08° ms^{-1}, 99th percentile of 0.13° ms^{-1}).

Information Operation task is interacted with first and second order time terms in the SF ratio regression model. Doing so will isolate the differences in the change, and rate of change, of the SF ratio for each Information Operation that are observed in the time trends in Fig. 2. The magnitude, and the rate of change of the magnitude, of the SF ratio is likely to change with time according to which Information Operation is being performed. For example, *familiarisation* is likely to be characterised by high values of the SF ratio throughout the trial as a person continues to make large amplitude movements followed by short fixations. This would be exhibited as a high first order coefficient and low second order coefficient.

A first and second order time term in the SF ratio regression model is included to isolate the effect of viewing a repeated information sheet. It is anticipated that recognising elements of repeated information sheets will affect behaviour. Inclusion of the repeat variables within the regression model will estimate the size of this effect on the SF ratio.

Individual participant time trends could be included in the regression model to account for the fact some people's behaviour will change with time e.g. some will get tired or bored quicker than others. However, as the participant sample size is small, and it is not the primary aim of this study, it would be difficult to identify these effects with the current data set. However, if there are large differentials in individual participant time trends this could potentially bias any estimate of mean SF ratio.

It is assumed that variables unaccounted for in the model are more likely to be similar when the same person is looking at the same information sheet, than when different people are looking at different information sheets. Accordingly, it is assumed that the variance of the error term is different for each person for each sheet, with estimations of SF ratio based on this assumption. The alternative if this assumption is not made is that the variance of the error term is the same for all people looking at all information sheets, which is unlikely, and increases the chance of overestimating the significance of any results.

Regression Results

All standard errors are displayed in square brackets, with asterisks indicating the following significance levels: ***$p < 0.01$; **$p < 0.05$; *$p < 0.1$. Regressor coefficients are interpretable as being the approximate percentage difference to the SF ratio for the baseline activity of reading (with the exception of Time, Time^2 which are the baseline SF ratio coefficients).

In Table 2 it is observed that the SF ratio rises over the trial duration for the baseline activity of reading (Time, Time^2) for both symbolic and iconic information interaction. In Table 3 the effect of viewing a repeated information sheet (symbolic or iconic) on the SF ratio is small for first and second order time effects (Repeat#Time, Repeat#Time^2) and is smaller for symbolic information than for iconic (Repeat#Time, Repeat#Time^2, Table 3).

For symbolic information interaction the effect of Information Operations on the SF ratio is large compared to the baseline activity of reading (Task, Table 4). All Information Operations affect the SF ratio significantly. Over time all Information Operations have a higher SF ratio that falls at a greater rate compared to the baseline of reading (Time#Task, Time#Task^2, Table 4). First order time effects are not significant for *familiarisation*.

The SF ratios for iconic Information Operations are larger than for symbolic, compared to the baseline activity of reading (Task, Table 4). Time trends for iconic Information Operations indicate a higher SF ratio that is falling over time at a greater rate compared to the baseline activity of reading (Task#Time, Task#Time^2, Table 4).

Mean SF ratio confidence intervals, overall and over time, overlap to a degree for the Information Operations but differ for symbolic and iconic information interaction (Fig. 3). Marginal effects for the SF ratio show that c*omprehension* is distinct from *familiarisation* and *finding* for symbolic interaction (Fig. 3). For iconic information interaction overall marginal effects indicate that *finding* is distinguishable from all of the other operations. The baseline activity of reading is again clearly different to the Information Operation tasks with a substantially lower SF ratio.

Table 2 Reading baseline coefficients

Time		Time^2	
Symbolic	Iconic	Symbolic	Iconic
−1.048 ***[0.145]	−1.060 ***[0.143]	1.303 ***[0.143	0.542 **[0.266]

Table 3 Coefficients for the effect of viewing a repeated information sheet

Repeat#Time		Repeat#Time^2		Constant	
Symbolic	Iconic	Symbolic	Iconic	Symbolic	Iconic
0.0202 [0.0783]	0.105 [0.111]	−0.0276 [0.103]	−0.131 [0.144]	−5.071 ***[0.0491]	−5.044 ***[0.0519]

Table 4 Symbolic and iconic information regression coefficients for each information operation

Info. operation	Task		Task#Time		Task#Time^2	
	Symb.	Icon.	Symb.	Icon.	Symb.	Icon.
FAMI	0.814 *** [0.0534]	0.808 *** [0.0502]	0.140 [0.252]	1.004 *** [0.266]	−0.582 ** [0.240]	−1.231 *** [0.277]
FIND	0.630 *** [0.0498]	0.782 *** [0.0671]	0.761 *** [0.255]	0.327 [0.312]	−1.025 *** [0.258]	−0.705 ** [0.301]
COMP	0.567 *** [0.0372]	0.816 *** [0.0500]	0.651 *** [0.198]	0.542 ** [0.266]	−1.054 *** [0.205]	−0.895 *** [0.265]
REVC	0.633 *** [0.0367]	0.911 *** [0.0438]	0.681 *** [0.185]	0.602 ** [0.234]	−1.127 *** [0.186]	−0.986 *** [0.249]
REVS	0.595 *** [0.0385]	0.967 *** [0.0401]	0.469 ** [0.184]	0.392 *[0.215]	−0.851 *** [0.183]	−0.868 *** [0.215]

Standard errors in square brackets. Significance indicated by *

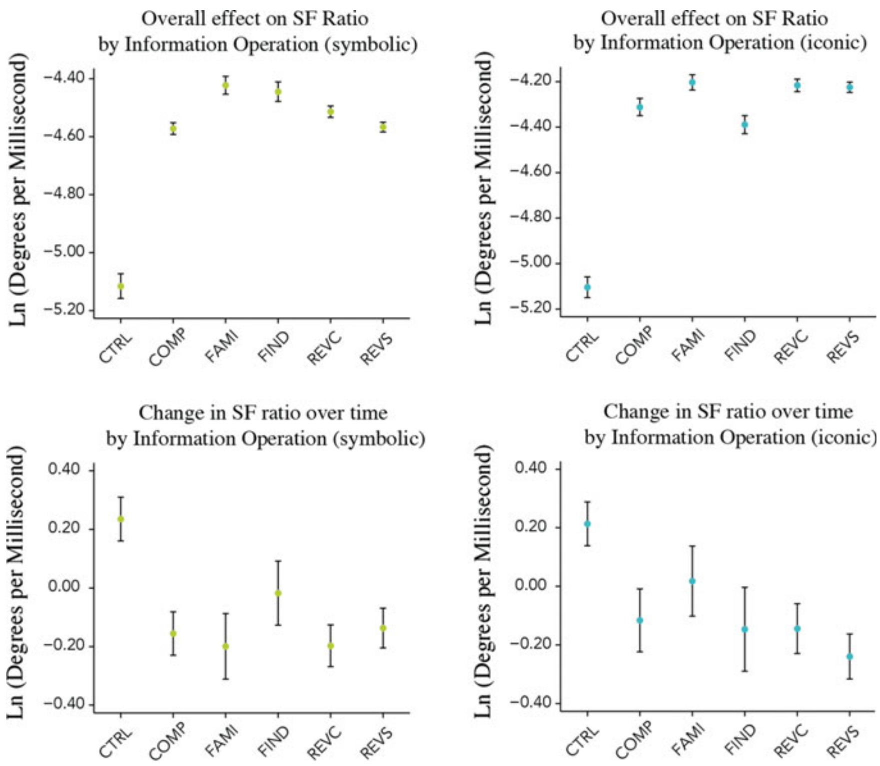


Fig. 3 Marginal effects (overall and over time) in natural logs for SF ratio. The centre point indicates the estimated mean, with bars for the 95% confidence interval plotted along the y-axis around the mean

Table 5 Marginal effects (overall and over time) transformed from log to levels for SF ratio

Information operation	Overall (degrees per ms)		Degrees per ms per % time	
	Symbolic	Iconic	Symbolic	Iconic
CTRL	0.009	–	2.0	–
FAMI	0.019	0.027	1.3	1.8
FIND	0.019	0.022	1.5	1.5
COMP	0.016	0.024	1.3	1.6
REVC	0.017	0.026	1.3	1.5
REVS	0.016	0.026	1.4	1.4

The confidence limits for how Information Operation vary with time overlap too much to be able to discern individual operations from one another based on this measure. However, all of the Information Operations, for symbolic and iconic information interaction, are significantly different to the activity of reading.

To aid interpretation marginal effects are transformed from log to absolute levels in Table 5.

Discriminating different forms of information interaction is challenging using eye movements as they are highly idiosyncratic and prone to significant variation. Combining saccade amplitude and fixation duration into a single ratio measure improves the ability to discriminate the Information Operations based on the explanatory power of the regression model. The SF ratio values of R-squared are improved by approximately double compared to comparable regression models for the base measures of fixation duration and saccade amplitude (not reported in this paper) (Table 6).

Regression post-estimation specification tests (Ramsey reset) return significant values for symbolic and iconic information interaction indicating missing regression terms in the model. The significant values combined with the low R-squared values suggest that the issue is as a result of the idiosyncrasy of the underlying

Table 6 Summary of regression models for SF ratio and corresponding regression models for the base measures of fixation duration and saccade amplitude (split by symbolic and iconic)

	Fixation duration		Saccade amp.		SF ratio	
	Symb.	Icon.	Symb.	Icon.	Symb.	Icon.
Observations	108.911	62.686	104.452	60.358	98.419	56.537
R-squared (%)	3.3	3.6	3.1	6.5	4.5	7.5
RMSE	0.424	0.47	0.851	0.924	0.953	1.075
No. of clusters	411	376	411	376	411	376

Table 7 Ramsey reset results for missing terms in the SF ratio regression model

Outcome measure	Info. type	F statistic Ramsey reset	*p* value
Ln SF ratio	Symbolic	2.77	0.04**
Ln SF ratio	Iconic	3.18	0.02**

measures (saccade amplitude and fixation duration) of the SF ratio (Table 7). A discussion of how the model could be improved is made later in this paper.

Analysis of information interaction from the regression results demonstrates that Information Operation stimuli tasks have an effect on eye movement. The effect is moderate and there is greater within person variation than between person variation. Styles of information interaction, based on ability or expertise, may potentially reduce the variation revealing participant specific trends. However, clustering of individuals into information interaction based groups remains an area for future work.

Discussion

Cognitive interpretations of the SF ratio are made in this section along with a broader discussion about the generalizability of Information Operations and their use in characterising information processing.

Improving Gaze Characterisation

A number of issues with the experimental procedure were encountered in the course of this research. Some are likely to have impacted the results by adding additional "noise" and confounding isolation of a gaze-task effect.

Fixed trial durations were employed to limit the maximum duration of any single recording sessions (typically 45 min), avoiding potential for fatigue (a known gaze determinant). However, more able participants frequently completed tasks early and responding being unoccupied for the remainder of the trial duration. During this time participants reported re-checking answers or simply being bored. Self-determined trial durations are important for isolating the correct gaze response to a stimuli task. Unfortunately this is not realistically feasible within the experimental software that was used and could potentially lead to excessive recording session lengths introducing fatigue. A more streamlined procedure and custom software could resolve this issue.

Eye movements are highly idiosyncratic and the degree of variation observed in the fundamental gaze measures of fixation duration and saccade amplitude is substantial. Corresponding R-squared values for the regression models (goodness of fit) are relatively low with large confidence intervals for estimated conditional means. However, data processing techniques, such as filtering to generate the SF ratio, lead to a substantial improvement in the predictive power of the models. More sophisticated signal-processing techniques, such as wavelets, and further data processing such as more noise filtering, could improve the power even further. The greater within person variation compared to between person variation suggests that the nature of gaze results should translate between individuals if a good enough model can be generated. Including individual participant time trends within the regression model would be one approach to achieving this.

Cognitive Interpretations of the SF Ratio

The link between eye tracking and cognition can be used to describe information processing by engineers in an alternative manner to the Information Operation construct or other similarly derived classification schema. Existing cognitive interpretations for several gaze measures that relate to information processing exist already within the literature, such the correlation between cognitive workload and pupil dilation.

The SF ratio may similarly be used to characterise information processes directly by inferring cognitive states associated with value ranges. For example, in the two-streams hypothesis, two visual systems exist; the ventral stream is responsible for object recognition and identification, the 'what' stream; the dorsal stream is associated with spatial processing, the 'where' stream (Goodale and Milner 1992). A bottom-up cognitive process is associated with an initial 'orientation' period in scene viewing that later drives more in depth scrutiny of the stimulus. It is theorised that the 'ambient' early orientation processing is linked to the dorsal "where" stream, and the later in depth scrutiny linked to the ventral "what" stream (Goldberg and Kotval 1999; Unema et al. 2005). Subsequently high values of the SF ratio are indicative of a more ambient style of processing whilst low values suggest a more focal style of processing.

In Tables 8 and 9 the SF ratios for symbolic and iconic information interaction data sets are displayed in ascending order. Ambient processing occurs when large saccade amplitudes are followed by short fixation durations and is linked to spatial processing. Focal processing occurs when small saccade amplitudes are followed by longer fixation durations and is linked to object recognition and identification.

Absolute values of the SF ratio that constitute ambient or focal processing are dependent on the stimuli scene. For the size of the display used in the experiment the conditional mean of the SF ratio for reading is 0.009°/ms. At the opposite end of the range *familiarisation* during iconic information interaction has a conditional mean SF ratio three times higher at 0.027 °/ms. Assuming that the control baseline

Table 8 Conditional mean SF ratio for each Information Operation in ascending order, symbolic information interaction

Rank order	Information operation	Mean SF ratio	
1	Comprehension	0.016	More focal
1 (=)	Review selection	0.016	
3	Review critique	0.017	
4 (=)	Familiarisation	0.019	
4 (=)	Finding	0.019	More ambient

Table 9 Conditional mean SF ratio for each Information Operation in ascending order, symbolic information interaction

Rank order	Information operation	Mean SF ratio	
1	Finding	0.022	More focal
2	Comprehension	0.024	
3 (=)	Review critique	0.026	
3 (=)	Review selection	0.026	
5	Familiarisation	0.027	More ambient

task of reading is at the focal end of the spectrum, iconic *familiarisation* is accordingly at the opposite ambient end.

Finding during symbolic information interaction has the joint highest conditional mean SF ratio, and the lowest in iconic information interaction. This suggests that the behaviour of *finding* is similar regardless of information form interacted with. Another potential cause for this is that for *comprehension* and *review* more time spent reading reduces the mean SF ratios relative to *finding*, in which more time is spent skimming.

Symbolic information and the requirement to read text makes the interaction inherently more aligned with focal processing. Iconic information interaction, which is not constrained by word and line spacing conventions, requires greater spatial processing and is more closely aligned to ambient processing.

Within the ambient/focal spectrum the relative alignment of symbolic and iconic Information Operations to ambient and focal processing is not necessarily mirrored. For example, *finding* relative to other symbolic operations is more ambient, but for iconic information interaction it is relatively the most focal. In contrast, *familiarisation* is the most closely aligned to ambient processing for both symbolic and iconic information interaction. This serves to highlight the importance of the stimuli information source on gaze and making any associated interpretations.

Conclusions

Using protocol analysis for measuring and characterising the information processes of engineers is problematic for a number of reasons. Existing research lacks commensurability or the prospect of creating automatic support tools. Eye tracking addresses some of these issues and a means for using the technique to measure and characterise information processing is discussed in this paper.

The proposed model of primitive information processes, Information Operations, is distinguishable using gaze. The operations in conjunction with eye tracking could be used to provide real-time support to engineers such as adaptive computer interfaces that detect comprehension difficulty and suggest helpful information. Compliance checking to determine if a drawing or report has been reviewed could also be potentially implemented.

Acknowledgements The research in this paper has been conducted with the support of the EPSRC funded Language of Collaborative Manufacturing project. Special thanks go to Professor Steve Payne, University of Bath, for loaning the use of the eye tracker and the space to use it.

References

Baya V, Leifer LJ (1997) Understanding information managements in conceptual design. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, London, p 480

Baykan C (1997) Design strategies. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, London, p 480

Birmingham R, Cleland G, Driver R, Maffin D (1997) Understanding engineering design: context, theory and practice. Prentice Hall, Englewood Cliffs

Boa D, Hicks B (2014) Information operations: a model for characterising the information interaction of engineers. In: International conference on human behaviour in design

Cross N (2011) Design thinking: understanding how designers think and work. Berg

Çukur T, Nishimoto S, Huth AG, Gallant JL, Cukur T, Nishimoto S, Huth AG, Gallant JL (2013) Attention during natural vision warps semantic representation across the human brain. Nat Neurosci 16(6):763–770

Dieter GE, Schmidt LC (2012) Engineering design, 5th edn. McGraw-Hill Higher Education, NY

Dorst K, Dijkhuis J (1997) Comparing paradigms for describing design activity. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, London, p 480

Duchowski A (2007) Eye tracking methodology theory and practice, 2nd edn. Springer, London

Gilchrist I (2011) Saccades. In: Liversedge SP, Gilchrist I, Everling S (eds) The oxford hanbook of eye movements. Oxford University Press, Oxford, pp 1–10

Goldberg JH, Kotval XP (1999) Computer interface evaluation using eye movements: methods and constructs. Int J Ind Ergon 24(6):631–645

Goodale MA, Milner AD (1992) Separate visual pathways for perception and action. Trends Neurosci 15(1):20–25

Günther J, Frankenberger E, Auer P (1997) Investigation of individual and team design processes. In: Cross N, Christiaans H, Dorst K (eds) Analysing design activity. Wiley, London, p 480

Holmqvist K, Nyström M, Andersson R, Dewhurst R, Jarodzka H, van de Weijer J (2011) Eye tracking: a comprehensive guide to methods and measures. Oxford University Press, Oxford

Hubka V, Eder W (1982) Principles of engineering design. Butterworth Scientific, London

Hurst K (1999) Engineering design principles. Butterworth-Heinemann, London

Newell A, Simon HA (1972) Human problem solving. Prentice-Hall, Englewood Cliffs

Pahl G, Beitz W, Wallace K (2007) Engineering design: a systematic approach, 3rd edn. Springer, London

Pirozzolo F, Wittrock M (1981) Neuropsychological and cognitive processes in reading

Simon HA (1996) The sciences of the artificial, 3rd edn. MIT Press, Cambridge

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(01):33

Unema PJA, Pannasch S, Joos M, Velichkovsky BM (2005) Time course of information processing during scene perception: the relationship between saccade amplitude and fixation duration. Vis Cognit 12(3):473–494

Visser W (2006) The cognitive artifacts of designing. CRC Press, Boca Raton

Personalised Specific Curiosity for Computational Design Systems

Kazjon Grace, Mary Lou Maher, David Wilson and Nadia Najjar

Abstract The Personalised Curiosity Engine (PQE, pronounced "pique") is a framework for computational design systems that models the curiosity of an individual user. This model is then used to synthesise designs that stimulate that user's curiosity. PQE extends our previous research in modelling surprise and curiosity by adding a model of a specific user to generate designs that are personally creative for that user: novel and valuable for them, but not necessarily for society. We describe PQE as a framework, and then describe Q-chef: a design system applying PQE in the domain of recipe generation with a goal of diversifying its user's diet over time. We evaluate our framework with several simulations of Q-chef components that serve as a proof-of-concept of the role of personalised curiosity modelling in computational design.

Introduction

Modelling curiosity has long been of interest to the computational design community (Saunders and Gero 2001; Merrick and Maher 2009). These models have been applied in a wide variety of contexts for the purpose of motivating and guiding exploratory behaviour in computational design reasoning. We extend this approach with the Personalised Curiosity Engine, or PQE (pronounced "pique"), by simulating the curiosity of an individual human user, rather than by modelling the curiosity of the computational design system itself. To do so we apply user modelling techniques as part of a model of curiosity that guides a design system to generate new artefacts.

The benefit of PQE arises when individuals are motivated to broaden their preferences within a domain, but either have difficulty sustaining that motivation, or insufficient knowledge to know how to get started doing so. PQE generates designs that are simultaneously novel and valuable to its user, aligning with the definition of

K. Grace (✉) · M.L. Maher · D. Wilson · N. Najjar
UNC Charlotte, Charlotte, USA
e-mail: K.Grace@uncc.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_32

creativity in Newell et al. (1959). These designs are unlikely to be creative to society as a whole, but they are personally creative to that individual, echoing the distinction between historical and psychological creativity in Boden (2003). Through exposure to these personally creative artefacts the user's curiosity will be stimulated, and we hypothesise that this will pique their interest and motivate the broadening of their preferences over time. This approach provides artefacts that are sufficiently novel to retain interest but sufficiently familiar to allow comprehension.

PQE builds on previous work in surprise, curiosity and reasoning about goals in design systems (Brown 2012; Macedo et al. 2009; Ortony and Partridge 1987; Grace et al. 2014a, b, Grace et al. 2015). Curiosity is modelled as a preference for moderately surprising observations, coupled with a situational formation of a new goal to explore a specific surprising stimulus. Surprise occurs upon an unexpected observation, one that the system's prior expectations would have considered highly unlikely.

PQE is able to model both diversive (i.e. general novelty-seeking) curiosity and specific (i.e. goal-based exploration) curiosity, based on Berlyne's model of specific curiosity (Berlyne 1966). Diversive curiosity values designs that are moderately novel, for example, using the Saunders and Gero approach (Saunders and Gero 2001) based on the Wundt curve (1874). We model specific curiosity as a situational narrowing of this diversive interest in response to a highly unexpected discovery, following the behaviour observed in humans (Suwa et al. 1999). Unexpected discoveries provide new goals for the design system to synthesise designs that are unexpected *in the same way*. PQE is metacognitive (Cox and Raja 2011) in that it reasons about its own goals through specific curiosity.

Our example application is the design of new recipes that increase nutritional diversity, a factor that is strongly correlated with overall health (Drewnowski 1997). We describe Q-chef, the "curious chef", a PQE-based design system for synthesising new recipes that broaden a user's diet over time. We have implemented a prototype of Q-chef's curiosity model, and demonstrate its ability to model a simulated user's curiosity, create exploratory goals, and then generate recipes based on those goals. We simulate users with different familiarity with different subsets of our recipe database, and Q-chef then designs recipes to stimulate their curiosity. The Q-chef prototype serves as a proof of concept of the PQE hypothesis that preferences and familiarities can be used by a design system to simulate and stimulate the curiosity of an individual.

Background

The background for our approach to personalised curiosity and recipe generation draws on research in computational curiosity and creativity as well as research into recipe generation as a domain for computational design.

Curiosity and Creativity in Computational Design

The notion of curiosity has been used to refer both to a trait and a state (Berlyne 1966). Curiosity-as-trait refers to an innate ability possessed by different people to different degrees, while curiosity-as-state refers to a motivation possessed by people to seek novel stimuli. The latter definition, curiosity-as-state, is most relevant to computational design as a motivation for exploratory behaviour in designing (Saunders and Gero 2001; Merrick and Maher 2009).

Developmental psychology has long supported the notion that the acquisition of new knowledge is dependent on past exposure to sufficiently similar knowledge. In education and the psychology of child development, Vygotsky (1978) conceived of the "Zone of Proximal Development": the region of adjacent knowledge approachable by the learner given their current knowledge. Seeking stimuli within this zone is known as curiosity, and new knowledge expands the zone outward, leading to yet more curiosity about newly adjacent knowledge. We propose that the gradual broadening of preference as an individual becomes more experienced with a domain follows an analogous adjacency principle. This iterative developmental cycle is the cognitive inspiration for the Q-chef system, and the grounding of our hypothesis for the effectiveness of diversity as a goal in an HBC system.

Berlyne (1966) proposes that state curiosity can be further subdivided by a two-axis model. The first axis distinguishes perceptual from epistemic curiosity: the former is the drive towards novel sensory stimuli, and the latter is the drive to acquire new knowledge. While both epistemic and perceptual curiosity are possible in a culinary domain, Berlyne's second axis is more relevant to Q-chef. The second axis distinguishes diversive from specific curiosity: the former is the unguided search for any new information, while the latter is the search for knowledge that satisfies a particular understanding goal. Diversive curiosity has been the subject of prior computational models of curiosity (Saunders and Gero 2001; Merrick and Maher 2009; Schmidhuber 2010) producing intelligent systems with the drive to explore beyond what they know. A related AI concept, novelty search, favours new solutions regardless of their effectiveness, and has been shown to outperform traditional goal-driven search in some contexts (Lehman and Stanley 2008). The Q-chef model incorporates diversive curiosity through its model of what an individual user will find surprising.

The computational modelling of specific curiosity has received significantly less attention. The question of how to model the triggers of specific curiosity is an area of active research. In our prior work (Grace et al. 2015; Grace and Maher 2015) we have taken inspiration from cognitive models of design that suggest specific curiosity is a key component of problem-framing (Schön 1983). Studies of human designers find that *unexpected discoveries* lead to reflective reinterpretation of the current problem, which in turn leads to further unexpected discoveries (Suwa et al. 1999). This reflective behaviour suggests that surprise is one possible trigger for specific curiosity, and that it can lead users to reformulate their goals and their approach to a problem. The Q-chef system models *surprise-triggered specific* *curiosity* in order to stimulate it in its users. When Q-chef generates a recipe that its user model indicates will be both highly surprising and highly valuable (i.e. preferred according to their tastes), it becomes a candidate for specific curiosity. The generation of such a creative recipe causes Q-chef to formulate a goal for further exploration of similarly surprising and valuable recipes, focusing both the system's and the user's attention.

Recipe Generation as a Domain for Computational Design

Food and nutrition are extraordinarily complex, as evidenced by the fact that their impact on health is still an area of active research. This rich and complex design space makes recipe generation a compelling domain for computational design research, especially when combined with the increasing availability of large-scale recipe data and food's nature as a universal everyday experience. Recipe generation has been the focus of AI research in the past (Varshney et al. 2013; Hammond 1986; Pagnutti and Whitehead 2015; Morris et al. 2012), but on few occasions has it been explicitly framed as computational design.

Not only is recipe generation a highly suitable domain for computational design, there is also a pressing need for technological interventions in dietary behaviour: obesity and associated health conditions account for a huge and growing fraction of medical expenditure (Cawley and Meyerhoefer 2012). Dietary diversity has long been recognised as a strong predictor of overall nutritional health (Vadiveloo et al. 2015; Drescher et al. 2007; Foote et al. 2004), even while the finer points of nutrition's impact on health remain an active area of debate within both academic research and the broader community. Q-chef is a computational design system driven by the need for dietary diversification: can we design recipes that will lead to an increase in health-promoting behaviour over time?

Changing food-related behaviour through technology is challenging, requiring education, motivation, and regular supervision (Barker and Swift 2009). Nutritional knowledge alone is insufficient (Worsley 2002) for changing food behaviour, as much of it is poorly accessible, conflicting, or too abstract for users to incorporate into their culinary decision-making. The PQE approach offloads some of the knowledge required for recipe generation onto the design system rather than the user. Computational design systems for health-related behaviour change also face the challenge of sustaining user motivation (Renner and Schwarzer 2005). Our curiosity-driven approach is inspired by this challenge, as curiosity is known to provoke intrinsic motivation.

A major challenge to any attempt to diversify diets is taste preference. The many reasons for widespread narrow taste preference include habit, culture, economics, and lack of knowledge. Studies of taste preference have identified a complex interaction between genetic, physiological, and metabolic variables (Drewnowski 1997), yet a significant portion of the effect can be attributed to *simple unfamiliarity* (Liem and De Graaf 2004). Put simply, unfamiliar foods are unpalatable, and the strong sensory displeasure they evoke represses the desire to eat more healthily. Dietary diversification is, however, a self-reinforcing practice: the greatest predictor of whether someone will like a new food is whether they have been exposed to other new foods (Steyn et al. 2006; Nicklaus 2009). These psychological traits make a curiosity-driven design system a natural fit for the recipe generation domain. Q-chef models a user's curiosity in order to generate recipes that stimulate their curiosity, and thereby drive the diversification of their diet over time.

The most well-known computational culinary design system is IBM Watson's "culinary cooking" project (Varshney et al. 2013). The Watson team tackled recipe generation from the perspective of computational creativity, and can generate recipes that are both fit to a user's request as well as surprising. Computational creativity researchers have also investigated the generation of slow-cooker recipes (Morris et al. 2012) and cocktails (Pagnutti and Whitehead 2015), both cases where the flavours are complex but the steps of each recipe are simple.

An early successful approach to computational recipe generation was case-based planning, most notably the CHEF system (Hammond 1986). CHEF generates new recipes by retrieving existing recipes and modifying them to contain the desired ingredients. The planning system can identify potential problems with a recipe and correct them using solutions it has applied in the past. For Q-chef we are initially treating recipes as sets of ingredients, rather than full plans with ordered steps.

Recommender systems approaches have also been applied to the recipe domain. Recommenders model user preferences in order to select one or more suitable recipes from a database. Culinary recommender systems have been used to suggest personalised recipes, diets, meals or ingredients (Freyne and Berkovsky 2010; Geleijnse et al. 2011). The Q-chef approach combines the user modelling components of these systems with a computational recipe design system.

PQE: A Personalised Model of Specific Curiosity

At the core of the Q-chef system is PQE, a model of curiosity that is personalised to a single user and capable of both specific and diversive curiosity. The Q-Chef system implements PQE in the domain of recipes, but we believe the PQE to be more generally applicable: hypothetically it could be applied to any design domain where designs that are p-creative (for the user) are desirable. The PQE has three components: a model of the user's preferences, the model of curiosity based on user behaviour and preferences, and a model that synthesises designs for the user. The interaction between these components and the user is shown in Fig. 1.

The PQE cycle iteratively *simulates* a user's curiosity in order to *stimulate* it. The user's feedback on designs refines the system's model of their preferences and familiarities, which are used to update a model of what will make them curious. The two models are then used as input to a computational design system, for which they provide the evaluation function by which designs that are both preferred and curiosity-stimulating can be generated. The output of that design system is

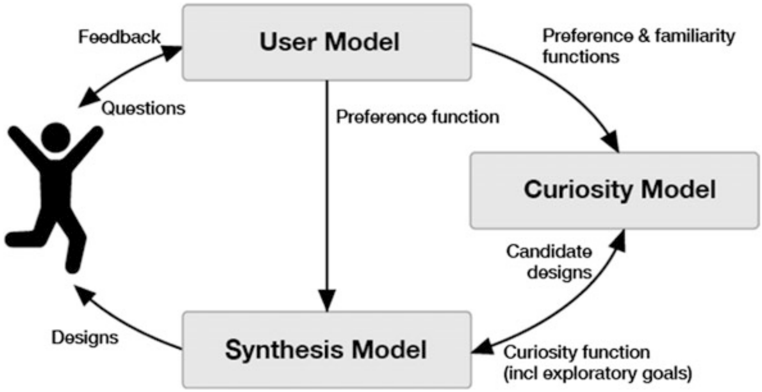


Fig. 1 To broaden its user's preferences, PQE learns what its user knows and likes, simulates their curiosity, then synthesises designs that they find creative

presented to the user, who can then refine the design, enact it, and then give feedback on it to the system, starting the cycle anew. The system is also capable of modelling the user's specific curiosity, by focussing its attention on particularly novel and preferred designs such that future generated designs share those traits. This is a form of metareasoning (Cox and Raja 2011) common to designers: (re)formulation as a result of surprise (Suwa et al. 1999). Curiosity-triggered goals limit what the system finds surprising, influencing the direction of synthesis.

The User Model incorporates feedback from the user about both their familiarity with designs as well as their preference for those designs—note that the two must be recorded separately in order to accurately model curiosity, as someone can be highly familiar with things that they greatly dislike, or vice versa. Familiarity ratings provide the basis for personalising the curiosity model, in the same way that preference rating provide the basis for personalising the preference function. Familiarity is the system's estimate of which recipes the user has seen—the subset on which curiosity evaluation will be based. Preference ratings are used for determining the value of a design and familiarity ratings are used to determine the novelty of the design. The User Model outputs a familiarity function and a preference function, each of which map from a recipe to a scalar value.

The Curiosity Model uses the familiarity function to generate a model of user expectations and create exploratory goals. Diversive curiosity is based on the Wundt curve model of (Saunders and Gero 2001), in which there is a peak level of novelty above which the stimulus becomes progressively more undesirable. PQE's curiosity model builds on prior approaches to modelling diversive curiosity (Saunders and Gero 2001) and on related systems that model user knowledge in order to recommend things that lie just beyond it (Adamopoulos and Tuzhilin 2014). The Curiosity Model comprises a probabilistic deep neural network that captures the user's expected likelihood for different ingredient combinations, coupled with algorithms for determining surprise and curiosity from those likelihoods. We model designs as being composed of a set of features. The Curiosity Model evaluates designs for *surprising combinations,* which consist of a *surprising feature* that is of low expected likelihood given the simultaneous presence in the design of a *surprise context*, a set of other features. Surprise can then be measured as the ratio of the rarity of the surprising feature in context to its rarity overall (Itti and Baldi 2005).

The Synthesis Model then uses the preference and curiosity functions, along with the exploratory goals, to generate recipe suggestions that are presented to the user. The Synthesis Model currently uses a simple sampling algorithm that leverages the same neural network as the Curiosity Model, but can be any design synthesis process. This contrasts with the majority of user-preference modelling systems, which recommend existing items drawn from a database. PQE applies technologies from recommender systems to personalise computational design, allowing for a much greater space of possible curiosity-stimulating designs.

The objective of synthesis is to produce designs that are in the user's *novelty sweet spot*, (i.e. is close to the user's preferred level of novelty, a parameter we will establish by experimentation). We use δ to refer to the user's preferred level of novelty, and it will be initially hand-tuned, but we hope to eventually incorporate it into the user model. When the Reformulation process identifies a design that is highly preferable and with novelty $\cong \delta$, it triggers specific curiosity. This can then be presented to the user as a suggested design, as well as lead to the generation of a new specific curiosity goal. This goal then weights the novelty term in the objective function towards similar surprises. Given set of design features $d = \{f, f_2 \ldots f_n\}$, the objective function is:

$$\frac{\sum_{k=1}^{n} pref(f_k)}{n} - abs\left(\max_{k \in \{1,2,\ldots,n\}} (surprise(f_k|d)) - \delta\right) \tag{1}$$

where *pref(f)* is a function from a feature *f* to a user preference rating, and *surprise (f|d)* is the curiosity-weighted surprise for a feature *f* in the context of the design *d.* The first term is the average of how preferred each feature in the design is to the user, representing the "value" part of the novelty/value definition of creativity. The second term is the distance between the most surprising feature in the design and the user's preferred level of novelty δ. This rewards designs for which the most surprising combination of features is equal to the user's level of surprise preference providing the "novelty" part of the definition of creativity.

PQE forms new design goals based on what it hypothesises might trigger specific curiosity in its user. This leads it to synthesise designs that are both personally creative and contextually interesting, as it uses its simulation of user curiosity to stimulate the user's contextual interest.

Q-Chef: A Personalised Curious Design System to Encourage Dietary Diversity

Q-chef is a prototype of the PQE framework in the domain of recipe generation, intended as a proof of concept. Q-chef internally represents each recipe as a list of ingredients, which we call an *ingredient set representation*. This is an incomplete abstraction to aid the synthesis process which leaves out amounts, cooking times and the steps to perform the recipe. We intend to post-process the ingredient set representations into full recipes using techniques from case-based reasoning, but the current prototype simply outputs ingredient sets. Q-chef utilises two datasets: one of users, and one of recipes. The Users database stores the preferences of all users. The Recipes database is stored in both ingredient set and full text formats.

The Q-chef User Model learns a representation of familiarity with recipes. In the current prototype our simulated users have no taste preferences, as we are focussing on simulating their familiarity for the purposes of validating our curiosity model.

Familiarity is simulated by partitioning the recipe database into recipes that users have or have not seen. This is then used as input into the Curiosity Model, which models user expectations and creates goals to explore surprising recipes. These curiosity-induced goals drive Q-chef's Synthesis Model to explore combinations that surprise it, and will surprise the user if its model of their curiosity is accurate. These goals are represented as temporary transformations of the surprise function, causing Q-chef to disregard surprises that are too dissimilar to the one that triggered the curiosity. The Synthesis Model generates recipes that fit the user, incorporating any specific curiosity goals into its evaluation of potential designs.

The Q-Chef User Model

In PQE, the role of the user model is to provide the Synthesis Model with what the user likes, and the Curiosity Model with what the user knows. In the current prototype user preferences are based on simulated profiles, rather than learned from real user ratings. We are currently focussing on demonstrating the feasibility of generation with specific curiosity, and thus are primarily concerned with modelling familiarity rather than preference. Through knowledge of how unusual the user finds recipes to which they have been exposed we can infer what recipes (and therefore ingredient combinations) they are likely to find surprising. We believe that this cognitively grounded model of curiosity will more greatly motivate users to adopt a broader diet.

The Q-Chef Curiosity Model

Q-chef simulates an individual user's specific and diversive curiosity in order to generate recipes that will stimulate that user. Diversive curiosity is implemented via a machine learning algorithm which models the user's expectations and thereby the flavour combinations they will find surprising, based on user familiarity data inferred by the User Model. Specific curiosity is an extension of the diversive model based on goal reformulation. When a recipe is both highly surprising according to the diversive curiosity model and highly valued according to the preference model, it triggers the in-depth exploration of similar recipes through specific curiosity. This is implemented as a transformation of the surprise evaluation function used for diversive curiosity. This model provides a cognitively grounded approach to diversifying diet, which has been shown to correlate strongly with health (Steyn et al. 2006).

The Q-chef User Model provides the familiarity function that is used as input to the Expectation process that outputs the curiosity function. The curiosity function evaluates the degree to which a recipe violates the user's expectations, and is used as input into synthesis. The base diversive curiosity function can then be perturbed by specific curiosity. Specific curiosity is simulated by the Goal Reformulation process, which focuses recipe synthesis on discoveries that are simultaneously surprising and preferred by the user. Q-chef's goal is to stimulate this process in its user, and drive them to explore a new ingredient or combinations thereof over the course of several recipes. The interactions between the synthesis and reformulation processes can be seen in Fig. 2. The Curiosity process implements diversive curiosity, while the Reformulation process implements specific curiosity. The Ingredient Set Synthesis, Recipe Requests and Post-Processing processes comprise the Synthesis Model.

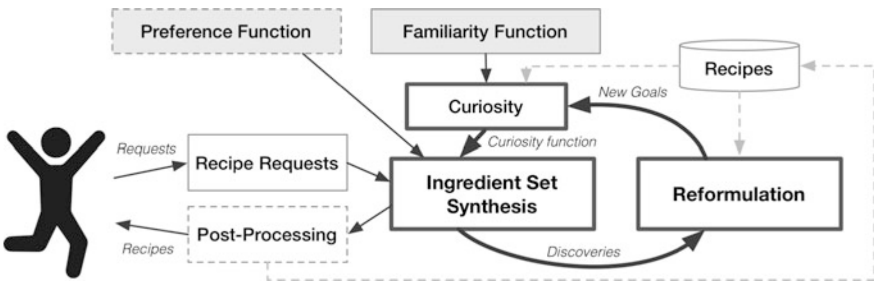


Fig. 2 Q-chef's curiosity and synthesis processes, with the preference and familiarity functions from the user model shown as input. The user curiosity function guides synthesis toward surprising (i.e. diversely curious) recipes, while the preference function guides synthesis toward preferred recipes. The reformulation process creates new goals that implement specific curiosity—these goals transform what the system finds surprising. The *dashed lines* around post-processing and the preference function indicate that those components are not yet implemented

The expectation process works with ingredient set representations of recipes, which form the design features described in the PQE model. Ingredients are only a partial model of the experience of eating, lacking smell, texture, temperature, plating, emotion, and many other contextual factors. In our current implementation this model is a proxy for defining a specific user's familiar and preferred foods. For example, a recipe for chocolate bacon cupcakes might involve sugar, butter, flour, bacon, cocoa powder, salt, eggs and baking powder. A user unfamiliar with bacon in sweet baked goods would rate *bacon* highly unlikely in the context of *sugar, butter and cocoa*. This recipe was amongst the most surprising using a model trained on 100,000 recipes (Grace and Maher 2016). This highly unhealthy example highlights Q-chef's motivation-driven long-term strategy: if a user finds this recipe both tasty and highly surprising, Q-chef will encourage them to explore this combination of sweet and salty flavours in contexts such as salads and stir fries. Exposure to new foods is the best known strategy for increasing dietary diversity in those with highly restricted tastes (Pliner et al. 1993), and we believe that by encouraging curiosity instead of prescribing health goals we can change nutrition behaviour.

Expectation modelling operates as described in (Grace et al. 2015) but is trained not on the entire database, but on the database filtered through the familiarity function from the User Model. The result of this restriction is that the curiosity model's expectations are derived only from the recipes that the user is believed to be aware of. This means that some combinations may be surprising to the user even though they are commonplace in the database. We implement this conditional model of expectation with deep unsupervised neural network, such as a Variational Autoencoder (VAE, Kingma and Welling 2013). In our experiments, surprise contexts are limited to a maximum size of three—it is expensive to evaluate significantly larger contexts due to the combinatorial explosion.

When Q-chef encounters an ingredient set containing a combination of ingredients that is both surprising and preferred, it creates a new goal to explore the underlying flavour combinations in future recipes. Goals are how Q-chef simulates its user's specific curiosity, and they temporarily shape what the system finds surprising to exclude anything unrelated to the surprising combination that created them. A series of similarly surprising ingredient sets will be synthesised due to the influence of the goal. This is a kind of reasoning about goals (Aha 2015) reported in our previous papers (Grace et al. 2015; Grace and Maher 2015). The Reformulation process (see Fig. 2) assesses such discoveries and decides whether they should engender new goals. This builds on diversive curiosity by allowing highly surprising ingredient sets to situationally affect what the system finds creative. We propose that this simulation of the user's curiosity will allow Q-chef to stimulate curiosity in its users by further exploring novel flavour combinations if they are verified as interesting to the user.

Specific curiosity reasons about new goals to focus ingredient set synthesis on a surprising combination. The Q-chef prototype implements specific curiosity by applying a weighting to ingredients based on their distance from the combination or ingredients that trigger surprise. This weighting is currently based on whether the new surprise (the one being evaluated for relevance to the specific curiosity goal) shares either the *surprising feature* or *surprise context* of the triggering event. We are investigating a variety of similarity functions for comparing surprising events, each of which would represent a different model of specific curiosity behaviour. This weighting is then applied to the curiosity function to influence synthesis.

The Q-Chef Synthesis Model

The role of the Synthesis Model in PQE is to synthesise designs that are both novel and valuable to the user. To synthesise novel recipes, Q-chef must search the powerset of all possible ingredient sets. To search such a large space a heuristic technique is required, as exhaustive search is unlikely to produce useful designs within feasible time constraints. The search objective is determined by the requirement for creativity to the user (i.e. a combination of user preference and surprise, modified by curiosity-induced goals). The Q-chef Synthesis Model comprises the processes of Ingredient Set Synthesis, Recipe Requests and Post-Processing in Fig. 2, where they are depicted alongside the Curiosity Model processes due to their tightly coupled nature.

The Reformulation and Synthesis processes of Q-chef iteratively interact to frame and solve (respectively) the design problem: Reformulation creates new goals based on the progress of Synthesis, and those goals affect Synthesis. Synthesis generates ingredient sets that are preferred by and novel (i.e. unexpected) to the user, using the PQE objective function defined in Eq. 1. The proposed objective function will maximise the mean user *preference* of all ingredients in the candidate ingredient set. In the current prototype this ingredient set representation is returned to the user, while in future it will be post-processed into a full recipe. This is inspired by models of design cognition (Schön 1983) and by our own previous work (Grace et al. 2015; Grace and Maher 2015; Poon and Maher 1997). Synthesis produces a set of candidates that are then investigated by Reformulation for any highly surprising discoveries that create new goals. Search proceeds with the (potentially) modified objective function. This iteration continues until an ingredient set that exceeds an objective threshold is found, or until the maximum time for searching is reached.

Preliminary Experiments in Simulating Curiosity

We have implemented the model of specific curiosity at the core of Q-Chef (Grace and Maher 2016), along with a functioning prototype of the Synthesis Model that allows us to synthesise recipes. The current prototype generates recipes by sampling from the deep neural network used in the Curiosity Model, which encodes the joint probabilities over all ingredients. We simulate users by manually partitioning our recipe database, creating a subset of recipes that each artificial user is familiar with. Here we briefly describe our implementation (which is detailed in depth in Grace and Maher 2016), and then present a set of simulations showing the effect of different expectations (arising from different familiarity with recipes) on our curiosity model.

Preliminary Q-Chef Implementation

We have gathered a database of $\sim$100,000 recipes from web sources, and removing those ingredients that appeared in less than 0.1% of recipes as they have insufficient training examples to build expectations about. We then trained a VAE (Kingma and Welling 2013) to serve as the model of expectations, with each recipe being represented as a binary vector of all possible ingredients (119 in our current experiments). Hyperparameters were tuned using Spearmint (Snoek et al. 2012), with the resulting network having 1000 neurons in each of two encoder and two decoder layers, and 100 neurons in the hidden layer. We have implemented our model of surprise using the "wows" method described in Baldi and Itti (2010) that calculates the likelihood of a *surprise feature* (an ingredient) given a *surprise context* (a set of other ingredients). One "wow" of surprise indicates that one extra bit of information is provided by the presence of that feature in that context (i.e. it is half as likely to occur).

We have implemented a sampling-based method to generate designs from the probability distribution over ingredient combinations captured by the neural network used in expectation. As a stochastic generative deep network, the VAE learns a hidden vector of random variables that, when sampled, can be transformed into vectors of design features (i.e. ingredients). These sampled designs are drawn from the same distribution used by the model of curiosity, and can therefore be considered "plausible" designs. We do not claim that this random sampling is a plausible model for design reasoning, only that it tends to produce designs that are considered plausible by the system, and thus can serve as an initial generative model.

The current implementation of specific curiosity generates a new design goal to explore designs similar to the triggering surprise. This is presently triggered based on a threshold for novelty and value. Generated recipes above this threshold result in Q-chef transforming its goal and generating a set of recipes with similar surprising ingredients to the trigger.

Simulating Personalised Curiosity with the Q-Chef Implementation

We separated the dataset of $\sim$100 k recipes into two mutually exclusive sets, as an extreme example of two users with highly different familiarity within the design space of recipes. The split was made based on the presence of sugar (brown, plain, or icing/confectioner's) in the recipe, yielding $\sim$60,000 recipes without sugar (test user #1) and $\sim$40,000 recipes with sugar (test user #2). We compare these two users with a control: a hypothetical omniscient user who has knowledge of all recipes in the database (test user #3). We produce such exaggerated simulations of user profiles as a proof-of-concept of the Q-chef model's ability to personalise. Specifically, that it is able to produce different expectations, construct different exploratory goals, and synthesise different recipes based on the user model.

We compared the most surprising recipes for to each user, noting the level of surprise elicited in all conditions in each case. The results can be seen in Tables 1 and 2. All values in wows are approximate, as the sampling method yields some variability. Listed values are the mean of ten runs. Note that wows are on a log scale, with 10 wows indicating that an ingredient was 2^{10} times less likely in its context than normally. Scores of <6 wows are considered low, 6–10 moderate, and >10 high.

The most surprising recipes in the "no sugar" condition involve combinations of ingredients that would be alien to someone who had never encountered sweet baked recipes. *Beef in a barrel* combined mushrooms, beef and garlic with pineapple, butter and salt, making it highly surprising for the "no sugar" and "all recipes" users, but unsurprising for the "only sugar" user, as they lack the experience with those ingredients to make a confident expectation (see Grace et al. 2014a for a discussion of the difference between novelty and surprise based on confidence). The bread pudding recipe featured bread in the context of raisins, eggs and pecans, which was highly surprising for the "no sugar" condition but mundane for the two "users" with experience in sweet baked goods. The *Kreatopita* recipe was one of the few to be surprising for all three conditions: it featured garlic, mushrooms and cumin in the context of breadcrumbs, eggs and cinnamon. Those three ingredients in the context are common both in sweetened and unsweetened recipes, so all three users formed confident expectations.

The most surprising recipes in the "only sugar" condition (#2) used ingredients rarely found in sweet baked goods in the context of ingredients that are common there. The Cajun chicken and sausage spaghetti recipe both featured capsicum, Worcestershire sauce and garlic (rare in sweet recipes) in the context of mint and flour (common in sweet recipes). Both recipes are from the same source—a Cajun barbeque cookbook—and involved similar roux bases and herb/spice profiles. None of the many other recipes involving mint and flour are savoury, leading to maximal surprise by user #2 and only moderate surprise (for different reasons) by the other

Table 1 The top three most surprising recipes for the "no sugar" user (#1)

Recipe name	Description	#1	#2	#3
Beef in a barrel	Minced beef served in a pineapple	**~13**	$\sim$4	$\sim$12
Whiskey bread pudding	Bread pudding with whiskey sauce	**~11**	$\sim$3	$\sim$3
Kreatopita	Greek pastry-topped casserole	**~10**	$\sim$10	$\sim$10

Surprise is expressed in "wows" (Ahn et al. 2011)

Table 2 The top three most surprising recipes for the "only sugar" user (#2)

Recipe name	Description	#1	#2	#3
Chicken sauce piquant	Cajun roast chicken	~7	**~15**	~5
Italian sausage spaghetti	Pasta with sausage chunks	~4	**~15**	~9
Bacon and onion muffins	Savoury muffins	~4	**~11**	~7

Surprise is expressed in "wows" (Ahn et al. 2011)

users. The bacon and onion muffins combined their eponymous savoury ingredients with those from baked goods: flour, baking soda, and milk.

We then took the surprising-to-all kreatopita recipe and used it as a specific curiosity trigger to generate recipes in each of the three contexts. Recipes synthesised by the system under these conditions must contain the same surprising feature and/or at least part of the same surprise context as the triggered recipe. Table 3 shows the most surprising three recipes from each condition, along with a (manual) interpretation of each.

The three generated recipes demonstrate the influence of their different familiarities within the recipe domain. User #3, with full knowledge of the recipe domain, is able to generate a recipe containing the same surprise context as the kreatopita but a different surprise feature (parsley). User #2 only possesses knowledge of sweet recipes, and creates surprise using the tension between breadcrumbs and vanilla. User #1 is able to use part of the triggering surprise (cinnamon, eggs and garlic) in combination with a new surprising ingredient (cream) and a number of spices. Each of these recipes is plausible according to Q-chef's estimate of the user's expectations, but simultaneously surprising in that they contain incongruous ingredients. The recipes generated by users #1 and #2 reflect the limitations of their knowledge: #1 found a savoury recipe that included eggs and cinnamon, and #2 found a sweet recipe that included eggs, cinnamon and breadcrumbs. This shows how PQE-generated designs will reflect their user's tastes while simultaneously stimulating their curiosity.

Table 3 The three most surprising recipes generated using Kreatopita as a specific curiosity trigger

User	Ingredients	Interpretation
#1	*Cream*, chillies, capsicum, *eggs*, *cinnamon*, **garlic**, coriander, oregano, black pepper, salt	Spicy whipped cream dip
#2	*Breadcrumbs*, eggs, milk, butter, brown sugar, cinnamon, **vanilla**	Breadcrumb and butter pudding
#3	Pork, *eggs, breadcrumbs*, chillies, cumin, brown sugar, *cinnamon*, **parsley**, salt	Deep fried spicy pork chop

Surprising features are bolded, surprise contexts are italicised

Future Development of Q-Chef

The future development of the Q-chef system has three current goals: extend the user modelling, develop complete recipes, and perform validation with real users. Creating models of real user preferences will require incorporating techniques from recommender systems, such as collaborative and content-based filtering. Q-chef will use these techniques to model user preferences for ingredients and tags, which will replace the current simulated model of familiarity. Recipe tags will help identify preference features not easily expressed at the ingredient level, such as cuisine tags that identify that a particular recipe is inspired by a particular culture's food: "Italian", "Japanese", etc.

Freyne and Berkovsky (2010) found that inferring ratings of individual ingredients provided more accurate recommendations than ratings of recipes. We intend to adopt a similar approach, transferring recipe ratings to their constituent ingredients (initially by assuming equal importance for all ingredients, although that is a simplification we hope to improve on). Recipe preference will be based on the average of ingredient ratings, which is an assumption that allows us to model preference in the relatively smaller space of ingredients ($\sim 10^2$) rather than recipes ($\sim 10^5$).

We plan to extend our model of recipes to include the concept of flavour. Ahn et al. (2011) created flavour networks, in which hundreds of common food ingredients were analysed based on their flavour-active compounds and composed into a graph of the relationships between ingredients based on the compounds they share. The flavour networks approach has huge potential for Q-chef, both as a way to calculate similarity between ingredients when determining user preferences, and as knowledge that can be used during synthesis.

In order for Q-Chef to generate complete recipes from ingredient sets we plan to apply case-based reasoning. The synthesised set of ingredients will be used as an index to a case-base of complete recipes. A new complete recipe is generated by adapting similar complete recipes that include cooking techniques. This process will use a separate knowledge base of recipes that include a representation of the ingredients and the cooking instructions (not depicted in Fig. 2).

Conclusions

The PQE framework describes a novel method for modelling the specific and diversive curiosity of an individual user as part of a computational design system that synthesises personally creative artefacts for that user. It builds on our previous work in computational curiosity and surprise to incorporate a model of user preference that guides it towards designs that would be considered creative by that user, instead of by the standards of a domain expert.

The Q-chef system describes an application of PQE to the computational design of recipes for the purposes of encouraging dietary diversity over time. Our prototype of the curiosity modelling components of the Q-chef system demonstrate that PQE finds different designs surprising given familiarity with different subsets of the design domain, which serves as a proof-of-concept for personalised curious design systems. Our prototype of specific curiosity demonstrates that different domain expectations also affect how the search trajectory is influenced by reformulation after an unexpected discovery.

This research demonstrates that models of curiosity can be used to personalise the evaluation functions used in computational design. Personalised creative design systems can provide individuals with new designs that are both interesting and valuable to them, leading to diversified tastes over time. In the case of recipe generation such diversification has been shown to correlate with overall health, providing a compelling domain in which to pursue curiosity in computational design.

References

Adamopoulos P, Tuzhilin A (2014) On unexpectedness in recommender systems: or how to better expect the unexpected. ACM Trans Intell Syst Technol (TIST) 5(4):54

Aha DW (ed) (2015) Goal reasoning: papers from the ACS workshop. Technical report GT-IRIM-CR-2015-001. Georgia Institute of Technology

Ahn Y-Y, Ahnert SE, Bagrow JP, Barabási AL (2011) Flavor network and the principles of food pairing. Sci Rep 1:196

Baldi P, Itti L (2010) Of bits and wows: a Bayesian theory of surprise with applications to attention. Neural Netw 23(5):649–666

Barker M, Swift JA (2009) The application of psychological theory to nutrition behaviour change. Proc Nutr Soc 68(2):205–209

Berlyne DE (1966) Exploration and curiosity. Science 153:25–33

Boden MA (2003) The creative mind: myths and mechanisms. Routledge, New York

Brown DC (2012) Creativity, surprise & design: an introduction and investigation. In: The 2nd international conference on design creativity (ICDC2012), Glasgow, UK

Cawley J, Meyerhoefer C (2012) The medical care costs of obesity: an instrumental variables approach. J Health Econ 31(1):219–230

Cox MT, Raja A (2011) Metareasoning: thinking about thinking. MIT Press, Cambridge

Drescher LS, Thiele S, Mensink GBM (2007) A new index to measure healthy food diversity better reflects a healthy diet than traditional measures. J Nutr 137(3):647–651

Drewnowski A (1997) Taste preferences and food intake. Annu Rev Nutr 17:237–253

Foote JA, Murphy SP, Wilkens LR, Basiotis PP, Carlson A (2004) Dietary variety increases the probability of nutrient adequacy among adults. J Nutr 134(7):1779–1785

Freyne J, Berkovsky S (2010) Intelligent food planning: personalized recipe recommendation. In: Proceedings of the 15th international conference on intelligent user interfaces, pp 321–324

Geleijnse G, Nachtigall P, van Kaam P, Wijgergangs L (2011) A personalized recipe advice system to promote healthful choices. In: Proceedings of the 16th international conference on intelligent user interfaces, pp 437–438

Grace K, Maher ML (2015) Specific curiosity as a cause and consequence of transformational creativity. In: Proceedings of the sixth international conference on computational creativity, June 2015, p 260

Grace K, Maher ML (2016) Surprise-triggered reformulation of design goals. In: Proceedings of AAAI 2016 (to appear)

Grace K, Maher ML, Fisher D, Brady K (2014a) Modeling expectation for evaluating surprise in design creativity. In: Design computing and cognition '14, pp 189–206

Grace K, Maher ML, Fisher DH, Brady K (2014b) Data-intensive evaluation of design creativity using novelty, value and surprise. Int J Des Creat Innov 3(3–4):125–147

Grace K, Maher ML, Systems ACS, Hartmann C, Dohle S, Siegrist M (2015) Surprise and reformulation as meta-cognitive processes in creative design. In: Proceedings of the third annual conference on advances in cognitive systems

Hammond KJ (1986) CHEF: a model of case-based planning. In: AAAI, 1986, pp 267–271

Itti L, Baldi P (2005) A surprising theory of attention. In: IEEE workshop on applied imagery and pattern recognition, pp 45–54

Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114

Lehman J, Stanley KO (2008) Exploiting open-endedness to solve problems through the search for novelty. In: ALIFE, 2008, pp 329–336

Liem DG, De Graaf C (2004) Sweet and sour preferences in young children and adults: role of repeated exposure. Physiol Behav 83(3):421–429

Macedo L, Cardoso A, Reisenzein R, Lorini E, Castelfranchi C (2009) Artificial surprise. Handbook of research on synthetic emotions and sociable robotics: new applications in affective computing and artificial intelligence. IGI Global, Hershey, USA

Merrick K, Maher ML (2009) Motivated learning from interesting events: adaptive, multitask learning agents for complex environments. Adapt Behav 17(1):7–27

Morris RG, Burton SH, Bodily PM, Ventura D (2012) Soup over bean of pure joy: Culinary ruminations of an artificial chef. In: Proceedings of the 3rd international conference on computational creativity, pp 119–125

Newell A, Shaw JC, Simon HA (1959) The processes of creative thinking. Rand Corporation, Santa Monica

Nicklaus S (2009) Development of food variety in children. Appetite 52(1):253–255

Ortony A, Partridge D (1987) Surprisingness and expectation failure: what's the difference? In: Proceedings of the 10th international joint conference on artificial intelligence, vol 1, pp 106–108

Pagnutti J, Whitehead J (2015) Generative mixology: an engine for creating cocktails. In: Proceedings of the sixth international conference on computational creativity, p 212

Pliner P, Pelchat M, Grabski M (1993) Reduction of neophobia in humans by exposure to novel foods. Appetite 20(2):111–123

Poon J, Maher ML (1997) Co-evolution and emergence in design. Artif Intell Eng 11(3):319–327

Renner B, Schwarzer R (2005) The motivation to eat a healthy diet: how intenders and nonintenders differ in terms of risk perception, outcome expectancies, self-efficacy, and nutrition behavior. Pol Psychol Bull 36(1):7–15

Saunders R, Gero JS (2001) Artificial creativity: a synthetic approach to the study of creative behaviour. In: Computational and cognitive models of creative design V. University of Sydney, pp 113–139

Schmidhuber J (2010) A formal theory of creativity, fun, and intrinsic motivation (1990–2010). IEEE Trans Auton Ment Dev 2(3):230–247

Schön DA (1983) The reflective practitioner: how professionals think in action. Basic Books, New York

Snoek J, Larochelle H, Adams RP (2012) Practical Bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959

Steyn NP, Nel JH, Nantel G, Kennedy G, Labadarios D (2006) Food variety and dietary diversity scores in children: are they good indicators of dietary adequacy? Public Health Nutr 9(5):644–650

Suwa M, Gero JS, Purcell T, Design IV (1999) Unexpected discoveries and s-inventions of design requirements: a key to creative designs. In: Computational models of creative design, pp 297–320

Vadiveloo M, Dixon LB, Mijanovich T, Elbel B, Parekh N (2015) Dietary variety is inversely associated with body adiposity among US Adults using a novel food diversity index. J Nutr 145(3):555–563

Varshney LR, Pinel F, Varshney KR, Schörgendorfer A, Chee Y-M (2013) Cognition as a part of computational creativity. In: ICCI* CC, 2013, pp 36–43

Vygotsky L (1978) Interaction between learning and development. Read Dev Child 23:34–41

Worsley A (2002) Nutrition knowledge and food consumption: can nutrition knowledge change food behaviour? Asia Pac J Clin Nutr 11(3):S579–S585

Wundt WM (1874) Grundzüge de physiologischen Psychologie. W Engelman, Leipzig

Part IX
Design Knowledge

Traversing the Barriers to Using Big Data in Understating How High School Students Design

Robin S. Adams, Molly Goldstein, Şenay Purzer, Jie Chao, Charles Xie and Saeid Nourian

Abstract The context of this paper is a "large learner data" project that seeks to respond to existing challenges by introducing educational data mining and learning analytics into K-12 engineering design research. To probe deeply into student learning, we are developing and refining computational techniques to analyze large process analytics datasets generated through a CAD-based software, Energy3D, that logs design process data as students complete an assigned design challenge, such as a net-zero energy efficient building. We are combining these process analytics with demographic data and pre/post-tests of science and design knowledge. In this paper, we revisit three illustrative research cases to reflect on our experiences and lessons learned with navigating big data, generating useful data visualizations, and integrating process analytics with traditional performance assessment methods to relate design actions to knowledge and learning outcomes.

Aims

The context of this paper is a "large learner data" project that seeks to respond to existing challenges by introducing educational data mining and learning analytics (Bienkowski et al. 2012) into K-12 engineering design research. Through a 5-year collaboration, we are applying a data-intensive approach to study student design learning and performance. The project involves engaging secondary students with Energy3D (http://energy.concord.org/energy3d), a computer-aided design (CAD) software tool for designing energy efficient solutions for the built environment based on Earth science, physical science concepts, and engineering principles required by the Next Generation Science Standards (NGSS) ETS1 (Achieve 2013).

R.S. Adams (✉) · M. Goldstein · Ş. Purzer
Purdue University, West Lafayette, USA
e-mail: rsadams@purdue.edu

J. Chao · C. Xie · S. Nourian
Concord Consortium, Concord, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_33

To collect large learner data, Energy3D automatically logs design process data as students complete an assigned design challenge, such as a net-zero energy efficient building. This includes fine-grained information on student actions, experimentation results, electronic notes, and design artifacts. For a single student, process data over the duration of a project can sum up to 20 MB, ranging from 200 to 2000 construction and analysis actions. To probe deeply into student learning, we are developing and refining computational techniques to analyze these large process analytics datasets. These techniques are being used to reconstruct the entire learning trajectory for each individual student with high resolution, providing a holographic method for assessing student performance. We are combining these process analytics with demographic data and pre/post-tests of science and design knowledge.

To date, we have produced research findings that focus on investigating common patterns of student design behaviors (e.g., using scientific experimentation to make design choices, making trade-offs, idea fluency, and reflection), as well as how patterns of design behaviors are associated with science and design learning outcomes measured using traditional performance assessment methods. Throughout these experiences we have been traversing the challenges of relating design actions (as logged in Energy3D) and knowledge (as evidenced in Energy3D performance and measured via pre/post-tests), and how these relationships offer explanations of learning outcomes. In this paper, we revisit three illustrative research cases to reflect on our experiences and lessons learned with navigating large learner data, generating useful data visualizations, and integrating process analytics with traditional performance assessment methods. We feel that sharing our reflections is a critical contribution to a larger discussion on what it means to gather, analyze, interpret, and eventually use large learner data to guide improvements in how students design.

Significance

In the context of K-12 science education, engineering design is a complex cognitive process in which students learn and apply science and design concepts to solve open-ended problems to meet specified criteria. Our understanding of what K-12 students learn from engineering design is limited (Katehi et al. 2009). A 2008 literature review concluded that many K-12 engineering education projects lacked data collection and analysis to provide reliable evidence of learning (Svihla and Petrosino 2008). The Committee on Standards for K-12 Engineering Education found “very little research by cognitive scientists that could inform the development of standards for engineering education in K-12” (National Research Council 2010). Similarly, how K-12 students learn and apply science concepts in engineering design processes is a fundamental interest in the learning sciences. Through engineering design projects, students practice science as they gather and analyze data through experiment-based inquiry and apply this knowledge to conceive, compare, and optimize solutions. Although previous research suggests that engineering design is an effective pedagogical approach to promoting science learning (Apedoe and Schunn 2013; Hmelo et al. 2000; Kolodner 2002; Mehalik et al. 2008), there are also concerns about the so-called “design-science gap” (Vattam and Kolodner 2008) that fails science learning in design projects (Apedoe and Schunn 2013; Hmelo et al. 2000). Overall, there is considerable need for approaches that can accurately and efficiently assess student design performance and learning of both science and design inquiry in engineering design projects.

There is a rich history of techniques for understanding how people design; however, most of these have been implemented in postsecondary and professional contexts and many emphasize research, not assessment. A common approach is to capture “think aloud” data to conduct verbal protocol analyses of design processes or design cognition (Cross 2001). Often verbal data is translated into visualizations to explore design behavior patterns such as structure-function-behavior design cognition diagrams (Kan and Gero 2009), process timelines (Atman and Bursic 1998; Atman et al. 2009; Mentzer et al. 2015), and linkography diagrams (Goldschmidt 2014). Some also use observation and video-based analyses (Purzer 2011; Purzer and Fila 2013). Another approach involves using design documentation such as journals to analyze relationships between design processes and design performance (Costa and Sobek 2004) or conduct latent semantic analyses to characterize designer performance (Dong et al. 2004). Others use technology-based tools that support documenting and reflecting on design processes (Hynes and Danahy 2010; van Schaik et al. 2011). Some performance-based methods include using concept maps to assess student understanding of the engineering design process (Sims-Knight et al. 2004), asking students to explain the relative importance of various design activities (Adams and Fralick 2010), asking students to critique a design process timeline and identify process improvements (Hsu et al. 2014), and using design scenarios to assess problem formulation capabilities (Atman et al. 2008; Adams et al. 2010).

Translating these research-focused approaches for use as assessments in K-12 contexts is a significant challenge. While each approach has strengths, each requires time-consuming data collection, data management, and data analysis procedures, often involving extensive human labor. An additional challenge is that the complexity and open-endedness of a design task can make it difficult to discern design patterns or correlate patterns to performance. For example, a pattern that looks like “gaming the system” in an inquiry activity (Baker et al. 2008) may be a legitimate search in a vast problem space for meaningful alternatives in a design project. For design, performance is not based on “getting the right answer” because multiple solutions are possible; rather, performance needs to be a function of understanding students’ growth in knowledge and skills necessary for informed designing (Crismond and Adams 2012). Collectively, these issues can significantly limit scale-up and broader use of existing approaches in K-12 classrooms (Xie et al. 2014).

Opportunity: Large Learner Data and Technology-Based Assessments

Information technology-based assessments offer a cost-effective solution for scaling up educational research. Large amounts of relevant data, real time feedback, and scalable and personalized support can be achieved now with the use of these technologies (U.S. Department of Education 2010). Similar to the Energy3D project, researchers have used technology-based assessments to study inquiry within interactive media and games (Clarke-Midura et al. 2011; Sao Pedro et al. 2013; Horwitz 2011; McElhaney and Linn 2011). These approaches have rarely been exploited for assessing design, a process that includes inquiry but is fundamentally distinct in many ways (Lewis 2006).

While we anticipate many affordances of integrating technology-based assessment into research on how people design, we also expect this will come with its own set of challenges. Some of these challenges may be unique to open-ended tasks such as engineering design that might make it necessary to combine learning analytics with human-based qualitative analysis to be able to draw strong conclusions about student learning (Worsley and Blikstein 2014). As Socha et al. (2016) note, some challenges may be the complexity and scale of the data itself such as being able to navigate a complex dataset that combines multiple modes of data (e.g., activity logs, reflection notes, video playbacks) which traverse fine-grained to more macro-level units of analysis; some challenges may be the nature of the cross-disciplinary collaboration, which may be a requirement for these kinds of endeavors, which will likely involve negotiating among different perspectives (e.g., quantitative-qualitative, software programming-educational research dynamics).

In this paper we focus on sharing lessons learned from using large learner data to identify, develop, and test approaches for assessing design performance and learning through engineering design projects in secondary school. In the following sections, we describe Energy3D, which serves as simulated engineering design environment for this project. We then present three illustrative cases of research studies to critique and debrief on our experiences with using large learner data to understand how secondary students learn design.

Method: Energy3D as a Learning and Research Platform

Energy3D is a free, open-source software that allows students to create 3D buildings and simulate energy consumption (Xie et al. 2014). The software offers a simple 3D graphical user interface for drawing buildings, and evaluating their performance using cost and energy (solar and heat) simulations (see Fig. 1).

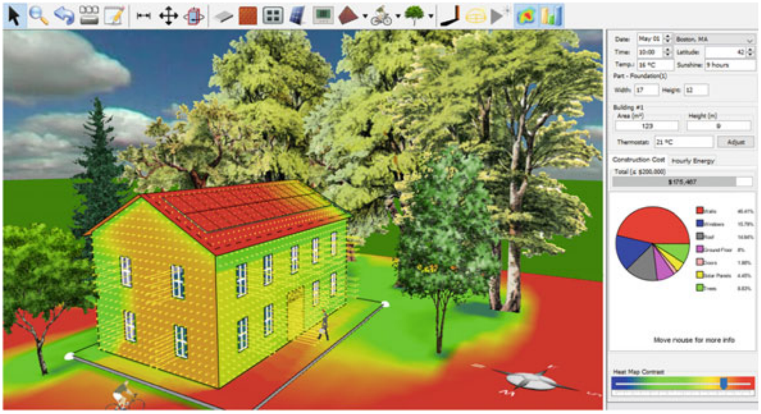


Fig. 1 Energy3D performance calculated (e.g., energy and cost)

As a learning platform, Energy3D provides computer-aided engineering tools for students to design, analyze, and construct green buildings that utilize renewable energy. For a given design challenge, students can quickly sketch up a realistic-looking building and then evaluate its energy performance for any given day and location (see Fig. 1). Energy3D can rapidly generate energy consumption simulations (i.e., time graphs, heat maps, and a solar simulator) based on computational physics to allow students to make informed design decisions. Students can use a notepad tool to describe and reflect on their designs and science simulations. At the end of the design, Energy3D allows students to print out a design, cut out the pieces, and use them to assemble a physical scale model.

As a research platform, Energy3D logs all design process data in a non-intrusive way as students complete an assigned design challenge. This includes fine-grained information on student actions, experimentation results, electronic notes, and design artifacts. This interaction data is translated into a JSON data stream for each student with a list of all interactions including: (1) the date/time the action was carried out, (2) the file in which the action was carried out, (3) the description of the action (e.g., add, edit, move, resize, notepad, etc.), and (4) the object towards which the action was directed. Energy3D also has the capability of reproducing the design process as a video display, similar to time-lapse photography, which integrates both activity log and notepad data.

These fine-grained CAD logs of large learner data possess all four characteristics of "big data" (IBM 2012): high volume, high velocity (data is collected in real time to support rapid feedback), high variety of data types (from learner actions to simulation data and experiment results), and high veracity (data is comprehensive and accurately documented to ensure fair and trustworthy assessments of student performance).

Results: Three Illustrative Cases

We are using Energy3D to investigate: (1) patterns and relationships in engineering design processes and how these are associated with prior knowledge, design performance, project duration, demographic factors, and learning outcomes, (2) the effect of engineering design process on science learning outcomes (e.g., to what extent does design iteration contribute to science learning of energy concepts), and (3) the effect of science inquiry processes on engineering design outcomes (e.g., scientific experimentation via Energy3D simulations and how these relate to design choices and revisions).

Over the past 2 years we have conducted multiple studies in which students use Energy3D to complete a design challenge. Here we focus on three studies, summarized in Fig. 2. Each study has been previously published, and as such we provide only limited details relevant for the purposes of this paper. Overall, each case speaks to different research approaches and lessons learned regarding: conducting multimodal analyses linking micro and macro grain data, integrating process data (internal to Energy3D) with pre/post-test data (external to Energy3D), pursuing targeted analyses using single data modes, and generating visualizations to support both human and computer-based pattern analyses.

As shown in Fig. 2, the three cases included high school students in the Midwest or the Northeast. The number of participants ranged from 44 to 109 students: Case 1

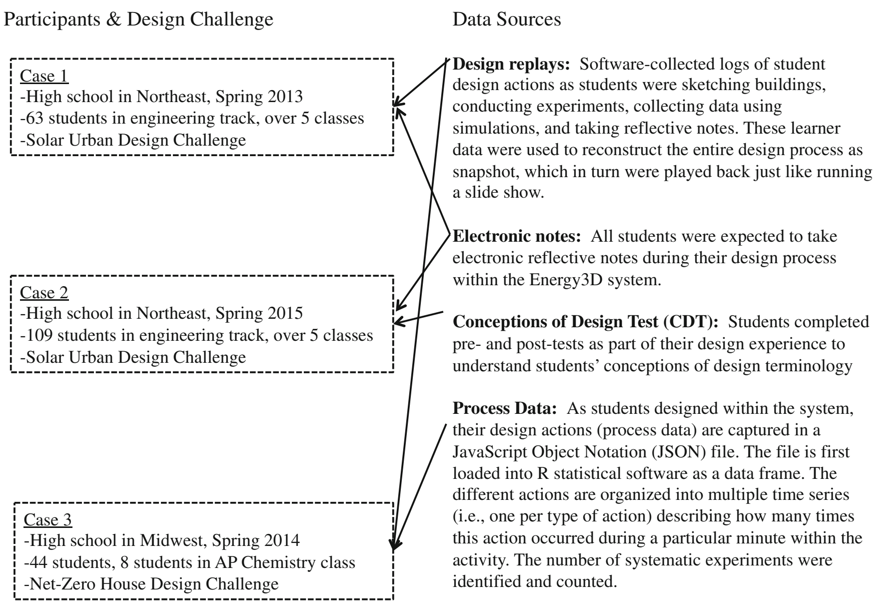


Fig. 2 Overview of three illustrative cases—context, participants, design challenge, and data sources

(n = 63), Case 2 (n = 109), and Case 3 (n = 44). While the number of participants may appear small, each Energy3D log file is on the order of thousands of design actions. Students in both contexts completed either one or both of two design challenges. One challenge, net-zero House, involved designing an energy-efficient single-dwelling home (Xie et al. 2014); the other, solar urban design, involved designing an energy-efficient urban block, in which each building could impact the energy-efficiency of adjacent buildings (Xie et al. 2014). For both scenarios, students were provided with a one-page handout that summarized design requirements and provided instructions on how to open, use, and save files on Energy3D. Teachers were provided with similar curricular resources. Each design challenge project was implemented during regularly scheduled class hours, and teachers were encouraged to link the project to other curricular goals including NGSS standards.

Large learner data was collected for each design challenge from every student. As shown in Fig. 2, this encompassed a variety of data sources and data types. Data sources included: (1) data automatically collected through Energy3D (i.e., process data, electronic notes), (2) data collected through Energy3D with additional post-processing (i.e., design replays, design performance), and (3) data external to Energy3D (i.e., conceptions of design pre/post-test). In the following sections, we summarize these three cases to illustrate data visualization outcomes and discuss lessons learned.

Case 1: Science Connections Through Tradeoffs

For this exploratory study, we were interested in understanding connections between science inquiry and design, in particular students' use of Energy3D science experiments in relation to informed design behaviors (Crismond and Adams 2012) such as balancing benefits and trade-offs (Purzer et al. 2015). A first step was to identify which automatically generated Energy3D data would allow directly investigating these relationships. Energy3D log files document student process data at a very fine or micro-level grain size—e.g., add, edit, move, and test. Most coding frameworks on design cognition or design processes articulate design activities at a larger grain size such as problem formulation or idea fluency, which are likely comprised of sequential combinations of finer grain actions. Given our research goals, we selected the design replays and electronic notes (see Fig. 2).

Design replays are generated from Energy3D activity log and allow a researcher to observe a student's unfolding design process as a sequence of design actions captured within the software. Like a video, the design playbacks can be replayed multiple times either as a whole, from beginning to end, or as smaller episodes to support focused analysis on a sequence of design actions. Researchers can also set the playback speed. For example, a design replay can be observed in real time (e.g., an hour-long project as an hour-long video) or at a pace such as 10-s increments (e.g., an hour long project as a 6 min video). The electronic notes are collected within Energy3D activity logs, and provide a place for students to write reflections,

Table 1 Action visualization coding protocol (Purzer et al. 2015)

Framework	Description	Source of evidence	'Symbol' in representation system
Design goal	The goal of a sequence of activities or the frame of actions	Student notes	⬭
Design moves	The activities students can do within the system (e.g., make, edit, delete, change shape, change orientation, simulation, etc.)	Design replays from log file	▽
Self-reflection	Reflection or thinking about the actions in relation to the goal or frame	Student notes—explanations that link goals to actions	◇
Science concept	Denotes existence of or reference to a science concept	Student notes—explanations	●

outcomes of experiments, ideas for what they could improve, and any other kinds of issue they want to document. Like all Energy3D data, the electronic notes are time stamped and logged in the system, and therefore integrated into the design replay function. In this way, researchers can observe a student's overall design process along with their evolving written comments. By combining design replays with the accompanying electronic notes we could "chunk" a student's design process into goal-directed design sequences, which could then be coded with existing empirically grounded frameworks.

A second challenge was determining useful coding frameworks that would easily map to the kinds of observable activities evident in the design replays and electronic notes. Valkenburg's coding scheme (Valkenburg 1998) provided a useful mechanism for characterizing sequences of design activities that link goals to moves (observed in the design replay) and reflections (observed in the electronic notes). For example, the electronic notes allowed us to see when a student was working towards a particular goal, such as optimizing a roof design, and the design replay allowed us to observe the associated design actions or "moves". Table 1 summarizes this coding scheme, which includes a code for denoting references to science concepts. It also illustrates how data sources were integrated to analyze patterns.

Data Visualizations Linking Log Files, Data Replays, and Reflection Notes

As shown in Table 1, an added benefit of Valkenburg's coding scheme (Valkenburg 1998) is the use of symbols that could be used to generate visualizations of interconnecting science and design inquiry processes. For example, each design episode began with a goal, represented with an oval. The triangles represent moves or student actions such as making, editing, or changing the shape of a roof. When a connection to a science concept is made, this can be indicated with a blue dot. Student reflections are signified with a diamond. By generating these visualizations we could quickly identify connections between science and design inquiry.

Full detailed analyses for two students through visualizations are available in previous work (Purzer et al. 2015). Design replays supported with detailed student reflective notes provided sufficiently detailed information for characterizing student design thinking and design behaviors. One of our research findings was how these visualizations showed a notable progression of student behaviors starting with idea generation and evolving to more sense-making behaviors, such as balancing benefits and tradeoffs.

The visualizations also made evident meaningful applications of science learning when students attempted to balance design benefits and trade-offs. In a particular design episode (Purzer et al. 2015), the student conducted experiments based on iterative revisions to the roof and wall of his building, while reflecting upon the size and direction of window placement and the resulting solar gains. By mapping design and science inquiry moves and reflections into a combined representation we can see evidence of important design behaviors such as systematic experimentation and decision-making with a trade-off analysis. By deconstructing the design process and offering a visualization for the interdisciplinary research team, we were able to better articulate behaviors that were leading to science connections in students.

Lessons Learned

Overall, this study highlights many of our ongoing experiences with large learner data. It illustrates issues with navigating multi-modal data—the design replay and electronic notes are different types of data, although the design replay function allows integrating these different data types and streams in ways that leverage the benefits of each data source. The design replays also supported iterative refinement of our analysis approach. This case also illustrates challenges with mapping activity-level units of analysis (i.e., captured automatically in the system as edit, move, resize, etc.) to process-level units of analysis (i.e., observed as design replays but translated into visualizations through coding). We are developing a sharper awareness of the non-trivial challenges of translating across different units of analyses, and how continuing down a pathway of technology-based assessments will require considerable work to build bridges between existing design inquiry frameworks and frameworks that can used for fine-grain analyses. This case also illustrates the value of generating visualizations as intermediate representations for identifying and characterizing patterns, even though these visualizations are manually created. These visualizations enabled our research team to collectively understand features of design performance and learning and investigate new kinds of visualizations that can support discovery-driven research.

Case 2: Connecting Reflection and Informed Design

This study investigated students' improvements in design thinking in association with level and breadth of design reflectivity (Goldstein et al. 2015a). Understanding students' design thinking, particularly at the K-12 level is challenging. To tap into design thinking, we used a Conceptions of Design Test (CDT) (Goldstein et al. 2015a) to assess student understanding of design through ranking and explaining the relative importance of a list of terms representing informed design (e.g., understand the problem, iteration, modeling). This performance-based test was given prior to starting the first Energy3D design challenge design activity and at the conclusion of the final design challenge. Assessing reflection presents an additional challenge. The electronic notes option in Energy3D provides one pathway for capturing students' reflections during their design process. For this study, all students were expected to write reflective notes while designing their net-zero houses and were prompted to "describe your design ideas and explain why you think they are good ideas." These electronic notes were examined and scored using a coding protocol based on existing literature that focuses on level and breadth/amount of reflectivity (Goldstein et al. 2015a).

Integrating System-Generated Data with Questionnaire Data

Using statistical analysis, we sought to understand if a relationship exists between student reflectivity and their understanding of informed design. As such, this study provides an example of integrating data generated from Energy3D (the electronic notes) with data generated external to Energy3D (pre/post Conception of Design Test). A paired *t* test was used to evaluate gains in informed design thinking and a one-way ANOVA was used to evaluate the relationship between student reflectivity and gains in informed design thinking.

The analysis showed gains in recognition of informed design. We found that highly and moderately reflective students had higher gains in informed design thinking compared to those with low reflectivity scores. However, the results did not indicate that students who demonstrated a higher level of reflectivity also became more informed designers. One possible explanation is that students in the study were beginning designers with limited experience. That we observed some gains in informed design thinking in relation to reflectivity suggests that Energy3D provides a learning experience that may help students develop awareness even though their reflexivity skills may lag behind. While reflection is an important component of designing and design learning, perhaps other behaviors are as essential. The path to informed design, it seems, cannot be predicted by reflection alone, indicating the need to better understand how other patterns of informed design interact.

Lessons Learned

This study used more traditional forms of assessment (i.e., pre/post-tests and students responses/reflections) as opposed to log data. This case provided an innovative way to think about assessing reflection in terms of both breadth and depth. However, as we move toward using larger datasets, this method of coding reflections may prove too difficult from a scale-up perspective. In comparison, analyzing the Conception of Design Test is quite straightforward and could be automated. Looking at the relationship between reflection depth and reflection breadth might allow a macro-level view of reflectivity in the future.

By providing students with a reflection prompt, we had a more consistent quality of reflections than when students are not given any guidance other than to "think and reflect like an engineer" (Goldstein et al. 2015a, b). Even with guidance, we have observed that the quality of reflections can vary significantly from student to student.

We anticipate many future opportunities as we expand on this study leveraging the large learner data. First, the Conceptions of Design Test provides one vantage point for eliciting what students consider to be important in designing (and why). In the future we are exploring ways to triangulate these types of data with the process activity log files to investigate relationships between what students express as important to design and what behaviors they employ while designing. Similar to the first case, this will require finding ways to link the fine-grained process data and perhaps sequences of these actions to performance outcomes from the Conceptions of Design Test as well as reflectivity level to more fully characterize student design performance.

Case 3: Connecting Design Replays and Process Data

Although idea fluency plays an important role in design, it can be hard to identify in student design activities because we may not have access to the full realm of design possibilities a student considers before focusing on a smaller subset of options. Similar to the first case we presented, we used the design replays for this exploratory study but now in combination with the process data (the micro-grained activity logs) to examine if idea fluency is observable from watching the student design behaviors. The research goal was to determine if and how learning analytics can confirm the presence or absence of idea fluency (Goldstein et al. 2015c).

As an exploratory study, we selected a subset of an existing large learner dataset (n = 44). We reviewed 3 hours of design activity time for a class of eight students, representing approximately 160 MB of Energy3D process data as design replays and the corresponding process data. A coding framework was iteratively developed for idea fluency as observed through watching Energy3D design replay files. The coding framework links levels of idea fluency to distinct design actions documented in the Energy3D log files such as building, modifying or adding walls, roofs, windows, solar panels, and trees.

Data Visualizations Linking System-Generated Process Timelines with Human Observations

Two researchers coded for idea fluency and were able to distinguish a very idea fluent student from a student who generated considerably fewer ideas. By combining the design replays and the coding protocol, we were able to determine that idea fluency is directly observable through Energy3D. The most idea fluent student was observed building and modifying the windows and solar panels in order to achieve better solar performance of the building. This was observed in the design replays as she changed the size, shape and position of windows in order to have a higher functioning home with lower energy usage requirements. She also explored many positions and quantities of solar panels. In contrast, while the least idea fluent student in the sample did modify windows and solar panels in his design, he did not explore a wide range of options. The coding protocol allowed researchers to discuss student range of ideas numerically, as students' overall idea fluency scores could range from 0 to 2.

Challenges with linking observed design behavior (idea fluency) and process data (build, add, or modify a design element) was further investigated using statistical analysis. When students use Energy3D, design actions (process data) are captured in a JavaScript object notation (JSON) log file. We analyzed this process data for each individual student in the study by loading the log file into R statistical software as a data frame. The different actions (e.g., build/modify windows) could then be organized into multiple time series diagrams (i.e., one per type of action) that show how many times an action occurred during a particular minute within the log file. Figure 3 represents this action count output for the (a) most and (b) least idea fluent students identified from the design replay analysis. The process data analysis confirms that these students are distinguishable by their process data, just as they were from the design replays.

Figure 3 allows a way to visualize a design process generated from the fine-grained log files for the most idea fluent student in the class for all possible

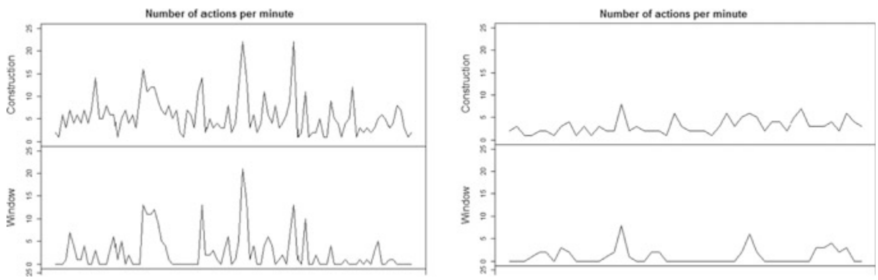


Fig. 3 Essentially switching the ideas to read left to right): (*Left*) Idea fluency as seen from the process data for the least fluent students (*Right*) Idea fluency as seen for the most fluent students

construction activities (i.e., build/modify: walls, roof, windows, solar panels and tree) and for specific window for building or modification actions. Not only does the process data correspond with observations from the design replays, the graphs also offer a useful visual tool to graphically assess the extent to which students practice idea fluency. As such, we anticipate this approach will be fruitful for analyzing other kinds of informed design behaviors.

Lessons Learned

In general, this case provides additional insights into using multi-modal data (i.e., design replays and process data) to understand design behavior, preliminary stages of quantifying design behavior (i.e., idea fluency), and generating data visualizations (i.e., graphs of process data counts) as intermediate representations for identifying and characterizing patterns. Coding the design replays for levels of idea fluency allowed us to quantify qualitative data, while visualizations of the process data allowed a qualitative perspective for the quantitative data. Through the case of two students, we demonstrated how micro-level process data could be used to validate macro-level observations made from viewing student design process through design replays. Together process data and design replays essentially tell the same story, and might be able to be used interchangeably. However, future work will need to investigate design behavior that might not be easily detected from observing the design process, and we will need to continue our growing understanding of ways to link fine-grained micro-level design data (distinct design actions) to macro-level design behaviors (informed designing) to combinations and sequences of macro-level design behaviors.

Summary and Future Work

Engineering design is a skill that is hard to measure, but it must be fairly assessed if it is to be taught in every K-12 classroom as required by the Next Generation Science Standards. The large learner data techniques we are developing through this project are likely to make an impact on the assessment of engineering design in K-12 contexts. In this paper, we offer three cases as pathways for thinking about the kinds of research that can be conducted with large learner data about how students design and for reflecting on lessons learned. By sharing these experiences we hope to contribute to crucial conversations on what it means to gather, analyze, interpret, and eventually use large learner data to guide improvements in how students design.

Challenges and Opportunities

As shown in these three cases, we are using Energy3D to investigate (1) patterns of engineering design processes and how these relate to design performance including science and design learning outcomes, and (2) relationships between science and design inquiry. While we are finding that large learner data provides many opportunities, there are also many challenges. In some cases, we have developed and tested approaches to resolve challenges; however, the essence of these challenges remains as areas for ongoing development. These are summarized below in terms of navigating the complexities of multi-modal data, translating among different units of analysis and inquiry lenses, and generating intermediate visualizations.

Navigating the Complexities of Multi-modal Data

The data set generated through Energy3D contains data of different types (design replays, activity logs, electronic notes, process analytics, etc.) and streams (some data is generated within the system, some requires post-processing). This creates a rich and complex data set with many opportunities to integrate and triangulate among different data sources. As an example, the design replays affords zooming in and out, fast forwarding, and rewinding to locate a phenomenon of importance that can then be investigated through other data sources. Also, some of the data can be automatically analyzed; some requires manual coding but could be automated in the future. However, navigating such a complex dataset to make informed research design decisions can be its own challenge. What stream or combined streams of data can best provide the most direct evidence for a given research goal? How to combine streams that have different scales or units of analysis?

Translating Among Different Units of Analysis and Inquiry Lenses

A central theme in our on-going research is finding ways to map fine-grained activities captured in the Energy3D logs (e.g., edit, move, add) to more coarse-grained design process activities (e.g., balancing trade-offs, reflection, idea fluency). This translation challenge has many elements. In part, it involves mapping across different units of analysis; in part, it involves mapping patterns that can emerge through data mining to patterns that have theoretical or practical value. As shown in these cases, certain kinds of data afford building bridges between different units of analysis and conceptual frameworks. Case 1 demonstrates how the electronic notes feature provided a bridge for connecting individual “moves” (activities in the log file) and reflections (comments documented in the electronic notes) to design process sequences (observed in design replays). Similarly, Case 3 demonstrated how process data could be used to validate observations of idea fluency in the design replays. Case 2 tells a different story of integrating process analytics with performance assessments. Other researchers have noted this difficulty in translating big data from students into actionable intelligence, and our research attempts to address these difficulties.

Generating Intermediate Visualization

Perhaps the critical importance of visualizations is no surprise; we know that visual representations can be powerful. The cases presented in this paper continue a history that illustrates the power of design process visualizations as both outcomes and intermediate tools for making meaning of design behaviors. Our research team has repeatedly experienced the many benefits of iteratively generating intermediate visualizations to aid pattern discovery and characterization, as well as collective sensemaking. This is providing an added push towards developing visual process analytic techniques, as illustrated in Case 3. In addition, we are learning how intermediate data visualizations are helping us traverse the challenges of multimodal data: visualizations generated from the micro level data offer a more macro level view more in line with existing research and frameworks for analyzing design behavior.

Using the Affordances of Human Analysis as a Pathway for Scaling Up Big Data Analysis

Many of our studies rely on some element of human labor to establish links between micrograin design process actions and macro level patterns of design process behaviors (made up of many combinations of micrograin actions). This is not feasible at the scale of big data, yet we are finding useful ways to integrate learning analytics with human based qualitative analyses that could be scalable. Case 3 illustrates how we combined labor-intensive human analysis with visualizations generated from Energy3D log files to test for observed variations of idea fluency. In other words, we used the affordances of human analysis to characterize macro-level observations that could then be tested with system generated micrograin design action representations. This appears to offer a pathway linking initial development of design patterns that meaningfully distinguish variations in design patterns (via human analysis) with log file generated patterns (via automated analysis).

Iterative and Integrative Co-development

Through these reflections we came to understand the iterative and integrative dimensions of large learner data research for the context of design cognition. Most lessons learned, from these three studies as well as the larger body of research connected with this project, are used as feedback to improve the Energy3D software. For example, we are developing ways to enable automatic assessment of solution quality, which has consequences for what features are logged in the system, how these can be easily extracted for analysis, and what kinds of quality assessment information might be provided to students as a formative feedback. As another example, our interest in the informed design behavior of balancing trade-offs (Case 1) has resulted in integrating cost calculations into the energy calculations so that students can more easily explore and grapple with trade-offs between cost and energy efficiency. Similarly, our interest in reflection (Cases 1 and 2) has changed the ways reflection notes appear on the monitor as a prompt for students to write electronic notes more frequently and with greater detail and intent.

Our experiences also support the insights of Worsley and Blikstein (2014) in how integrating learning analytics with human-based qualitative analysis may be necessary for situations that involve open-ended tasks such as engineering design. Each case in this paper illustrates integrating qualitative and quantitative perspectives as one mechanism for bridging the gap between different units of analyses. In some cases, qualitative approaches such as observing the design replays are used to inquire into elements in the log files; in others, quantitative approaches are used to inquire into qualitatively observed design patterns. Each case also illustrates integrating top-down and bottom-up approaches, which perhaps explains why these three cases are exploratory studies along two intersecting tracks: exploring features of a design phenomenon (e.g., reflection, science-design integration, idea fluency) and exploring what aspects of the system allow investigation or visualization of that phenomenon. The top-down approach starts with existing design cognition theories that are appropriate to the study, and then looks for ways to formulate these theories in forms computable from the Energy3D logs. Case 3 provides a useful example of this approach. The bottom-up approach starts with the Energy3D logs and attempts to reveal features in the data using design cognition and performances questions without necessarily being guided by a specific design theory. While the cases presented in this paper do not provide as much detail regarding this particular approach, our other research on visual process analytics provides examples of using data mining to visualize low-level data for researchers to recognize high-level patterns.

Acknowledgements The work presented in this manuscript is based upon work supported by the National Science Foundation under Grant DUE #1348547 and DUE #1348530. Any opinions, findings, and conclusions or recommendations expressed in this paper, however, are those of the authors and do not necessarily reflect the views of the NSF.

References

Achieve (2013) The next generation science standards. Achieve, Washington, DC

Adams R, Fralick B (2010) Work in progress: a conceptions of design instrument as an assessment tool. In: Proceedings of the annual ASEE/IEEE frontiers in education conference, Washington DC, Oct 2

Adams RS, Beltz N, Mann L, Wilson D (2010) Exploring student differences in formulating cross-disciplinary sustainability problems. Int J Eng Ed 26(2):324–338

Apedoe XS, Schunn CD (2013) Strategies for success: uncovering what makes students successful in design and learning. Instruct Sci 41(4):773–791

Atman CJ, Bursic K (1998) Verbal protocol analysis as a method to document engineering student design processes. J Eng Ed 87(2):121–132

Atman CJ, Yasuhara K, Adams RS, Barker T, Turns J, Rhone E (2008) Breadth in problem scoping: a comparison of freshman and senior engineering students. Int J Eng Ed 24(2): 234–245

Atman CJ, Deibel K, Borgford-Parnell J (2009) The process of engineering design: a comparison of three presentations. In: International conference on engineering design (ICED), Palo Alto, CA

Baker RSJD, Corbett AT, Aleven V (2008) More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In: 9th international conference on intelligent tutoring systems, Montréal, Canada

Bienkowski M, Feng M, Means B (2012) Enhancing teaching and learning through educational data mining and learning analytics: an issue brief. Office of Educational Technology, U.S. Department of Education, Washington, DC

Clarke-Midura J, Dede C, Norton J (2011) Next generation assessments for measuring complex learning in science. Rennie Center for Education Research and Policy

Costa R, Sobek DK (2004) How process affects performance: an analysis of student design productivity. In: ASME international design engineering technical conferences and computers and information in engineering conference, Salt Lake City, Sept 28–Oct 2

Crismond DP, Adams RS (2012) The informed design teaching and learning matrix. J Eng Ed 101 (4):738–797

Crismond D, Hynes M, Danahy E (2010) The design compass: a computer tool for scaffolding students' metacognition and discussions about their engineering design process. In: Proceedings of association for the advancement of AI spring symposium

Cross N (2001) Design cognition: results from protocol and other empirical studies of design activity. In: Eastman CM, McCracken WM, Newstetter W (eds) Design learning and knowing: cognition in design education. Elsevier Press, New York

Dong A, Hill AH, Agogino AM (2004) A document analysis for characterizing design team performance. ASME J Mech Des 126(3):378–385

Goldschmidt G (2014) Linkography: unfolding the design process. MIT Press, Cambridge, MA

Goldstein MH, Purzer Ş, Adams RS, Xie C (2015a) Exploring the relationship between student reflectivity and their understanding of informed design. Presented at research in engineering education symposium (REES), Dublin, Ireland

Goldstein MH, Purzer Ş, Zielinski M, Adams RS (2015b) High school students' ability to balance benefits and tradeoffs while engineering green buildings. In: 122nd American Society for engineering education annual conference and exposition, Seattle, WA

Goldstein MH, Purzer Ş, Vieira Meji C, Zielinski MK, Douglas KA (2015c) Assessing idea fluency through the student design process. In: Conference proceedings: frontiers in education (FIE) annual conference, El Paso, TX

Hmelo CE, Holton DL, Kolodner JL (2000) Designing to learn about complex systems. J Learn Sci 9(3):247–298

Horwitz P (2011) Interactive technology for formative assessment: how we got here and what comes next. In: Noyce PE, Hickey DT (eds) New frontiers in formative assessment. Harvard Education Press, Cambridge

Hsu M, Cardella ME, Purzer Ş (2014) Assessing design. In: Purzer Ş, Cardella ME, Strobel J (eds) Engineering in pre-college settings: synthesizing research, policy, and practices. Purdue University Press, West Lafayette, pp 303–313

IBM (2012) What is big data? Retrieved 1 May 2013 from http://www-01.ibm.com/software/data/bigdata/

Kan JWT, Gero JS (2009) Using the FBS ontology to capture semantic design information in design protocol studies. In: McDonnell J, Lloyd P (eds) About: designing, analysing design meetings. CRC Press, London, pp 213–229

Katehi L, Pearson G, Feder M (2009) Engineering in K-12 education: understanding the status and improving the prospects. The National Academies Press, Washington, DC

Kolodner JL (2002) Facilitating the learning of design practices: lessons learned from an inquiry into science education. J Ind Teach Educ 39(3)

Lewis T (2006) Design and inquiry: bases for an accommodation between science and technology education in the curriculum? J Res Sci Teach 43(3):255–281

McElhaney KW, Linn MC (2011) Investigations of a complex, realistic task: intentional, unsystematic, and exhaustive experimenters. J Res Sci Teaching 48(7):745–770

Mehalik MM, Doppelt Y, Schunn CD (2008) Middle-school science through design-based learning versus scripted inquiry: better overall science concept learning and equity gap reduction. J Eng Ed 97(1):71–85

Mentzer N, Becker K, Sutton M (2015) Engineering design thinking: high school students' performance and knowledge. J Eng Ed 104(4):417–432

National Research Council (2010) Standards for K-12 engineering education? The National Academies Press, Washington, DC

Purzer Ş (2011) The relationship between team discourse, self-efficacy, and individual achievement: a sequential mixed-methods study. J Eng Ed 100(4):655–679

Purzer Ş, Fila ND (2013) Innovation process mapping protocol: an approach to assessing students' understanding of innovation as a process. In: ASEE annual conference and exposition

Purzer Ş, Goldstein MH, Adams RS, Xie C, Nourian S (2015) An exploratory study of informed engineering design behaviors associated with scientific explanations. Int J STEM Ed 2(1):1–12

Sao Pedro MA, Baker R, Gobert J, Montalvo O, Nakama A (2013) Leveraging machine-learned detectors of systematic inquiry behavior to estimate and predict transfer of inquiry skills. User Model User-Adap Interact 23(1):1–39

Sims-Knight JE, Upchurch RL, Pendergrass N, Meressi T, Fortier P, Tchimev P, VonderHeide R, Page M (2004) Using concept maps to assess design process knowledge. In: Proceedings of the 34th ASEE/IEEE frontiers in education conference, Savannah, GA

Socha D, Gornil A, Watler S, Adams RS, Tenenberg J, Roth WM (2016) Exploratory analysis of large ethnographic datasets. In: Proceedings of a workshop on developing a research agenda for human-centered data science as part of the annual computer supported cooperative work conference, San Francisco, Feb

Svihla V, Petrosino AJ (2008) Improving our understanding of K-12 engineering education. Paper presented at the international conference on engineering education

U.S. Department of Education (2010) National educational technology plan 2010: transforming american education: learning powered by technology. Office of Educational Technology, U.S. Department of Education, Washington, DC

Valkenburg R (1998) The reflective practice of design teams. Des Stud 19(3):249–271

van Schaik J, Scanlan J, Keane A, Takeda K, Gorisse D (2011) Improving design rationale capture during embodiment design. In: International conference on engineering design (ICED), Copenhagen, Denmark, 15–19 Aug 2011

Vattam SS, Kolodner JL (2008) On foundations of technological support for addressing challenges facing design-based science learning. Pragmat Cogn 16(2):406–437

Worsley M, Blikstein P (2014) Analyzing engineering design through the lens of computation. J Learn Anal 1(2):151–186

Xie C, Zhang H, Nourian S, Pallant A, Bailey S (2014a) A study of the instructional sensitivity of CAD logs based on time series analysis. Int J Eng Ed 30(4):760–778

Xie C, Zhang H, Nourian S, Pallant A, Hazzard E (2014b) A time series analysis method for assessing engineering design processes using a CAD tool. Int J Eng Ed 30(1):218–230

Generalizability of Document Features for Identifying Rationale

Benjamin Rogers, Connor Justice, Tanmay Mathur and Janet E. Burge

Abstract One of the challenges in using statistical machine learning for text mining is coming up with the right set of text features. We have developed a system that uses genetic algorithms (GAs) to evaluate candidate feature sets to classify sentences in a document. We have applied this tool to find design rationale (the reasons behind design decisions) in two different datasets to evaluate our approach for finding rationale and to see how features might differ for the same classification target in different types of data. We used Chrome bug reports and transcripts of design sessions. We found that we were able to get results with less overfitting by using a smaller set of features common to the set optimized for each document type.

Introduction

A significant amount of Software Engineering (SE) research uses data mining to look for interesting patterns in development artefacts such as documentation or code. Some of these techniques fall into the category of text mining, where we try to extract useful information from text documents. We have been working on the task of looking for design rationale (discussion of design decisions) in documents using statistical machine learning. This has been a challenging task for many reasons and we have experimented with varying the classification algorithms (Rogers et al. 2012) and data features (Rogers et al. 2012, 2014) to get better results. One of the difficulties we had with our initial dataset, Chrome bug reports, was the sparcity of the data. Our results sent us working in two directions: searching for an "optimal" feature set and looking at less sparse datasets to see if we could achieve better results.

B. Rogers · T. Mathur
Miami University, Oxford, OH, USA

C. Justice · J.E. Burge (✉)
Wesleyan University, Middletown, CT, USA
e-mail: jburge@wesleyan.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_34

We developed a system called GAFFS, Genetic Algorithms For Feature Set optimization, which uses a genetic algorithm (GA) to evaluate different feature combinations to look for the set that gave the best results (where "best" was the F-measure resulting from 10-fold cross validation). Our hope was that the computer-generated combinations might discover some with better results than feature combinations hand-crafted by the researchers. We also annotated a dataset from the Studying Professional Software Design (SPSD) workshop (http://www.ics.uci.edu/design-workshop/) to see if this data would be less sparse and give us better results.

The results obtained in our initial experiments comparing the two datasets were interesting. As expected, the SPSD dataset was significantly richer in rationale than the Chrome Bug Reports. The Chrome Bug Reports had 10.9% of the sentences containing rationale while the SPSD data had 52.4%. We expected that using the less sparse dataset as input to our learning algorithms would result in higher classification accuracy. This assumption was incorrect. Instead the accuracy was poorer (Mathur 2015). The feature set used was one we had identified during some of our early experiments using GAFFS, which poses the questions:

- How different are the feature sets needed to look for the same classification target in two different datasets?
- Is there a set of features in common that we could use to avoid having to use the GA to create custom feature sets for each set of data?

In this paper we discuss the following. In "Motivation and Issues" section we give a brief introduction to the motivation for and issues behind extracting rationale. "Related Research" section presents related work. In "Genetic Algorithm For Feature Set Optimization" section we describe GAFFS. In "Experiment Design" section we describe our experiments in comparing feature optimization results for the two datasets. "Results" section gives our results. We then end with our summary and conclusions in "Conclusions and Future Work" section.

Motivation and Issues

We have been working towards our ultimate goal of mining rationale from text documents by using text mining to classify (identify) which sentences contain rationale. Our approach uses features found in the documents being classified as input into machine learning classifiers to build models that can be used to classify additional data. Finding good feature sets for text mining is a critical part of this process. The number of potentially useful text features (such as the parts of speech of words, specific words, or the length of sentences), and combinations of these features, is very large. While it is sometimes possible to use all of the features when text mining, this can cause problems when noisy features, features that do not relate to whether a sentence contains rationale or not, are used and less then optimal results are returned. Using all of the features is also very expensive when considering the time and space requirements of a text mining algorithm. Our earlier work investigated using vocabulary terms (Rogers et al. 2012) and linguistic features (Rogers et al. 2014) along with a variety of machine learning algorithms.

In the work described in this paper, rather than working with "hand crafted" feature sets, we used a genetic algorithm to search for optimal or near optimal feature sets to identify rationale. Genetic algorithms imitate natural selection, by having possible feature sets represented as individuals of a population. How well a feature set performs when used in text mining determines that individual's fitness, which, in turn, determines how likely that individual will "mate" and pass on its features to later, higher performing feature sets. The genetic algorithm constructed for these experiments is called GAFFS, or the Genetic Algorithm For Feature Set optimization. Our approach used GATE (General Architecture for Text Engineering) (Cunningham et al. 2002) to annotate our text corpus with rationale (manually) and linguistic features (automatically) and WEKA (Waikato Environment for Knowledge Analysis) (Hall et al. 2009) to build the machine learning classifiers.

The goals of this work were to create a mechanism for generating and evaluating GA-generated feature sets and to evaluate these feature sets to see which were more effective at extracting rationale. We evaluated effectiveness by looking at the F-1 measure of the accuracy of the classification and at the number of features used to build the classification model.

Related Research

The following subsections describe related research in two areas: feature set selection and rationale extraction.

Feature Set Selection

The information contained in natural language documents can only be utilized by machine learning models if it is separated into discrete pieces of data. An important task of natural language processing is figuring out how to discretize the information contained in the document, preserving context and other meta-information in a way that is useful to the models. These features, such as n-grams, parts of speech, and sentence length, can then be passed on to machine learning classifiers.

While generating many different features for a data set or document is a good first step, problems start arising when the whole group of features is sent to classifiers. Features that are irrelevant or noisy can confuse classifiers that will try to fit to them and lead to sub-optimal performance than if only a subset of the features were used. Wasikowski and Chen (2010) addresses the problems caused by unbalanced data by using feature selection among other methods. On average, feature selection produced the most improvement when compared to selection and algorithm solutions.

Filtering is a process where a filter algorithm picks out features and attributes that it thinks are important and throws the rest away. Wrappers are similar, but they rely on the classifier to determine what features and attributes are important. The problem that arises with filters and wrappers is that they sometimes, if not often, take more time than the classifiers they are filtering or wrapping to run, which can end up being expensive if the classifier takes a very long time to run.

Fortunately, the time it takes wrappers and filters to run is a function of how many attributes they have to look at to determine the most relevant ones. If irrelevant features can be removed from the set sent to the filter or wrapper and then to the classifier, speedup will be achieved with little to no loss in performance.

Salcedo-Sanz et al. (2002) used a genetic algorithm as part of a wrapper in order to provide timely accuracy, however, it is limited when applied to our large problem, where it is still too slow because it learns over the super-set of features. Tan (2007) addresses the problem of feature selection by using a genetic algorithm and a hybrid filter. When used on the 20 Newsgroup and Reuters-21578 datasets, small to modest gains are observed over information gain filtering. Oliveira et al. (2010) addresses the problem of feature selection again by using a number of genetic algorithms this time on six datasets. These experiments showed that the genetic algorithms produced small to moderate gains over classification without genetic algorithms. However, these datasets were related to software man/hour cost estimation, and did not cover design rationale identification.

GAs have also been used in classification tasks for Software Engineering. One example is Yi et al.'s (2012) work using a genetic algorithm to optimize a support vector machine to mine binary constraints from partially constructed feature models. The GA is used to optimize the input parameters to LibSVN: the weight of the two target classes (requires and excludes relationships) and a γ parameter needed by LibSVN.

A GA was also used to configure Latent Dirichelet Allocation (LDA) parameters for several different SE text analysis tasks (Panichella et al. 2013). Each candidate solution contained the four parameters needed to configure the LDA: the number of topics, the number of Gibbs iterations, the topic distributions, and term distribution per topic.

Rationale Extraction

Design rationale can be found in relatively unstructured informal documents, such as inter-developer emails, structured informal documents, such as bug reports, and structured formal documents, such as requirements documentation. The amount of design rationale contained in a specific document can range from very sparse, to very full depending on the document's intent. Different researchers have different rationale notations and even definitions.

Liang et al. (2012) focused on extracting design rationale from patent documents. A three-tiered model was used to capture issues, design solutions, and artefacts. First, artefacts were identified using a modified PageRank (Brin and Page 1998) algorithm on frequently appearing words. Next, issue summarization is performed using issue language patterns as part of manifold ranking. Finally, reason language patterns are used to identify reason sentences, which are then paired with the remaining solution sentences to create reason-solution pairs. They achieved a .185 F-measure for artefact identification, a .520 F-measure for issue summarization, and a .562 F-measure for reason-solution extraction.

Toeska Rationale Extraction (TREx) (López et al. 2012) is based on GATE and extracts "knowledge units," only some of which correspond to rationale. It does this using manually created extraction rules to identify properties defined in architecture and rationale. They used this tool to extract knowledge units from 26 pages of architecture documents and achieved an F-1 value of 50% for their aggregated extraction results. This approach used Information Extraction NLP techniques rather than the statistical Machine Learning approach used in our work.

A similar, although not identical, problem is identifying arguments in legal texts. Palau and Moens (2009) reported an accuracy of 73% on to a corpus of sentences where 50% of the sentences contained arguments and 80% on a corpus of legal texts. Their approach used linguistic features (specifically modal auxiliaries, adverbs, and verb tense) along with n-grams and keywords.

Our earlier work (Rogers et al. 2012) explored using ontologies to provide feature sets with a large number of different classifiers to extract design rationale from Chrome bug reports. The best F-1 measure achieved was 59.7% for binary classification (rationale/not rationale). Using linguistic features and n-grams rather than the ontologies improved the classification to 67.6% for binary rationale and 56.9% for the argumentation subset (Rogers et al. 2014). The two papers used different datasets, with the (Rogers et al. 2014) set following a more rigorous annotation process (this is the data we use here).

Genetic Algorithm for Feature Set Optimization (GAFFS)

The goal of GAFFS is to use a genetic algorithm to determine optimal feature sets. The chromosomes are the feature combinations and the fitness function is the F-measure output by a machine learning classifier. This process used the following steps:

- Annotating data to produce test and training sets. We did this using the process described in our earlier work (Rogers et al. 2014).
- Extracting the features from the documents in our training set to produce data we could then use to train and test our classifier. The data was split into folds at this time so that it could be used in cross-validation.

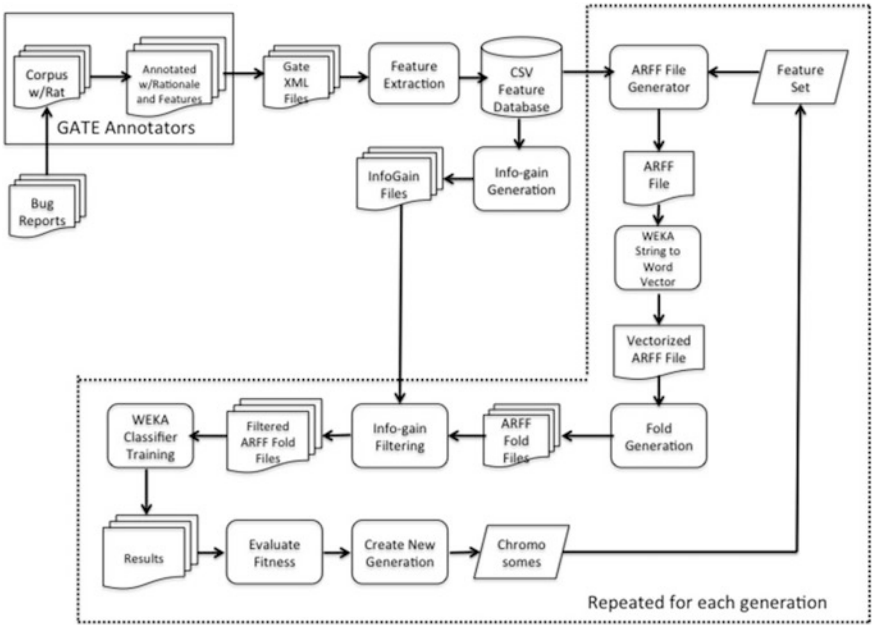


Fig. 1 GATE–WEKA pipeline for classifier building

- Calculate the Information Gain for each feature instance for each fold. This was time consuming so we developed our own Information Gain implementation and calculated it for each feature and fold outside of the GA.
- [Looping for all generations] Create the chromosomes for use in the genetic algorithm. Each chromosome would be a feature set (a set of feature types).
- Use the chromosomes to extract the relevant features from the documents to build the Attribute-Relation File Format (ARFF) files needed by WEKA to build the classifier. The information gain calculated earlier is used to reduce the number of instantiated features used to 1000.
- Build and evaluate a classifier for each feature set, using 10-fold cross validation with the folds we created earlier.
- Use the fitness of each chromosome (the classification result) to create a new set of chromosomes to evaluate during the next generation.

Figure 1 shows the processing steps used by GAFFS.

Feature Extraction

The feature extraction algorithm is used to pull the features out of the documents prior to classification so that this part of the process would only need to be performed once rather than for each iteration of the genetic algorithm. The extracted features can then be combined to form each feature set evaluated as candidates for building the final classification model. The XML documents exported from GATE have each document annotated with tokens (individual pieces of text separated by white space, such as words), sentences, and feature categories. The feature extractor then pulls this information into a CSV file that maps each input sentence to the features present. Each row of the CSV file corresponds to a sentence and contains the name of the XML file that the sentence came from, the sentence ID, the output of feature generation for each feature (including closest sentences as separate features), the type of rationale that is or is not present, and the fold number (the fold numbers are generated randomly and are used later when the dataset is split into cross-validation folds).

Simplified Information-Gain Filtering

Each training set (one for each fold) then goes through simplified information-gain filtering that selects the 1000 most relevant learning attributes from the training set and creates a new ARFF file, with only the 1000 most relevant attributes included. In initial tests, where WEKA's info-gain filtering was used, it became obvious that filtering was taking up much more time than classification, and was increasing the length of time necessary to run experiments. For this reason, we developed our own simplified version that separated out the information gain calculation from the information gain filtering so that the calculation only needed to be performed once for the data set. Information gain filtering, or info-gain, is a simple filter that tries to measure the information entropy of attributes based on the number of times it shows up in rationale and non-rationale containing sentences. Information entropy can be estimated by the absolute value of the difference of the number of times an attribute is included in a rationale sentence and the number of times an attribute is included in a non-rationale sentence divided by the total number of occurrences of that attribute, shown in (1).

$$\frac{|A_{rat} - A_{non-rat}|}{A_{total}} \tag{1}$$

A_{rat} is the number of times an attribute is found in a rationale containing sentence, $A_{non\text{-}rat}$ is the number of times an attribute is found in a non-rationale containing sentence, and A_{total} is the number of times an attribute is found in any sentence. This means that attributes that occur more frequently in either rationale or non-rationale sentences have lower entropy.

For each feature and each fold, a list of the attributes and their info-gain score were generated and then sorted. Each list was then saved as a text file, where each line contained an attribute and its info-gain score where the best scoring attribute was recorded at the beginning of the document and the worst scoring attribute was recorded at the end. This was used in filtering to choose the 1000 attributes with the best info-gain scores.

The Genetic Algorithm contains three main components:

- Capturing feature sets as chromosomes
- Using the WEKA classification results and feature set size to compute the fitness of each chromosome (feature set)
- Generating a new population of chromosomes through selection, mating and mutation.

These components are described in the following sections.

Feature Set Chromosomes

Chromosomes describe the feature sets used to build the input ARFF files and consist of the binary inclusion of all of the possible features. They are initially generated randomly with a uniform distribution.

After the chromosomes are used in ARFF generation, they are saved as binary files, and loaded back in at the end of classification, for combination and mutation. An example chromosome showing which features are included in the set is shown in Fig. 2. In this example, the features shown are parts of speech and the numbers shown before them are the n-gram count in parenthesis and the number of neighbouring sentences included in curly braces.

Fitness Calculation

After each chromosome is loaded in after classification, each chromosome is given a fitness based on how long the classification took and what the chromosomes' F-measure was. The fitness function determines how "fit" a chromosome is, that is, how likely that chromosome will produce offspring for the next generation. The fitness functions use the F-measure generated by WEKA's 10-fold cross validation for each chromosome. The F-measure was multiplied by one minus the fraction of the number of features in the feature set and the number of possible features in that set. Fitness is calculated by (2).

nouns	(2)nouns	(3)nouns	...	{2}(3)verbs	{2}(4)verbs	{2}(5)verbs
yes	no	no	...	no	yes	yes

Fig. 2 Chromosome example

$$fitness = F_{measure} \times \left(1 - \frac{Setnum}{Setpossible}\right) \quad (2)$$

where *Setnum* is the number of features included in the feature set and *Setpossible* is the number of possible features in that set.

Population Generation

After each chromosome is given a fitness score, the GA needs to select which ones should be used to produce the new generations. The fitness function is transformed according to Boltzmann Selection (de la Maza and Tidor 1993) before using roulette wheel selection to get the next generation. Boltzmann Selection is used to preserve diversity in early generations, but to force convergence in later generations. The post-Boltzmann Selection fitness is equal to the fitness divided by the quotient of the generation number and the total number of generations, all over the average of the finesses divided by the quotients of the generation number and the total number of generations.

Roulette wheel selection then gives each chromosome a slice of the wheel proportional to its fitness where the proportion is modified so that the beginning generations allow less fit individuals a greater chance of producing offspring, preserving some initial variance.

For each pair of parents selected, two new chromosomes are created by copying the old chromosomes. Every gene (feature) has a chance of being selected as a crossover point candidate, referred to here as the crossover candidate rate. This is different than some other genetic algorithms that have a set number of crossover candidates for each chromosome (Mitchell 1996). Doing it this way removes location bias from combination. Every crossover point candidate has a chance of being selected as a crossover point, referred to here as the crossover rate. After the combination of the two parent chromosomes, switching at the crossover points, each gene on each child chromosome is then subject to random mutation (Fig. 3).

Experiment Design

The following sections describe key components of our experiments: the dataset used in the experiments, our classification goal, the types of features considered, and the GA parameters.

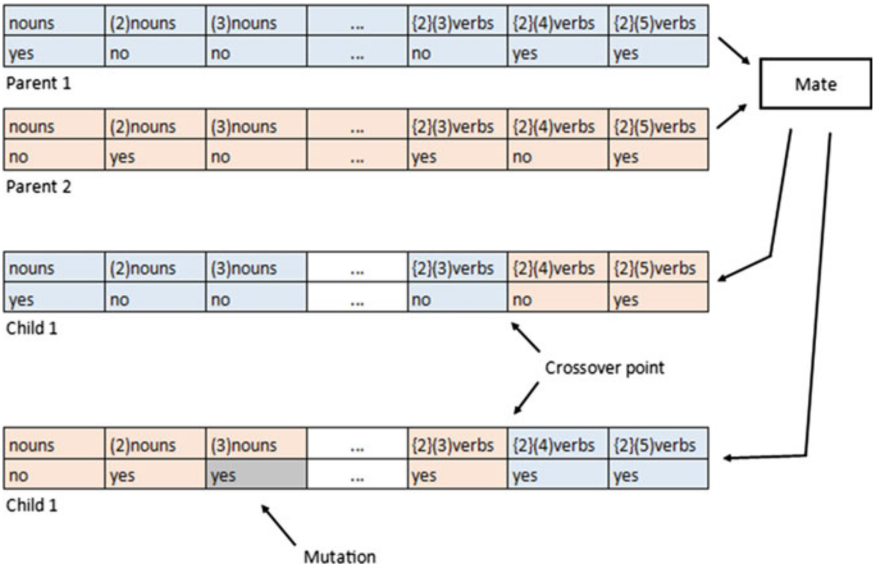


Fig. 3 Combination and mutation example

Input Dataset

We used two datasets for this experiment. The first was a set of 200 Chrome bug reports extracted randomly from a subset of the data provided for the Mining Software Repositories 2011 mining challenge (http://2011.msrconf.org/msr-challenge.html). We started our work using bug reports because we found that they contained discussion between the developers about how the bug should be repaired. The data was annotated by having two researchers annotate each document and a third researcher adjudicate. The bug report data contained significant amounts of rationale but is still considered a relatively sparse data set because the percentage of rationale is low—under 11%. Details on how bug reports were selected and annotated can be found in Rogers et al. (2014).

The second was from the Studying Professional Software Designers (SPSD) project. This data consisted of three interview transcripts, each of which had two experienced software designers brainstorming a design for a traffic-light simulator. The same annotation process was used for this dataset. The SPSD dataset was more balanced—52.4% of the sentences contained rationale.

Classification Goals

The classification goals of this set of experiments were the same as in our earlier work—identification of rationale. The data was annotated looking for specific types of rationale:

- *Requirements*—statements that the software was required to do something;
- *Decisions*—statements that indicated what issue had to be resolved. Bug report titles indicated decisions although others could arise in the text;
- *Alternatives*—different options for resolving the issue described by a decision;
- *Arguments*—reasons for or against an alternative;
- *Assumptions*—claims made in the bug report where the author indicated uncertainty;
- *Questions*—questions posed in a bug report. These often appeared as part of "boiler plate" occurring in each bug report.
- *Procedures*—description of actions needed to gain information required to answer a question or make a decision.
- *Answers*—answers to questions posed in the bug report.

A sentence can be classified as more than one type of rationale.

The experiments described here looked to see if a sentence contained argumentation, i.e. Requirements, Decisions, Alternatives, Arguments, or Assumptions. These were grouped into one category. Our earlier work referred to this as the Argumentation Subset.

Candidate Classification Features

Our earlier work looked at two types of features—specific domain vocabulary words (Rogers et al. 2012) and linguistic features (Rogers et al. 2014). Features are the most important part of the machine learning process since those are what the classifiers are using to determine if rationale is or is not present. We used a total of 753 different feature types as candidates for inclusion in our feature set.

- *N-grams* are series of N adjacent words concatenated with each other. When N-grams are referred to in this document, they may alternatively be called terms like "3-grams" or "bigrams", but the "-gram" suffix designates them as made up of actual words. N-grams can provide strong correlation information, but at the cost of an extremely large set of attributes to learn over.
- *Parts of speech (POS)* are Penn-Treebank (Marcus et al. 1993) parts of speech for a word. Like N-grams, they can be concatenated adjacently by adding a number suffix, "2-pos" for example. Parts of speech can provide some correlation information, but tends to be limited due to the extremely small set of attributes to learn over.

- *N-grams* for specific parts of speech are the word instances of a specific part of speech. Only verbs, for example. They can also be concatenated in the same manner as above. N-grams for specific parts of speech can provide strong correlation information, as N-grams, but also cut down the set of attributes to learn over, due to only having a subset of N-grams.
- *General and domain terms* are like N-grams for specific parts of speech, but instead of being a specific part of speech, the word must be on either a general rationale argument vocabulary list (Burge 2005) or a security domain vocabulary list (AuditMyPC 2010). The general vocabulary list is derived from an argument ontology that contains common reasons for making software design decisions. The security domain vocabulary terms list is a list of terminology that applies specificity to web services, in order to more accurately target the data used (Chrome bug reports). It has similar benefits to the N-grams for specific parts of speech, with two exceptions: First, because the lists are created with rationale containing sentences in mind, general and domain terms are more likely to be relevant. Second, because the lists are created before looking at the data, general and domain terms may overlook words that have strong correlation but are not on the list, limiting discovery.
- *Sentence length* is the number of words in a sentence. While obviously limited, it can be used to find sentences that are unlikely to contain rationale, such as those that are very short (1–4 words).
- *Acronyms* were annotated by looking for words that did not appear in the WordNet dictionary. They were added as feature because an individual acronym might not appear frequently enough to impact classification on its own but the presence or lack of technical terminology might indicate rationale.
- *URLs* were also tagged in the rationale as features. We did this since it was likely that rationale would not include URLs and we hoped this would make it easier to exclude this text.

N-closest sentences allows for context to be shared between sentences. When any of the above features uses closest sentences, that sentence's attributes as well as adjacent sentence's attributes within N (so 1-closest sentences include the sentence before and after, 2-closest the two sentences before and after) are weighted (to emphasize nearer sentences) and added as a feature. This way a sentence can indicate the presence or absence of rationale around it.

GA Parameters

The GA has a number of parameters that can be adjusted. These are the population size, the number of generations, the combination rate, and the mutation rate.

To make sure that all of the features had a good chance to be in a set and have an impact to that set, the population size was set to 100.

There are usually two ways to decide the number of generations a genetic algorithm will have: iterate until there is little to no improvement of the best fitness, or choose the number of generations beforehand (Mitchell 1996). Because Boltzmann selection is being used, the number of generations is needed beforehand. Two things dictate how many generations should be run, the number of generations it takes for little to no increase in the fitness between generations and the time available for experiments. Preliminary trials showed that a fairly small number of generations was sufficient to reach convergence so we chose to work with 20. Figure 4 shows that the GA appears to converge fairly quickly.

Because of the small number of generations, the combination and mutation rates should be higher than average, in order to promote greater variance. Combination describes the process of taking two parent chromosomes and diving their genes between two child chromosomes. Every gene has a chance of being selected as a crossover point candidate, referred to here as the crossover candidate rate. This is different than some other genetic algorithms that have a set number of crossover candidates for each chromosome (Mitchell 1996). Doing it this way removes location bias from combination. After the combination of the two parent chromosomes, switching at the crossover points, each gene on each child chromosome is then subject to random mutation, the chance of which is known as the mutation rate.

The crossover candidate rate was chosen to be 1/7 (.14), the crossover rate was chosen to be .7 and the mutation rate was chosen to be .01.

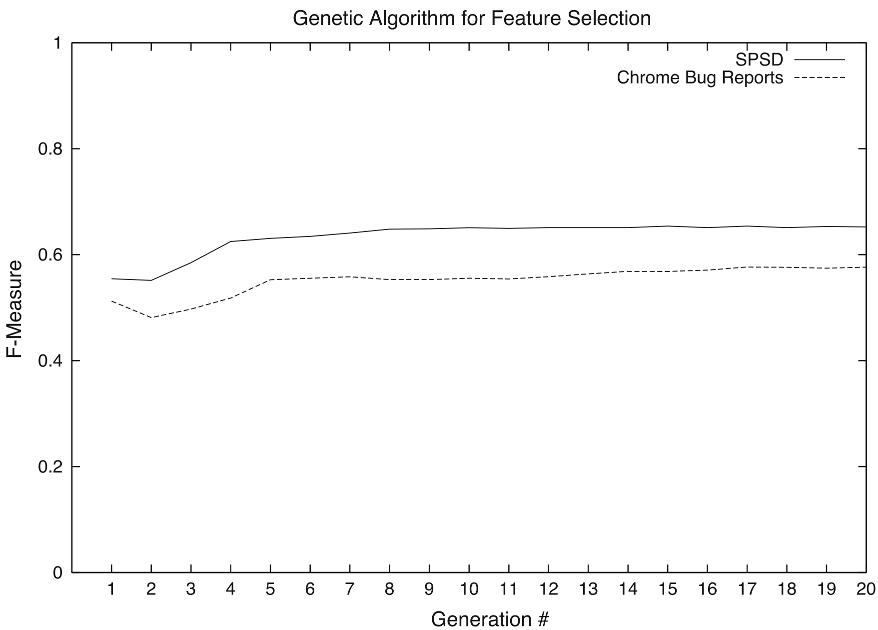


Fig. 4 F-measure over generation

Results

After the experiments were run, the results were collected and then evaluated. When a GAFFS experiment ends, it produces a text file that contains all of the feature sets of the last generation, with their F-measure. They are also sorted by F-measure so that the best performing individual is the first on the list.

We were interested in answering the following questions:

- How different are the feature sets needed to look for the same classification target in two different datasets?
- Is there a set of features in common that we could use to avoid having to create custom feature sets for each type of data?

Our earlier experiments used all 200 Chrome bug reports to build our classifiers, which were evaluated using 10-fold cross-validation. In the GAFFS experiments we used 70% of our corpus for training, reserving 30% for testing the classifiers. Earlier work used a variety of classification algorithms while these experiments only used the Naïve Bayes Multinomial classifier because it took less running time. We ran our experiments using the Chrome bug reports and the SPSD data.

Classification Accuracy

Table 1 gives the results for classifying the Chrome bug reports using GAFFS to generate optimal feature sets. This is displayed as precision (P), recall (R), and F-1 measure (F)—the harmonic mean of precision and recall. The last row of the table gives the Inter-Annotator agreement F-1 measure when we compared the results from the two human annotators. The results from GAFFS had a better accuracy than the human annotators when training (with 10-fold cross validation) and were close in value for the test set. We did improve on our earlier experiments with Naïve Bayes and hand-crafted feature sets but were only slightly better than the results we achieved using WEKA's Stacking classifier with SGD, PART, and BayesNet (which runs much more slowly than Naïve Bayes).

Table 2 gives the results for classifying the SPSD data using GAFFS generated optimal feature sets. The accuracy was better than the human annotators when training (using 10-fold cross validation) but were much worse when testing.

Table 1 Chrome bug report results

Strategy	P	R	F
GAFFS (train)	.446	.815	.576
GAFFS (test)	.288	.859	.432
Naïve Bayes (Rogers et al. 2014)	.857	.209	.335
Stacking (Rogers et al. 2014)	.698	.484	.569
Inter-Annotator Agreement			.461

Table 2 SPSD data results

Strategy	P	R	F
GAFFS (train)	.576	.752	.652
GAFFS (test)	.630	.246	.354
Inter-Annotator Agreement			.61

Table 3 F-measures bug report and SPSD comparison

Data set	Chrome bug report optimized	SPSD optimized
Bug Report (10-fold cross validation)	**.576**	.422
Bug Report (test)	.432	.439
SPSD (10-fold cross validation)	.625	**.652**
SPSD (test)	.466	.354

Best results for feature set

The SPSD data had a higher percentage of rationale than the bug reports but was a much smaller dataset overall with 2310 sentences versus the 19,521 in the bug reports.

We also wanted to see how the feature sets optimized for one dataset would work on the other—how big an impact is the feature set versus the data? When we used 10-fold cross validation and the training data the results were worse when we used features optimized on the other data but when we were running in test mode (the training data built the classifier and the test data tested it) the results improved when the features were swapped, by almost 10% for the SPSD data. Table 3 shows these results.

Comparison of Feature Sets

We used 753 different features as input to the Genetic Algorithm. These can be broken into types:

- Specific part of speech features—these are the 36 Penn-Treebank features (Marcus et al. 1993).
- General part of speech features—these are generic parts of speech—adjectives, adverbs, conjunctions, nouns, pronouns, verbs.
- Ontology terms—security ontology terms (AuditMyPC 2010), SEURAT argument ontology terms (Burge 2005), and both sets combined.
- Verb roots—stemmed verb values
- Parts of speech types (Marcus et al. 1993)—converting specific words to their POS type to look for type patterns.
- Bag of words (unigrams)—combinations of adjacent words.
- Acronyms and URLs. We tagged URLs as features and also any words that we could not find in WordNet (which would capture acronyms).

- Sentence length—the number of words in a sentence.

For each of these (except the sentence length) we looked at N-grams from 1 to 5. We also include a two more sets of features that combined data from the sentence we were classifying and the 1 and 2-closest sentences (the sentences immediately before and the immediately after). The closest sentences were used to try to capture the influence that context might have on classification.

When we compared the feature sets for the Chrome Bug Reports and SPSD data we found they had 14 features in common:

- Conjunctions
- 2-gram verbs in 1-closest sentences
- Symbols in 1-closest sentences
- 4-gram preposition or subordinating conjunctions in 2-closest sentences
- 2-gram adjective, comparative in 2-closest sentences
- Preposition or subordinating conjunctions
- 3-gram verb, past participle in 2-closest sentences
- 5-gram determiner
- 2-gram personal pronoun in 1-closest sentences
- 3-gram proper noun, singular in 2-closest sentences
- Unigrams (bag of words)
- Domain ontology terms in 2-closest sentences
- 3-gram conjunctions in 2-closest sentences
- Determiners

Common Feature Classification Accuracy

We used this common set of features with the training sets in cross-validation mode and using the training data to train and the test data to test. The results are given in Table 4.

The common feature set gave comparable results to the optimized features with the exception of the Chrome bug report training set, where it had a much worse performance. Precision and recall were similar for the training and test data but the recall was much worse for the training data. At this point we can only speculate on the reason. One difference in the test and training runs is that when we were using

Table 4 Results with a common feature set

Data set	P	R	F
Chrome Bug Report (training w/10-fold cross validation)	.300	.463	.364
Chrome Bug Report (test)	.384	.881	.534
SPSD (training w/10-fold cross validation)	.535	.683	.600
SPSD Test	.657	.660	.657

cross-validation we used our own InfoGain calculations for the information gain filtering while when we did the test runs we had WEKA do the information gain filtering for us.

Conclusions and Future Work

The experiments in this paper were designed to answer the following questions:

- How different are the feature sets needed to look for the same classification target in two different datasets?
- Is there a set of features in common that we could use to avoid having to create custom feature sets for each type of data?

The feature sets were similar in the total number of features (only one different) but only had 14 features in common (a less than 2% overlap). The SPSD features had 14 more features that utilized neighboring features than the Chrome set. For N-grams, the SPSD features had more 1- and 2-grams but fewer 5-grams. The number of specific parts of speech were the same, general were only different by one. The SPSD features used more general ontology terms while the Chrome features used more domain ontology terms. The SPSD features used acronyms and URLs but not sentence length while the Chrome features used sentence length but no acronyms or URLs. The Chrome features used more verb roots and general parts of speech while the SPSD features used more unigrams (bag of words).

With the exception of the Chrome training data in cross-validation, we were able to get similar or better results when we ran experiments with the smaller common feature set.

We have a number of future experiments planned that build on this work. These involve updates to the GA fitness function, adding learning algorithm optimization, identifying additional features, and working with new datasets and classification goals. We also plan to run some experiments with different algorithms. We will start with some of the algorithms that gave good results in our previous papers. We also have the GA set up so that we can add the algorithms to the chromosome and optimize their selection along with the feature sets.

While 753 is a large number of candidate feature types it is still possible we may have missed some. We plan on studying the miss-classified instances to see if there are any common features we need to add to our algorithms.

GAFFS can be configured to classify any target, not just rationale. We would like to apply this process to other text mining tasks to see how effective the approach will be. Classifying rationale has proved to be a challenging task, as shown by the inter-annotator agreement values, but the techniques we have developed can be applied to classify any text data.

Acknowledgements We would like to thank Miami graduate students John Malloy and Jennifer Flowers for their work in annotating the SPSD data. The design sessions that produced the SPSD data were funded by the National Science Foundation (Award CCF-0845840). We would like to thank the workshop organizers, André van der Hoek, Marian Petre, and Alex Baker for granting access to the transcripts. We would also like to thank Dr. Mike Zmuda for suggesting we move the information gain calculation outside of the GA. This work was supported by NSF CAREER Award CCF-0844638 (Burge). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

References

AuditMyPC (2010) Glossary of internet security. http://www.auditmypc.com/glossary-of-internet-security-terms.asp. Retrieved 23 Nov 2010

Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 30:107–117

Burge J (2005) Software engineering using design RATionale. Ph.D. thesis, Worcester Polytechnic Institute

Cunningham H, Maynard D, Bontcheva K, Tablan (2002) GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th anniversary meeting of the Association for Computational Linguistics (ACL'02). Philadelphia, July 2002

de la Maza M, Tidor B (1993) An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In: Forrest S (ed) Proceedings of the 5th international conference on genetic algorithms. Morgan Kaufmann, San Francisco, pp 124–131

Hall M, Frank E, Holmes G, Pfahringer B, Reutmann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor 11(1):10–18

Liang Y, Liu Y, Kwong C, Lee W (2012) Learning the 'whys': discovering design rationale using text mining—an algorithm perspective. Comput Aided Des 44(10):916–930

López C, Codocedo V, Astudillo H, Cysneiros LM (2012) Bridging the gap between software architecture rationale formalisms and actual architecture documents: an ontology-driven approach. Sci Comput Program 77(1):66–80

Marcus M, Marcinkiewicz M, Santorini B (1993) Building a large annotated corpus of English: the penn treebank. Comput Linguist 19(2):313–330

Mathur T (2015) Improving classification results using class imbalance solutions & evaluating the generalizability of rationale extraction techniques. Master of Computer Science thesis, Miami University. https://etd.ohiolink.edu/ap/10?0::NO:10:P10_ETD_SUBID:100565

Mitchell M (1996) An introduction to genetic algorithms. MIT Press, Cambridge

Oliveira A, Braga P, Lima R, Cornelio M (2010) GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Inf Softw Technol 52:11

Palau M, Moens M-F (2009) Argumentation mining: the detection, classification and structure of arguments in text. In: Proceedings of the 12th international conference on artificial intelligence and law (ICAIL '09). ACM, New York, pp 98–107

Panichella A, Dit B, Oliveto R, Di Penta M, Poshyvanyk D, De Lucia A (2013) How to effectively use topic models for software engineering tasks? An approach based on genetic algorithms. In: Proceedings of the international conference on software engineering, pp 522–531

Rogers B, Gung J, Qaio Y, Burge JE (2012) Exploring techniques for rationale extraction from existing documents. In: Proceedings of international conference on software engineering. IEEE Press, pp 1313–1316

Rogers B, Qaio Y, Gung J, Mathur T, Burge J (2014) Using text mining to extract rationale from existing documentation. In: Gero J (ed) Design, computing, and cognition. Springer

Salcedo-Sanz S, Prado-Cumplido M, Perez-Cruz F, Bousono-Calzon C (2002) Feature selection via genetic optimization. In: Dorronsoro JR (ed) Proceedings of the international conference on artificial neural networks (ICANN '02). Springer, London, pp 547–552

Tan F (2007) Improving feature selection techniques for machine learning. Ph.D. dissertation. Georgia State University, Atlanta. Advisor(s) Anu G. Bourgeois. AAI3293841

Wasikowski M, Chen X (2010) Combating the small sample class imbalance problem using feature selection. IEEE Trans Knowl Data Eng 22(10):1388–1400

Yi Z, Zhao J, Mei (2012) Mining binary constraints in the construction of feature models. In: Proceedings of the IEEE international requirements engineering conference (RE 2012). IEEE, pp 141–150

The Topology of Social Influence and the Dynamics of Design Product Adoption

Somwrita Sarkar and John S. Gero

Abstract This paper presents the results of studies on how the dynamics of design product adoption is affected by the topological structure of social communication and influence between consumers, without any changes in the designed product. The dynamics of product adoption are studied over random, modular small world, and scale free social network structures, under local rules of communication. Results show global behaviors emerging from these local agent communication rules, including states where populations completely accept or reject products, starting from similar initial states, as well as regimes and cycles of synchronized behaviors of adoption and rejection. It is claimed that without modeling consumer interactions, understanding and modeling of innovation in design will remain inadequate. Since there could be fundamental limitations to the predictability of adoption behaviors under social influence, innovation strategies could fare better when they focus on the quality, novelty, and technological advances they could bring instead of being guided only by social popularity and influence.

Introduction

Traditionally design research has focused on the design processes of the individual designer (Bayazit 2004; Cross 2008; Dym and Brown 2012). This was followed by research into the design behavior of teams of designers, whether collocated or remotely located (Badke-Schaub et al. 2007; Singh et al. 2013; Stempfle and Badke-Schaub 2002). More recently the product's consumer has been brought within the ambit of product design and innovation through feedback of consumer

S. Sarkar (✉)
University of Sydney, Sydney, Australia
e-mail: somwrita.sarkar@sydney.edu.au

J.S. Gero
George Mason University, Arlington, VA, USA

J.S. Gero
University of North Carolina at Charlotte, Charlotte, NC, USA

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_35

behavior and its effect on product design cycles. Information about consumer behavior is becoming increasingly available through their use of social media as they search for and decide to consume products (Forbes and Forbes 2013; Solomon et al. 2014). As a consequence the consumers' behavior is increasingly taken into account as part of an expanded view of the design activity.

Designing, then, ceases to be a linear activity in which an individual or group of producers generate products, deliver it to consumers, who then consume the product. It also ceases to be an *iterative feedback based cyclic activity* in which the feedback from consumers is used to improve or innovate on products for the next cycle of generation. Instead, designing becomes a *networked social* activity, where the dynamics of social communication between the actors (producers and consumers) governs whether or not and how a product is adopted.

It has been shown in live Web based social experiments, for example, that with the quality of the products remaining exactly the same, the presence, absence and degree of social communication between agents could lead to completely different regimes of adoption (Salganik et al. 2006). While product quality did matter, in that while the best products rarely did poorly and the worst ones nearly always suffered, but with a large margin, increasing the strength of social influence amongst consumers made the rankings both more unequal and unpredictable. Thus, predicting exactly how successful a product will be, and how a producer must change their innovation or production strategy becomes a particularly hard problem to solve, given that even chance communication on social networks could change the outcomes dramatically, with no change in the profile of products competing in the market.

Further, social influence may or may not always be positive. For example, the networked structures of technological innovation diffusion that are too deeply influenced by a majority supported social communication could actually end up limiting, hindering, or slowing down innovation (Robertson et al. 2007). Thus, to understand the scenarios under which innovations succeed or fail, it will be important to understand the structure of topological social interactions between consumers, even if one possible emerging policy from such studies is to weight personal values and opinions more than the prevailing social opinion in specific cases in order to meaningfully advance innovation.

As part of a larger scale project that covers the interactions between designers, the interactions between designers and consumers and the interactions between consumers, in this paper we present a preliminary effort to model the diffusion of social influence by consumers over social networks and study how this affects the dynamics of product adoption over time. In particular, under simple and standard local rules of social interactions between consumers, we empirically study whether the topology of the social network has any significant effect on global adoption behaviors for the entire population of consumers.

We study whether under a specific network structure, there is any certainty of a product failing or succeeding, without changing its design quality. We note here that in this preliminary model, we consider design quality and other factors as constant. This allows us to focus on a simple model of social communication and its effect on adoption. Since (real) adoption behaviors are dependent on both design quality as well as social communication, along with host of other factors such as production and distribution strategies, marketing techniques, and demographics, we believe it will be useful to focus on a simple model where all else is held constant. This allows us to specifically focus on the question of social communication in order to understand whether the structure of the network on which social communication occurs has any significant role in the set of emerging behaviors.

This research is part of a larger project that treats designing as an activity that includes the social behavior of both designers and consumers as groups and their interactions within each group and across each group through designed products within its ambit. Social behavior involves both interactions and the consequential change of the values of designers and consumers. It is claimed that without modeling such interactions it is not possible to adequately model innovation. We find that there could be fundamental limitations to the predictability of adoption behaviors under social influence, innovation strategies could fare better when they focus on the quality, novelty, and technological advances they could bring instead of being guided by social popularity and influence. Since it has been observed in the Salganik study that good products rarely do poorly (Salganik et al. 2006), but there is a sufficient level of uncertainty added on account of social influence, the state of the design field could progress faster when innovation or advancement of quality becomes a major focus (reflecting ultimately in the quality of the product, making it one of those good products that would rarely fail).

However, the inherent unpredictability of product adoption leads also to a more counter-intuitive observation: if, holding design quality constant, adoption behaviors are fundamentally unpredictable, then a lazy designer or producer could simply aim to produce many below standard products rather than a few well designed products, in the hopes that at least some of them would succeed purely by contingency. This more counter-intuitive behavior is also empirically observed. For example, in the case of online products such as songs and dances, where people post multiple videos (e.g. on YouTube) that are not of sufficiently high artistic quality, but very large in number. The hope is that at least one of them will become "viral" purely by contingency, making the artist famous, after which even sub-standard work will continue to be adopted. However, we argue, that while such behavior may ensure a short-term advantage to the producer, it may actually hinder real innovation or advancement of the design field.

We model random, small world modular (clustered), and scale free networks of social relationships between consumers, with consumers who could either be *excitatory* (send out positive influence about the product), or *inhibitory* (send out negative influence about the product) for the population of consumers. Each consumer has a *state,* that is, *adopt* or *reject*. This state is a decision for the consumer to make at each time step as socially connected neighbors send out influence messages. Then, we study the dynamics that evolve over several time steps as consumers send out messages to their neighbors. Each consumer receives a positive or negative influence from its neighbors, which helps to inform their state, but the consumer also listens to "itself", that is, its own recommendation. While the current system has no history, the same framework could incorporate history of use as a factor informing the consumer's decision for the next time step in future work.

We report several expected as well as unexpected results on the dynamics, their attractor states, and implications for product adoption behaviors that can emerge out of social communication over networks.

Significance

In its most general form, the problem addressed in this paper has historically been addressed in both biology and the physical sciences (Bornholdt and Rohlf 2000; Gong and Van-Leeuwen 2004), as well as social sciences and economics (Anderson and Holt 1997; Bikhchandani et al. 1992; Friedkin and Johnsen 1999; Kempe et al. 2003). It could be described as the problem of predicting how a system's elements would behave as a collective when local information transfer occurs between the elements. Not all approaches have adopted network based analyses (for example, game theory and information theory have been widely employed to study the problem), and the network based modeling approach shows the fundamental importance that networks of social communication play in deciding consensus emergence, information cascades, or targeting and identification of most influential customers to maximize adoption of innovations and products.

Here we show empirically that while different types of social network structures have different forms of adoption behaviors (modular and clustered networks, for example, could enter into regimes where groups of customers collectively adopt or reject products, and sparse random networks could demonstrate synchronized or cyclic behaviors of adoption and rejection). There is an inherent unpredictability associated with the dynamics even with all network parameters remaining constant. While the model presented here is abstract, and we do not treat the properties of the model analytically, or make any claims or comparisons about the empirical similarity of the model's behavior with real world regimes (which is out of scope for this first preliminary paper), the inherent unpredictability of dynamics suggests an interesting, albeit debatable, finding about innovation strategies by designers. This is to do with the decision on the degree to which social popularity and influence should be used as a guiding basis for driving innovations. If there is inherent unpredictability of dynamics in the system, with system structure, initial conditions, and the parameters remaining constant, an innovation could fare better by focusing on the quality, novelty, and technological advances that it could bring instead of being guided only by social popularity and influence. That is, as discussed above, focusing on design quality lends a higher chance of advancing the design field as well as success of adoption, whereas the strategy of many sub-standard products being introduced in hope of success, could lead to short term gains, but not necessarily advancement of the field.

Method

A network of consumers is defined as a graph $G = (V, E)$, where $|V| = n$ is the number of consumers, and E is a set of edges connecting two consumers i and j from V. The graph G is represented as an $n \times n$ matrix A, where $A_{ij} = 1$ if consumers i and j are connected, and $A_{ij} = 0$ otherwise. The set E could be constructed in several ways, which will affect the resulting social network structure of G. Here we consider three such models: random, small world modular, and scale free connectivity. We note that the construction of the social network could be motivated from several bases, but a common one is patterns of co-usage of a product. That is, a link exists between consumers i and j if they have used common products before.

Random Connectivity Between Consumers

Random connectivity between consumers is modeled as an Erdos-Renyi (ER) random network, where a link exists between each pair of consumers with an independent probability of p. That is $G_r = (V, E_r)$, with A_r representing the adjacency matrix for the random graph. While p can take any value from 0 to 1, we choose p to be low, as most product customer usage networks are very sparse. The lowest values of p could be chosen based on the criteria that a value of $p > \frac{2 \ln n}{n}$ would produce a connected graph. A p lower than this limit would likely produce disconnected components. We will consider values of p above and below this limit.

Under conditions of random connectivity, a consumer is equally likely to receive social messages from any other customer in the population.

Small World Modular Connectivity Between Consumers

Different from random connectivity is the idea of small world modular connectivity. Modular social networks capture the idea that consumers are likely to have tightly clustered social interaction groups, where a group of customers could be using a product and passing recommendations to only a small part of the entire population with large probability, and to the rest of the network with lower probability. This models the situation where people may recommend to close friends or other consumers similar to themselves more than they do to distant friends and acquaintances or other dissimilar consumers in different social groups.

It has been shown that modular networks are likely to be small world (Pan and Sinha 2009). That is, modular networks will have the properties of high clustering coefficients and low average path lengths between nodes. A modular network is modeled using a stochastic block model (Sarkar and Dong 2011; Sarkar et al.

2013a, b). A modular network is modeled as a graph $G_m = (V, E_m)$, where the set of edges E_m represents the links between two nodes from V. The graph is represented by its adjacency matrix A_m, and has a special structure to represent modularity. Let there be q communities of size z nodes each, such that the number of communities $M = n/z$. Thus, each node has a label from the set $\{1, 2, \ldots, q\}$. If two nodes i and j have the same label, then they are connected with a probability of p_{in}, and if they have different labels, they are connected with a probability of p_{out}. Since the probability of connection within a module is higher than that between modules, we have $p_{in} > p_{out}$. Further, we model sparse networks with the total number of edges scaling as $O(n)$, since most real world social networks have sparse connectivity.

In a modular network, each consumer has a higher probability of communicating with members of its own group than members of other groups.

Scale Free Connectivity Between Consumers

Another basis of connectivity between consumers could arise from the idea of influential consumers in the network, that is consumers who are enthusiastic adopters, and therefore use lots of products, and consequently are connected to lots of other consumers through commonality of product use. We model a situation where there are a low number of these influential well connected customers, amongst many more consumers who are not as well connected (i.e., the not so well connected consumers use few products, and therefore are connected to only a few others). Since this network is a mix of a low number of well connected and a high number of not so well connected consumers, it is likely that each well connected consumer will be connected to lots of other consumers themselves who will have low connectivity. The ideal situation in this condition would be a “star” network with one consumer in the middle, connected to and recommending a product all the other consumers in the network.

But, to model a more realistic situation with the above properties, we use the Barabasi-Albert scale free network structure model (Barabasi and Albert 1999). The scale free connectivity network is modeled by a graph $G_s = (V, E_s)$, where E_s represents the set edges between any two consumers from V. In this model, the network is built in successive time steps, where each new incoming node j is connected to an existing node i with a probability p_s proportional to the existing degree of node i. That is, a “rich get richer” model, where nodes with high connectivity keep receiving a high proportion of the connectivity from incoming nodes, resulting in a situation where the network has a low number of highly connected nodes and a high number of nodes with low connectivity, at each stage (hence the name scale free, since recursively removing all the top degree nodes, one would again see the same scale free structure of degree distribution).

A scale free network models the situation where a low number of very influential customers are able to influence and send recommendations to many customers who are "influenced".

A Simple Model of Dynamics of Social Communication

On each of the network types described above, each consumer has a state s, with $s = +1$, if a consumer has adopted a product, $s = -1$, if they reject the product, and $s = 0$, if they are in an undecided state (equally likely to adopt or reject). Further, the current state of each consumer is the social message that is passed on to other neighbor consumers, but all the four following possibilities could exist:

1. Consumer i uses product $(s_i = 1)$ and sends out an excitatory positive recommendation to its neighbor j $(A_{ij} = +1)$.
2. Consumer i uses product $(s_i = 1)$ and sends out an inhibitory negative recommendation to its neighbor j $(A_{ij} = -1)$.
3. Consumer i does not use or rejects product $(s_i = 0)$ or $(s_i = -1)$ but sends out an excitatory positive recommendation to its neighbor j $(A_{ij} = +1)$.
4. Consumer i does not use product $(s_i = 0)$ or $(s_i = -1)$ but sends out an inhibitory negative recommendation to its neighbor j $(A_{ij} = -1)$.

Thus, we have a structure for A where an entire column could be positive or negative (where a link exists), depending on whether the consumer is sending out excitatory (positive) or inhibitory (negative) recommendations to others. The proportion of excitatory and inhibitory customers could then be varied as a parameter to study the dynamics.

The networks are initialized with a $1 \times n$ state vector S, where the n components represent the initial states of each of the n customers, and could be initialized randomly to either +1 or −1 with equal probability. This decision is not trivial and deciding how many and which customers to target in order to maximize influence is a defined problem (Kempe et al. 2003), but we fix this for the purpose of this particular paper since we will look at the dynamics under different types of network structures. Then, based on the local rules above, similar to Bornholdt and Rohlf (2000), Gong and Van-Leeuwen (2004), and a pre-decided proportion of excitatory and inhibitory agents and choosing a particular network connectivity type, the state of each customer at time step $t + 1$ is decided by the inputs it receives from other customers about their states at time t and their recommendations as:

$$S_i(t+1) = \operatorname{sgn}\left(\sum_{j=1}^{n} A_{ij} S_j(t)\right),$$

where the sgn function represents the sign function with $\mathrm{sgn}(c) = 1$ if $c > 0$, $\mathrm{sgn}(c) = 0$ if $c = 0$, and $\mathrm{sgn}(c) = -1$ if $c < 0$. Note that here the agent also receives its own recommendation from the previous time state, since we assume that the self-link A_{ii} is 1 or -1 depending upon whether the customer i is an excitatory or inhibitory customer.

Starting from an initial assignment of S, we let the dynamics run till the whole network stabilizes to a stable point or a limit cycle. We present the results of these runs in the Results section.

Results: Random Connectivity Dynamics

In this section the dynamics of product adoption under random social communication conditions is reported. For the experiments, we varied network sizes n from a few hundred to a few thousand nodes and varied the connection probability p of the random network from below the limit $2 \ln n/n$ (producing disconnected components) to above the limit $2 \ln n/n$ (producing a connected component). When the connection probability is higher than the limit, i.e., the graph is a connected network, the networks settle down to states of either complete adoption or complete rejection by the entire population of consumers. Figure 1a, b shows one state of complete rejection and one of complete adoption, starting from the same initial starting states of product assignment. That is, at time step 1, states S for each customer are assigned, in which each customer is equally likely adopt (+1) or reject (-1) a product. As the model of dynamics of social recommendations discussed in the previous section is allowed to unfold, it is observed that for different runs the entire network either quickly adopts or quickly rejects the product. Once the stable state is reached, it is unchanged for the future (Fig. 1).

When the random network is made sparser and allowed to have several connected components, with probability of connection at or lower than 2 ln n/n, the

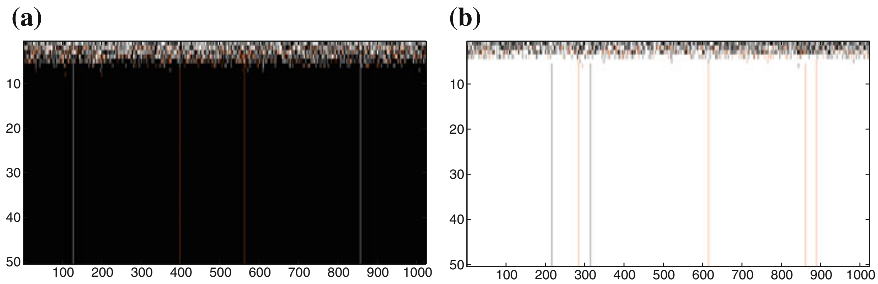


Fig. 1 Connected random social network communication dynamics. **a** $n = 1024, p = 0.014$ (just above the limit $2 \ln n/n$). **b** $n = 1024, p = 0.02$ (above the limit $2 \ln n/n$). The *rows* show the time steps of dynamics evolution, the *columns* show consumer ids. States are represented by *color*, *Black* reject (-1), *white* = adopt (+1), *red* = 0 (undecided)

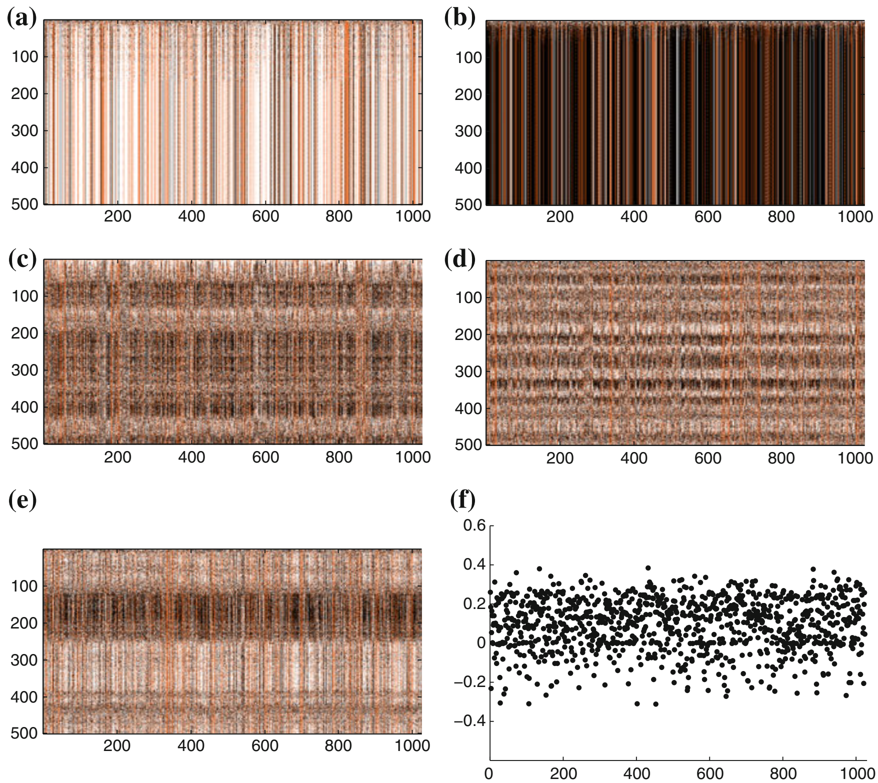


Fig. 2 Disconnected components random social network communication dynamics, $n = 1024, p = 0.003$ (below the limit $2 \ln n/n$). **a** Most consumers adopt the product, **b** most consumers reject the product, **c** bands of synchronized adoption and rejection behaviors through time. The *rows* show the time steps of dynamics evolution, the *columns* show consumer ids. States are represented by *color*, *black* = reject (−1), *white* = adopt (+1), *red* = 0 (undecided). **f** A plot of the average activity of each agent over all the time steps from **e**

results are more unexpected. Figure 2 shows the different regimes of observed behavior: the earlier behaviors of near complete adoption or near complete rejection are observed, but also observed are cases where the stable state shows a mix of both, Figs. 2(a) and (b). Perhaps the most interesting case is observed, shown in Figs. 2(c), (d), and (e), where it appears that several major components of the population (but not all), or the entire network, goes through synchronized phases of adoption and rejection, seen as horizontal bands of black and white in the figures. This behavior shows that it is possible for synchronized behavior to appear in consumers even under random connectivity conditions and simple interaction rules. Further, denser networks show simpler dynamics, whereas sparser networks show richer and more unexpected complex dynamics.

Results: Modular Connectivity Dynamics

In this section the dynamics of product adoption under sparse modular social communication conditions is presented. Figure 3 shows examples of the dynamic behavior that unfolds. Once again, with similar initial state assignments S, two final states of near complete adoption and near complete rejection are observed in a few runs; i.e., a final state of in which either all modules adopt the project or reject the product. However, it is also seen that different modules can reach different stable states, with a few modules in states of adoption co-existing with other modules in states of rejection. Particularly interesting is the behavior of a few individual customers who cycle between the combinations adopt-undecided, adopt-reject, and reject-undecided. Another interesting observation is that the primary intra-module

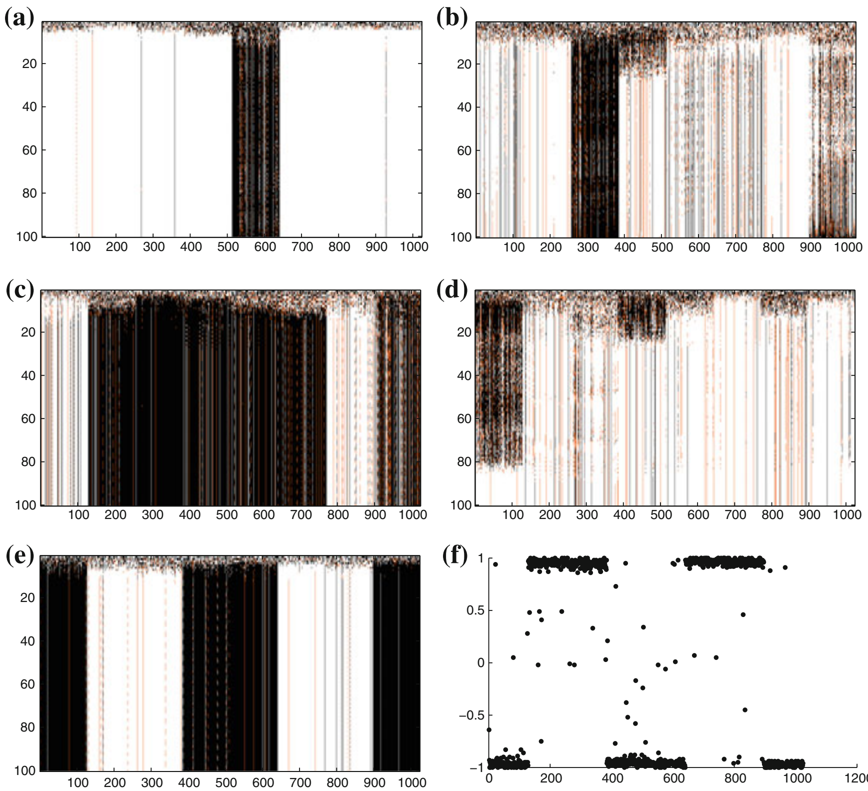


Fig. 3 Modular social network communication dynamics, $n = 1024, p_{in} = 0.1\text{–}0.005$, $p_{out} = 0.005\text{–}0.001$, results from several combinations are shown and discussed in the main text. **a–e** Dynamics of modular adoption and rejection behavior over time, some modules completely reject while some others completely adopt the product. The *rows* show the time steps of dynamics evolution, the *columns* show consumer ids. States are represented by *color*, *black* = reject (−1), *white* = adopt (+1), *red* = 0 (undecided). **f** A plot of the average activity of each agent over all the time steps from **e**

global behavior can continue for long spans of time, but can suddenly switch to another state, Fig. 3d. The mean activity of consumers over 100 time steps in Fig. 3e is shown in Fig. 3f, where the difference between global behaviors in different modules is clearly observed.

The probability of intra and inter modular connectivities, p_{in} and p_{out} have a similar unexpected effect as observed for the random case: more complex behavior is seen for sparser cases. When p_{in} is high, for example, $p_{in} = 0.1, p_{out} = 0.005$ in Fig. 3e, leading to denser networks, the most common outcomes are consumers confirming strongly to the global behavior of their module. Adoption is strong, but so is rejection, and once a state of rejection sets in it is impossible to change it to adoption. However, when p_{in} is low (for example, $p_{in} = 0.05, p_{out} = 0.001$ in Fig. 3a–d), leading to sparser networks, outcomes are more unpredictable, rich and varied. This is unexpected, because it shows richer product adoption dynamics occur when networks are sparser: long spans of rejection could still flip and show regimes of adoption behavior.

Results: Scale Free Connectivity Dynamics

In this section the dynamics of product adoption under scale free social communication conditions is reported. While in the modular case entire modules or groups of consumers show similar behaviors, in the scale free case, the global behavior is more

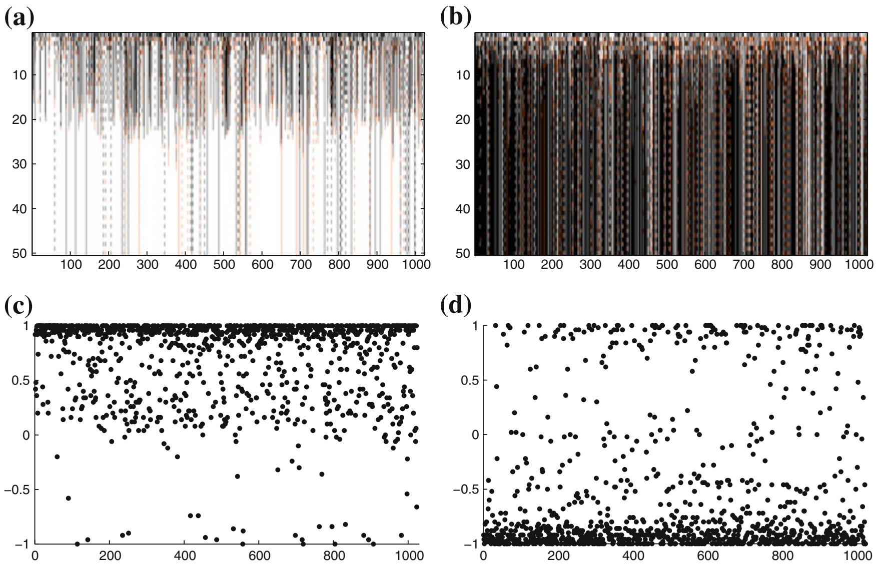


Fig. 4 Scale free social network communication dynamics, n = 1024. Results for scale free networks are shown: **a**, **b**. Dynamics of adoption on scale free networks. The rows show the time steps of dynamic evolution, the columns show consumer ids. States are represented by colour, *black* = reject (-1), *white* = adopt (+1), *red* = undecided (0). **c**, **d**. A plot of average activity of each agent over all the time steps

individual and governed by each consumer's connectivity to highly connected influential consumers.

In this case, an interesting result emerges, where starting from the same initial state assignments *S*, a majority of consumers in the population settle to a stable adoption or rejection state, but a few customers show persistent strong behaviors that are opposite to the main population behavior. Figure 4a, b shows a primarily adopting and primarily rejecting population, but a few individuals (not negligible in number) continue to show the opposite stable behavior. Figure 4c, d shows the corresponding mean activity states for each consumer in (a) and (b) respectively.

Similar to the modular case, a significant number of consumers also show cyclical behaviors, switching between the combinations adopt-undecided, adopt-reject, and reject-undecided. This cyclical behavior emerges as stable, continuing indefinitely. Further, the patterns of the cycles (the numbers of time steps over which one state lasts followed by another) show a rich diversity.

Conclusions

In this paper, we modeled the diffusion of social influence by consumers over social networks and studied how this affects the dynamics of product adoption over time. In particular, under simple and standard local rules of social interactions between consumers, we empirically studied whether the topology of the social network has any significant effect on global adoption behaviors for the entire population of consumers.

Using a simple model of social influence, the dynamics of product adoption are studied over random, modular small world, and scale free social network structures, under local rules of communication. Results show global behaviors emerging from these local agent communication rules, including states where populations completely accept or reject products, starting from similar initial states, as well as regimes and cycles of synchronized behaviors of adoption and rejection. Even with very simple social communication structures, inherently unpredictable complex global behaviors of agents emerged, and the nature of the structure of social communication played an extremely significant role even with the structure or attributes of a designed product remaining unchanged and constant. An analytical assessment of the model was out of scope for this paper, but in future work, we will study both the model the different regimes of results analytically. In future work we will extend the model with data from real world recommendation systems and product adoption records, in order to test the findings and make the abstract model presented here closer to empirical observations of real product adoption and social communication over networks.

In particular, we observe that with unchanged parameters and initial states, different runs produce a rich variety of behaviors, suggesting that any pattern of adoptions and rejections is possible. We also observed that denser networks were rigid and necessitated quick convergence to stable behaviors for entire populations, whereas sparser networks demonstrated much richer diversity of outcomes. Without modeling such interactions, understanding and modeling of innovation in design would remain inadequate. Since there appear to be fundamental limitations to the predictability of adoption behaviors under social influence, in order to advance the design field, innovation strategies could fare better when they focus on the quality, novelty, and technological advances they could bring instead of being heavily guided by fragile collective behavior, social popularity and influence.

Acknowledgements This work is partly supported by a grant from the US National Science Foundation, Grant No. CMMI-1404466 to the second author.

References

Anderson LR, Holt CA (1997) Information cascades in the laboratory. Am Econ Rev 87(5):847–862

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

Barabasi AL, Albert R (1999) Emergence of scaling in random networks. Science 286:509–512

Bayazit N (2004) Investigating design: a review of forty years of design research. Des Issues 20 (1):16–29

Bikhchandani S, Hirshleifer D, Welch I (1992) A theory of fads, fashion, custom, and cultural change as informational cascades. J Polit Econ 100(5):992–1026

Bornholdt S, Rohlf T (2000) Topological evolution of dynamical networks: global criticality from local dynamics. Phys Rev Lett 84(26):6114–6117

Cross N (2008) Engineering design methods: strategies for product design. Wiley, London

Dym C, Brown D (2012) Engineering design: representation and reasoning. Cambridge University Press, Cambridge

Forbes LP, Forbes LP (2013) Does social media influence consumer buying behavior? An investigation of recommendations and purchases. J Bus Econ Res 11(2):107–112

Friedkin NE, Johnsen EC (1999) Social influence networks and opinion change. Adv Group Processes 16:1–29

Gong P, Van-Leeuwen C (2004) Evolution to a small world network with chaotic units. Europhys Lett 67(2):328–333

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: ACM SIGKDD, knowledge discovery and data mining conference. ACM, Washington

Pan RK, Sinha S (2009) Modularity produces small-world networks with dynamical time-scale separation. Europhys Lett 85(6):68006

Robertson M, Swan J, Newell S (2007) The role of networks in the diffusion of technological innovation. J Manage Stud 33(3):333–359

Salganik MJ, Dodds PS, Watts DJ (2006) Experimental study of inequality and unpredictability in an artificial cultural market. Science 311:854–856

Sarkar S, Dong A (2011) Community detection in graphs using singular value decomposition. Phys Rev E 83(4):046114

Sarkar S et al (2013a) Spectral characterization of hierarchical modularity in product architectures. J Mech Des 136(1):011006

Sarkar S, Henderson JA, Robinson PA (2013b) Spectral characterization of hierarchical network modularity and limits of modularity detection. PLoS ONE 8(1):e54383

Singh V, Dong A, Gero JS (2013) Social learning in design teams: the importance of direct and indirect communication. AIEDAM 27(2):167–182

Solomon MR, Dahl DW, White K, Zaichkowsky JL, Polegato R (2014) Consumer behavior: buying, having, and being. Prentice Hall, Upper Saddle River

Stempfle J, Badke-Schaub P (2002) Thinking in design teams—an analysis of team communication. Des Stud 23(5):473–496

Using Graph Complexity Connectivity Method to Predict Information from Design Representations: A Comparative Study

C.V. Sri Ram Mohinder, Amaninder Gill and Joshua D. Summers

Abstract The objective of this research is to compare the value of information in a design representation used in product development. Two representations are explored in this study: assembly models and function structures. These representations are used to predict assembly time and market value for electromechanical products. This works builds on previous work on using complexity connectivity method to predict assembly time. The precision error is used as a metric to understand how valuable a representation is in answering a specific question. By measuring the value of a representation, designers can select between different representations and monitor the information accumulation in the design project.

Aims and Significance

The aim of this research is to demonstrate how to measure complexity of a model and then use the complexity connectivity method as a tool to measure a representation's value. In this case, the design representation refers to function structures and assembly times. By measuring the value of a representation, engineers will be able to select between different representations and monitor the information accumulation in the design project. Researchers, in turn, will be able to develop new representations with greater value to the engineer. To do this, a previously developed method that predicts product performance values based on different product model graphs is used. With this new approach, engineering researchers will be able to systematically compare engineering design representations, such as benchmarking function models (Summers et al. 2013) to determine which representation contains more information through the topological structure of the models.

C.V. Sri Ram Mohinder · A. Gill · J.D. Summers (✉)
Clemson University, Clemson, SC, USA
e-mail: jsummer@clemson.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_36

Moreover, this capability will eventually enable engineers to monitor project health by more fully understanding value of the information that has been accumulated in the design project while also being able to know to what level of confidence they are able to answer different questions about the product performance metrics based on the available design models. This monitoring supports approaches such as the electric analogy based collaborative design model proposed in Ostergaard and Summers (2007).

Proposed Approach

This section lists the proposed research approach to accomplish the aims listed in the previous section. As can be seen in Fig. 1, a function structure or an assembly model of a product is initially created. Once this is done, this assembly model or the function structure (depending on the prediction model) is converted into a bipartite graph. Each of these bipartite graphs is used to generate 29 complexity metrics. These metrics are further fed into the Artificial Neural Network along with armlet values or assembly times (depending on the prediction model). Typically, fifteen products are used to train the neural network and five products are used to test the accuracy. The market values or assembly times obtained are then tested for accuracy in order to understand the value of the prediction model.

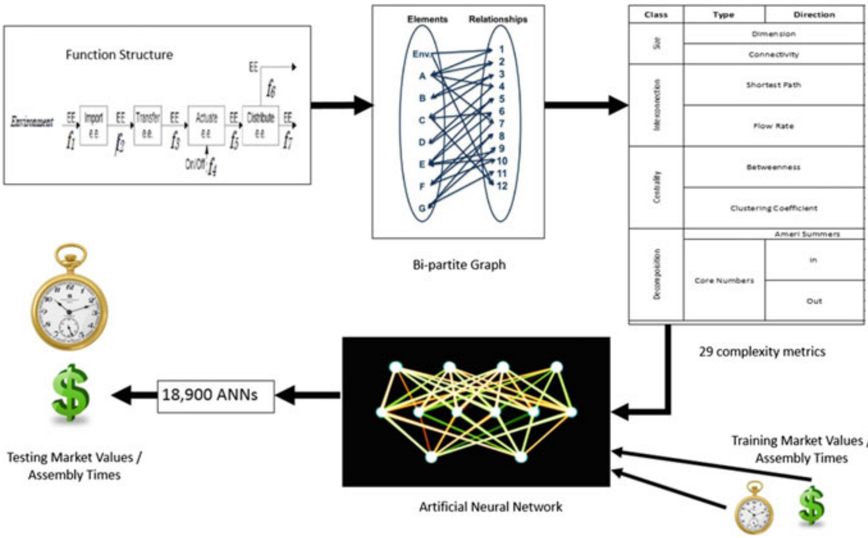


Fig. 1 Schematic representation of research approach

Previous Research on Complexity

Measuring complexity in engineering design is based upon work from different domains and perspectives, including information modeling, software analysis, and traditional manufacturing and design. From an information perspective, Independence and Information Axioms can be used to either reduce or manage the complexity of the design product (Suh 1999, 2001). Similarly, information theory has also been used as a baseline for measuring complexity (El-Haik and Yang 1999; Braha and Maimon 1998a, b). For example, researchers in software development have used complexity measures to determine the "Big-O" difficulty of a problem based on the best possible solution at hand, either implemented or theoretical. Engineers have adapted such complexity measures to model engineering design processes (Ahn and Crawford 1994; Varma and Trachterberg 1990; Phukan et al. 2005; Zuse 1991; Harrison and Magel 1981). Design researchers have long argued that a less complex design is preferable for many reasons (Fitzhorn 1994; Bashir and Thomson 2001; Simon 1998; Dixon et al. 1988; Pahl et al. 2007; Balazs and Brown 2002). For instance, Simon argued that engineering design is related to decomposable systems and that assessing the hierarchical interconnectedness of an engineered artifact enhances the management of such design complexities (Simon 1998). Similarly, we have shown the suitability of complexity measures for predicting assembly times, for elucidating mechanical engineering metrics for DSM (Design Structure Matrix) and representational directional node link systems, and explored how product complexity varies based on representation (Mathieson et al. 2013; Summers and Shah 2010; Mathieson and Summers 2009, 2010; Summers and Ameri 2008; Ameri et al. 2008; Owensby et al. 2012).

Unlike previous research that treats complexity as a single holistic value (Bashir and Thomson 2001; Shah and Runger 2011; Singh et al. 2012; Sinha and de Weck 2013a, b), novel approach of aggregating of different contributing properties is taken here. These metrics include: size, interconnectivity, centrality, and decomposition.

Size is a common measurement used in complexity measurement. The size of an object is based on the count of some classification of the object within the system; as the value increases so too does the complexity (Shah and Runger 2011). While counts are the most intuitive form of complexity measurement, we hasten to note that their contribution to complexity is non-linear (Barclay and Dann 2000). When the count is low, the addition of one more is significant, while the opposite is true of high-count systems. Counts can be modeled using information theory to define a number of bits (Deshmukh et al. 1998).

The size of the system fails to capture the construction of the system, however. Consider a deck of cards with the same deck assembled into a house. While both of these systems are identical in size, in terms of elements and the set of exhaustive possible relationships between these elements, viewed as an *interconnective* structure, the house of cards is clearly more complex. Consequently, the properties of this structure must be considered when evaluating complexity. Path length measurements are based on the number of relationships that must be passed through to travel from one element to another (Mathieson and Summers 2009; Pramanick and Ali 1994). For example, a path length of two from node A to node C is necessary to travel through the system A → B → C. Flow capacity measurements, in turn, are based on the number of unique paths between each pair of nodes. Here, the capacity is determined by the availability of edges, with each edge assumed to have a capacity of one and nodes assumed to have infinite capacity (Goldberg and Tarjan 1986). While shortest-path-length metrics address the existence of connection within the system, flow-capacity metrics elucidate the volume of information that is passed within the system.

Centrality, addressing relative importance of nodes within a system, assumes many forms in network analysis (Freeman 1977, 1979; Koschutzki et al. 2005; Sabidussi 1966). We measure two such forms in our research: betweenness centrality, a measurement on the number of shortest paths on which a node occurs (Freeman 1977); and the clustering coefficient, a measure of the degree to which nodes are grouped within the system (Watts and Strogatz 1998). With respect to an individual node, the clustering coefficient is a measure of the degree to which a given node and its neighbors are to forming a clique, or complete graph. This is defined as the percentage of nodes to which the given node is connected and which are connected to each other.

The final measurement to be explored is that of *decomposability*, the purpose of which is to inventory the requisite steps for structural disassembly of a system. As a measure of complexity, the decomposability score increases with ever larger and more complex systems; thus, what we are measuring is the difficulty of disassembling a system set-by-set. It is this iterative reduction of the system, which we, using the Ameri–Summers decomposability algorithm (Summers and Ameri 2008), seek to measure. Each step consists of removing those relationships that link to the elements with the fewest connections. Each additional step, relationship set, or relationships per separated element required to decompose the system is considered to increase the complexity. In an additional measure of decomposition, core numbers are largest integer such that the given element exists in a graph where all degrees are at least that integer (Bader and Hogue 2003). These degrees are subsequently separated into measurements relating to the in-degree and out-degree of each node in directed graphs.

The attributes derived from graph-based representations are divided into classes of size, interconnection, centrality, and decomposition. Each of these classes contains two measurement subtypes composed of multiple metrics each for a total of 29 dimensions of complexity. While this set is not exhaustive, it is considered sufficient for this application. Previous research has demonstrated that it is possible to use these 29 complexity metrics as a surrogate mapping to design performance measures (Mathieson and Summers 2010; Owensby et al. 2012; Mathieson et al. 2013; Namouz and Summers 2014). These complexity metrics have been listed in Table 1.

Table 1 Complexity metrics

Class	Type	Metrics
Size	Dim	Elements
		Rel
	Conn	DOF
		Conn
Interconnection	Shortest path	Sum
		Max
		Mean
		Density
	Flow rate	Sum
		Max
		Mean
		Density
Centrality	Betweenness	Sum
		Max
		Mean
		Density
	Clustering coefficient	Sum
		Max
		Mean
		Density
Decomposition	Ameri summers	
	Core numbers	
	In	Sum
		Max
		Mean
		Density
	Out	Sum
		Max
		Mean
		Density

Generating Complexity Metrics Using Assembly Models

Three dimensional solid assembly models generated in Solid Works software are used as graph sources. These assemblies are converted into connectivity graphs using a SolidWorks customized package (Namouz and Summers 2014). The connectivity graph method uses class detection methods to determine which parts in the assembly touch which other parts. Detailed information about this tool box is presented in Simon (1998). An assumption that is being tested in this research is whether the performance metrics (Assembly time and Market value) of the products are dependent on the amount of information contained within the topological structure of the assembly models. The specific details of the parts such as size and symmetry are not included in the graphs. The assemblies of the CAD model are used as inputs to graphs to predict the performance values.

Generating Complexity Metrics Using Function Structures

A function structure is a graph-based model of mechanical product functionality, whose nodes are transformative actions and edges are flows undergoing transformations through a design product (Varma and Trachterberg 1990). These graphs are a useful representation to support early design activities, such as problem decomposition and understanding, solution search (Phukan et al. 2005), concept generation (Braha and Maimon 1998b; Zuse 1991) and design archival (Harrison and Magel 1981). These function structures are either generated manually or with aid of the concept modeler tool (Bashir and Thomson 2001). The conversion of a function model into a bipartite graph begins with labeling the flows (energy, material, and signal) as (f1, f2, f3, f4, f5, f6, f7). Figure 2 shows a generic schematic of a function structure with flows and functions. This serves as a reference document for converting the function structure representation to bi-partite graphs (Suh 1999). Each function block in the model can then be assigned an element ID and mapped with the corresponding elements. Thus, a bi-partite graph can be created manually from function structure and can thereafter be used to generate the 29 complexity metrics.

Bi-partite Graphs

Bi-partite graphs consist of two independent sets where the connections are drawn between these two sets. The first independent set is the system elements, physical parts, or function blocks. The second independent set consists of relationships that might include contacts, connections, or flows within function models. These graphs are further converted into 29 complexity metrics vector. Generation of these graphs from the two design representations are discussed in the following sections.

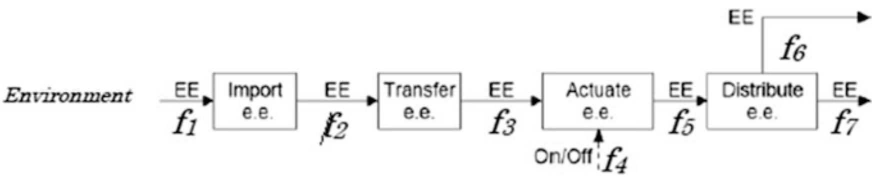


Fig. 2 Schematic of function structure

Artificial Neural Networks

The 29 complexity metrics vector of the products to the performance targets of the prediction models. The Artificial Neural Networks (ANN) is chosen to explore these relationships due to their ability to perform nonlinear statistical modeling. Some of the advantages of using these networks: (1) ANNs require less formal statistical training, (2) ANNs' ability to detect complex nonlinear relationships between independent and dependent variables, (3) the ability to discover all possible interactions between predictor variables, and (4) the ability to use multiple training algorithms (Zhang et al. 1998; Sharda and Patil 1992; Tu 1996; Shtub and Versano 1999; Kohzadi et al. 1996; Perzyk et al. 2005).

The vectors of metrics generated by the complexity analysis and the mean price quote and assembly time of each product are used as inputs and targets respectively for neural network training. The ANN then creates a relationship between the input values and the target values. A back propagation neural network is technically capable of fitting to any function given a sufficient number of neurons and training cycles (Summers et al. 2014). Thus, the networks we use consists of three layers with multiple neurons in each layer. This results in a total of 189 different architectures generated. To address the initial randomized weighting of the networks, each architecture is replicated 100 times. In this manner, the approach to ANN training is flipped from large training sets to train a single surrogate model to using few training samples to create large number of ANN surrogate models (18,900). Thereby, small data is made large for predictive purposes.

Experimental Procedure

This section briefly describes the sequence of steps (Fig. 3) involved in quantifying and comparing the information value in the two different design representations. First, the design representation for the corresponding product is chosen. Next, the representation specific bi-partite graph is generated for product. This bi-partite graph is used to construct a vector consisting of the 29 complexity metrics. This is done for twenty products. Fifteen products are used to train the ANNs and the other five are used to test them. These test are done for assembly times and market values. The results obtained from these prediction models are then analyzed to understand the information value for the design representations.

The four prediction models will used in this study are: (a) Assembly Models—Assembly Times (AM-AT) (b) Assembly Models—Market Values (AM-MV) (c) Function Structure—Assembly Times (FS-AT) and (d) Function Structures—Market Values (FS-MV).

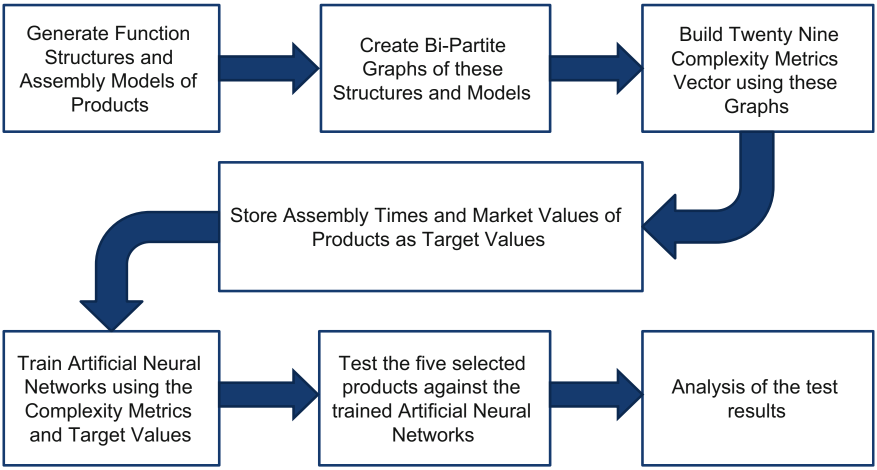


Fig. 3 Schematic of the experimental procedure

Data Used for the Experiments

The products selected for the experiment are consumer electro-mechanical products consisting of power tools and kitchen appliances. Readily available products are selected for this study. This was done to ensure that their CAD (Computer Aided Design) models were easily available. The assembly models of the products are taken from *GRAB CAD* online web store and *3D CONTENT CENTRAL* web site where as the function structures of sixteen products have been taken from *Oregon State Design Repository*. Wherever these models were not available, they were created manually by reverse engineering the products. This approach reduces the variance in the test results of the products. This is because the source of the products is accessible to all users and hence the experiment produce consistent test results for any number of repetitions.

The market values for each of the products were queried from *Amazon.com* web site. Five base-price quotes are taken spanning the price range for each product. The market values used are shown in the Table 2 indicating the market value of the products. For the purpose of this research, an average value of these five quotes is selected. These quotes of the products are used as the target values of the products for the two graph types. These values are compared against the test results of the experiment in predicting the accuracy of prediction for the four prediction models.

The assembly time of the selected products is computed manually using the *Boothroyd and Dewhurst* tables for Design for Assembly (DFA) (Bashir and Thomson 2001). Table 3 displays the assembly time of the products. The assembly time of the products computed manually is used as target values of the products for the two graph types. These values are compared against the test results of the experiments in predicting the accuracy of the four prediction models. The products bolded in Tables 2 and 3 are used as the test products.

Table 2 Market values for the various consumer electro-mechanical products

S. No.	Product name	Quote 1 (USD)	Quote 2 (USD)	Quote 3 (USD)	Quote 4 (USD)	Quote 5 (USD)	MEAN (USD)
1	Stapler	24.88	17.67	14.69	16.13	16.83	18.04
2	**Flashlight**	**17.89**	**17.76**	**20.38**	**20.65**	**24.92**	**20.32**
3	Blender	14.96	19.99	21.99	24.85	25.31	21.42
4	Electric grill	47.02	49.91	58.94	79.95	89.99	65.16
5	Solar yard light	1.66	1.94	3.00	3.75	4.12	2.89
6	Bench vise	38.38	39.15	40.71	40.72	43.37	40.47
7	Electric drill	42.99	48.42	49.97	59.26	69.46	54.02
8	**Food chopper**	**39.95**	**42.99**	**49**	**49**	**59**	**47.99**
9	Computer mouse	6.95	8.17	8.99	9	12.01	9.02
10	3-Hole punch	57.91	62.99	63.83	71.56	73.5	65.96
11	Electric tooth brush	79.99	95.99	96.9	119	129.95	104.37
12	Garage door opener	103.99	119.88	128	139	148	127.77
13	Juicer extractor	26.99	29.95	30.19	32.78	40	31.98
14	Mixer	8.99	9.89	13.22	14.96	19.99	13.41
15	**Hair dryer**	**14.99**	**20.96**	**23.99**	**24.49**	**26.95**	**22.28**
16	Nail gun	69.00	76.96	79.99	82.99	89.68	79.72
17	**Lawn mower**	**99.99**	**114.99**	**135.99**	**137.97**	**143.99**	**126.59**
18	Jigsaw	114.99	117.5	78.99	74.999	139.95	105.29
19	**Sander**	**169.95**	**189.9**	**204.97**	**214.95**	**295**	**214.95**
20	Sewing machine	75.00	125.00	175.00	129.00	69.99	114.80

Analysis of the Test Results

This section discusses the statistical analysis carried out on the test results obtained. The accuracy of the four prediction models can be estimated by computing the error in predicting the result where the error is measured as a difference between the test result and test target for all the 18,900 test result values of each prediction model. The percentage error of prediction is computed as the difference in predicting the test result divided by the test target (*Error* = (*Result* − *Target*)/*Target*).

Table 3 Assembly times computed using Boothroyd and Dewhurst DFA tables for the various consumer electro-mechanical products

No.	Product name	Assembly time (s)
1	Stapler	123.51
2	**Flashlight**	**75.4**
3	Blender	263.21
4	Electric grill	121.08
5	Solar yard light	128.79
6	Bench vise	143.69
7	Electric drill	189.65
8	**Food chopper**	**316.62**
9	Computer mouse	81.25
10	3-Hole punch	145.38
11	Electric tooth brush	395.821
12	Garage door opener	196.5
13	Jigsaw	339.38
14	Mixer	76.65
15	**Hair dryer**	**89.53**
16	Nail gun	90.44
17	**Lawn mower**	**296.61**
18	Juice extractor	76.65
19	**Sander**	**218.18**
20	Sewing machine	273.71

Results

In this section, the percentage error for each of the five test products used across all four prediction models will be calculated and represented using histograms.

AM-AT Prediction Model

Figure 4 shows a histogram graph displaying the accuracy (Percentage error) in predicting the assembly time of five products using assembly models. The accuracy in predicting the assembly time of this products varies from (−7.5 to 10.2%). The 'x-axis' in the graph represents the percentage error in the assembly time of the products and the assembly time of these five test products are portrayed. The 'y-axis' represents the frequency of predicting the assembly time of the products and a population size of 18,900 performance values has been used to test the performance values of each product. This AM-AT prediction model shows that assembly models of the products predict the assembly time of the products with good accuracy.

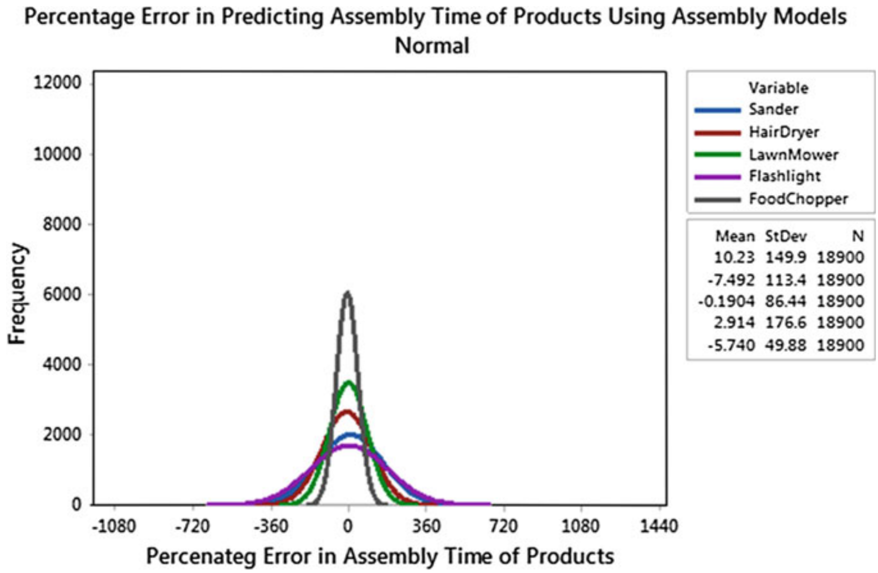


Fig. 4 Percentage error in predicting assembly time of products using assembly models

AM-MV Prediction Model

Figure 5 displays the accuracy (Percentage error) in predicting the market value of five products assembly models using a histogram. The accuracy in predicting the market value of this products ranges from (−7.2 to 23.2%). The 'x-axis' in the graph represents the percentage error in the market value of the products and these market costs of the test products. The 'y-axis' represents the frequency of predicting the market value of the products and a population size of 18,900 performance values has been used to test the performance values of each product. This AM-MV prediction model shows that assembly models of the products predict with lower accuracy as compared to the AM-AT prediction model.

FS-AT Prediction Model

Figure 6 depicts a histogram graph displaying the accuracy (Percentage error) in predicting the assembly time of five products function structures. The accuracy in predicting the assembly time of this products vary from (−59.9 to 19.0%). The 'x-axis' in the graph represents the percentage error in the assembly time of the products and the assembly time of these five test products are represented in Fig. 5.

The 'y-axis' represents the frequency of predicting the assembly time of the products for a population size of 18,900 performance values that have been used to

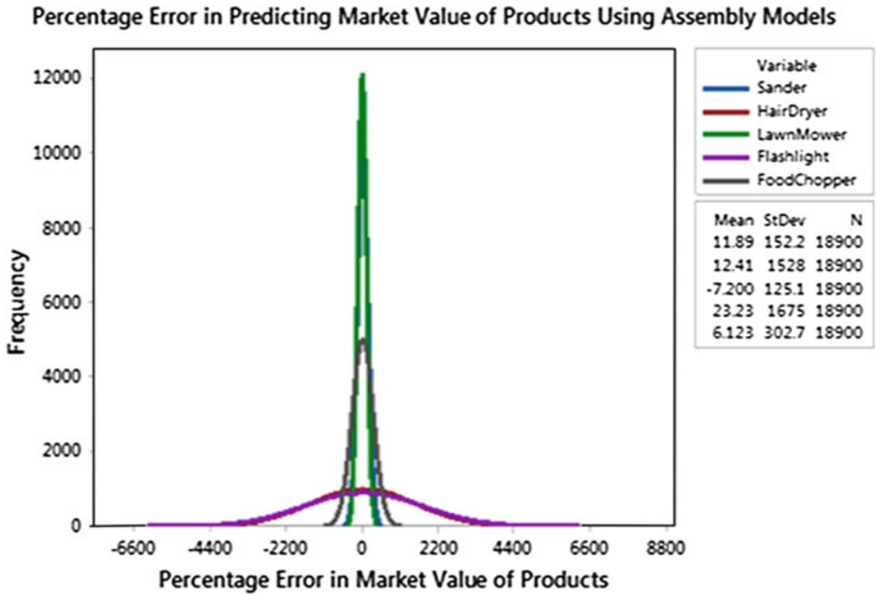


Fig. 5 Percentage error in predicting market value of products using assembly models

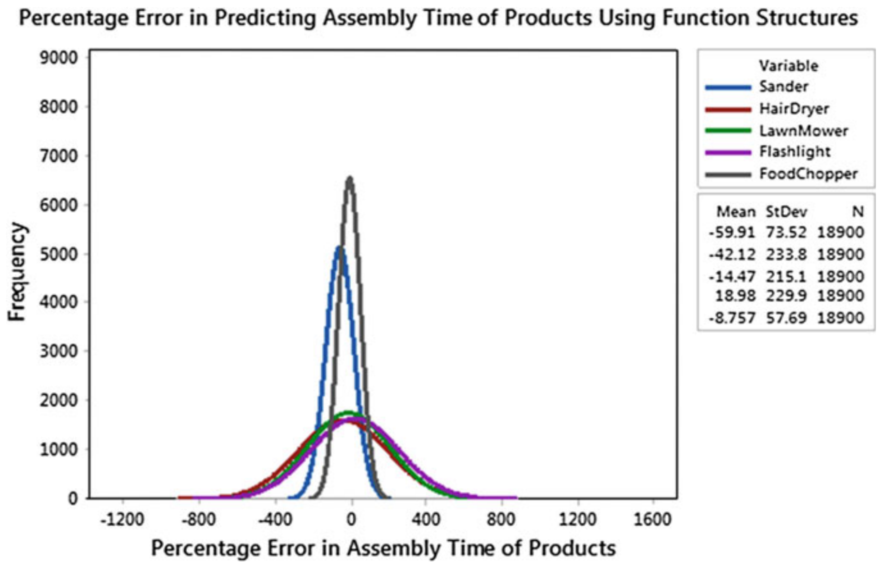


Fig. 6 Percentage error in predicting assembly time of products using function structures

test the performance values of each product. This FS-AT prediction model shows that function structures of the products predict with lower accuracy as compared to both AM-AT and AM-MV prediction models.

FS-MV Prediction Model

Figure 7 portrays a histogram graph displaying the accuracy (Percentage error) in predicting the market value of five products function structures. The accuracy in predicting the market value of this products ranges from (−59.3 to 153.9%). The 'x-axis' in the graph represents the percentage error in the market value of the products and these market costs of the test products. The 'y-axis' represents the frequency of predicting the market value of the products and a population size of 18,900 performance values has been used to test the performance values of each product. This FS-MV prediction model shows that function structures of the products predict with lowest accuracy in predicting the market value of the products when compared to other three prediction models (AM-AT, AM-MV and FS-AT).

Table 4 summarizes the absolute average percentage error of the five test products (Sander, Hair Dryer, Lawn Mower, Flashlight and Food Chopper) for the four prediction models. Table indicates the ranking order of the four prediction models based on the absolute average percentage error in predicting the performance of the five selected products.

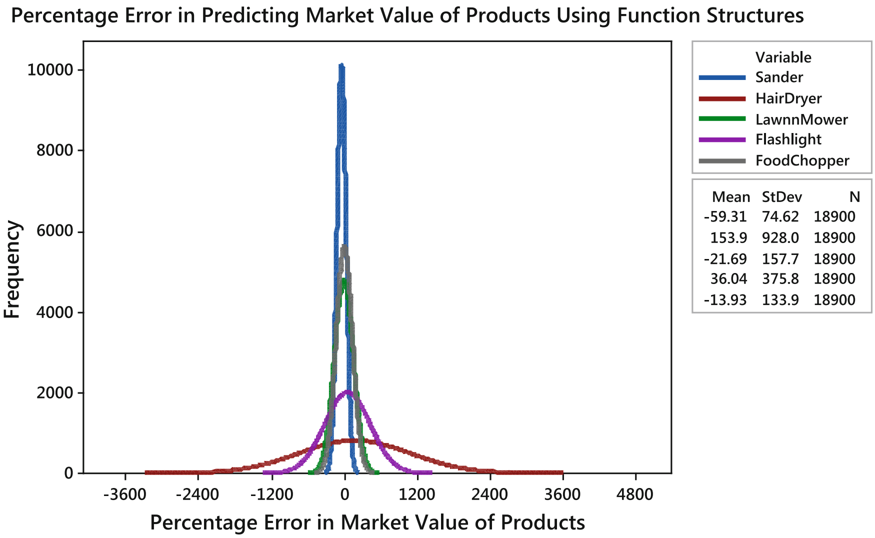


Fig. 7 Percentage error in predicting assembly time of products using market values

Table 4 Average percentage error of the four prediction models

Average Percentage Error of the Four Prediction Models				
	AM-AT	AM-MV	FS-AT	FS-MV
SANDER	10.2%	11.9%	-60.0%	-59.3%
HAIR DRYER	-7.5%	12.4%	-42.2	153.9%
LAWN MOWER	-0.2%	-7.2%	-14.5%	-21.7%
FLASH LIGHT	2.9%	23.2%	19.0%	36.0%
FOOD CHOPPER	-5.7%	6.1%	-8.6%	-13.9%
ABSOLUTE AVERAGE	5.3%	12.2%	28.8%	57.0%

The color coding scheme for the average percentage error representation is as follows: Percentage Error (< 10%), Percentage Error (10%-20%), Percentage Error (20%-40%), Percentage Error (40%-60%), and Percentage Error (>100%).

Table 5 A comparison of the four prediction models

	Assembly Time	**Market Value**
Assembly Models	AM-AT Rank 1	AM-MV Rank 2
Function Structures	FS-AT Rank 3	FS-MV Rank 4

Conclusions and Future Work

A comparison of the four prediction models based on the accuracy (percentage error) in predicting the performance metrics of the five consumer products has been computed and portrayed in Table 5. It can be observed that the assembly models predict the assembly time of the products with an absolute average accuracy of 5.3% and hence are ranked first. Similarly the assembly models predict the market value of the products at an average accuracy of 12.2% and hence ranked second in accurately predicting the performance of the products. The function structures predict the assembly time of the products at average percentage error of 28.8% and the function structures predict the market value of the at an average accuracy of 57.0%. Hence, it can be concluded that assembly models and function structures in early design stages can be used to predict the market value and assembly time of the products at an early design stage within reasonable bounds of error. Hence, this aids in producing qualitative concepts which can be used further in the embodiment design stage and informs designers early in the design process. This approach therefore helps in generating multiple different historically based predictors that can enable designers to make better informed decisions early on in the process.

The future extensions of this research would be to understand and identify key complexity metrics of different product model graphs (assembly models and function structures) that can be used for surrogate modeling of product performance metrics (assembly time and market cost). Future efforts need to be directed towards understanding why these surrogate models would work. It can be hypothesized that some complexity metrics are significant predictors for both market price and assembly even when applied against disparate models such as the function structures and CAD assembly models. Once the prediction models are created through the use of artificial neural networks (ANN), the level of significant contribution of each metric to the prediction model can be analyzed. Both principle component analysis and linear and nonlinear regression analysis to refine the complexity metric vector can be explored. Also, in order to study if one can extend this approach for other performance metrics like recyclability.

References

Ahn J, Crawford R (1994) Complexity analysis of computational engineering design. In: ASME international design engineering technical conferences and computers and information in engineering conference, ASME, Minneapolis, MN, pp 205–220

Ameri F, Summers JD, Mocko GM, Porter M (2008) Engineering design complexity: an investigation of methods and measures. Res Eng Des 19(2):161–179

Bader GD, and Hogue CWV (2003) An automated method for finding molecular complexes in large interaction network. BMC Bioinf 4

Balazs M, Brown D (2002) Design simplification by analogical reasoning. From knowledge intensive CAD to knowledge intensive engineering. Kluwer, Norwell, pp 29–44

Barclay I, Dann Z (2000) New-product-development performance evaluation: a product-complexity-based methodology. IEE Proc: Sci Meas Technol 147(3):41–55

Bashir HA, Thomson V (2001) An analogy-based model for estimating design effort. Des Stud 22:157–167

Braha D, Maimon O (1998a) The measurement of a design structural and functional complexity. IEEE Trans Syst Man Cybern (Part A Syst. Humans) 28(4):527–535

Braha D, Maimon O (1998b) A mathematical theory of design: foundations, algorithms, and applications. Kluwer, Dordrecht

Deshmukh A, Talavage J, Barash M (1998) Complexity in manufacturing systems—part 1: analysis of static complexity. IIE Trans 30:645

Dixon J, Duffey M, Irani R, Meurier K, Orelup M (1988) A proposed taxonomy of mechanical design problems. In: Computers in engineering conference, ASME, New York, pp 41–46

El-Haik B, Yang K (1999) The components of complexity in engineering design. IIE Trans 31:925–934

Fitzhorn P (1994) Engineering design as a computable function. Artif Intell Eng Des Anal Manuf 8:35–44

Freeman L (1977) A set of measures of centrality based on betweenness. Soc Netw 40:35–41

Freeman L (1979) Centrality in social networks: conectual clarification. Soc Netw 1:215–239

Goldberg AV, Tarjan RE (1986) A new approach to the maximum flow problem. In: Annual ACM symposium theory on computational, New York, pp 136–146

Harrison W, Magel K (1981) A complexity measure based on nesting level. SIGPLAN Not. 16:63–74

Kohzadi N, Boyd MS, Kermanshahi B, Kaastra I, Khozadi N (1996) A comparison of artificial neural networks and time series models for forecasting commodity prices. Neurocomputing 10 (2):161–181

Koschutzki D, Lehmann KA, Peeters L, Richer S, Tenfelde-Podehl D, Zlotowski (2005) Centrality indices. In: Brandes U, Erlebach T (eds) Network analysis: methodological foundations. Springer, New York

Mathieson JL, Summers JD (2009) Relational DSMs in connectivity complexity measurement. In Proceedings of the 11th international DSM conference, pp 15–26

Mathieson JL, Summers JD (2010) Complexity metrics for directional node-link system representations: theory and applications. In: ASME international design engineering technical conferences and computers and information in engineering conference, ASME, Montreal, Canada, pp DETC2010–28561

Mathieson JL, Wallace BA, Summers JD (2013) Assembly time modelling through connective complexity metrics. Int J Comput Integr Manuf 26(10):955–967. doi:10.1080/0951192X.2012.684706

Namouz E, Summers JD (2014) Comparison of graph generation methods for structural complexity based assembly time estimation. ASME Trans J Comput Inf Sci Eng 14(2):021003

Ostergaard KJ, Summers JD (2007) Resistance based modeling of collaborative design. Concurr Eng 15(1):21–32

Owensby JE, Namouz EZ, Shanthakumar A, Summers JD (2012) Representation: extracting mate complexity from assembly models to automatically predict assembly times. In: ASME international design engineering technical conferences and computers and information in engineering conference, ASME, Chicago, pp DETC2012–70995

Pahl G, Beitz W, Wallace K, Blessing L (2007) Engineering design: a systematic approach. Springer London Limited, London

Perzyk M, Biernacki R, Kochański A (2005) Modeling of manufacturing processes by learning systems: the naïve Bayesian classifier versus artificial neural networks. J Mater Process Technol 164:1430–1435

Phukan A, Kalava M, Prabhu V (2005) Complexity metrics for manufacturing control architecture based on software and information flow. Comput Ind Eng 49:1–20

Pramanick I, Ali H (1994) Analysis and experiments for a parallel solution to the all pairs shortest path problem. In: IEEE international symposium on circuits systems, New York

Sabidussi G (1966) The Centrality Index of a Graph. Psychometrika 31:581–603

Shah JJ, Runger G (2011) Misuse of information-theoretic dispersion measures as design complexity metrics. In: ASME international design engineering technical conferences and computers and information in engineering conference, ASME, Washington, pp DETC2011/DTM–48295

Sharda R, Patil RB (1992) Connectionist approach to time series prediction: an empirical test. J Intell Manuf 3:317–323

Shtub A, Versano R (1999) Estimating the cost of steel pipe bending: a comparison between neural networks and regression analysis. Int J Prod Econ 62:201–207

Simon H (1998) The sciences of the artificial. MIT Press, Cambridge

Singh G, Balaji S, Shah JJ, Corman D, Howard R, Mattikalli R, Stuart D (2012) Evaluation of network measures as complexity metrics. In: ASME international design engineering technical conferences and computers and information in engineering conference, ASME, Chicago, pp DETC2012–70483

Sinha K, de Weck OL (2013a) A network-based structural complexity metric for engineered complex systems. In: 2013 IEEE international systems conference (SysCon), pp 426–430

Sinha K, de Weck OL (2013b) Structural complexity quantification for engineered complex systems and implications on system architecture and design. In: ASME 2013 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, pp V03AT03A044–V03AT03A044

Suh N (1999) A theory of complexity, periodicity, and the design axioms. Res Eng Des 11:116–131

Suh N (2001) Axiomatic design: advances and applications. Oxford University Press, New York
Summers JD, Ameri F (2008) An algorithm for assessing design complexity through a connectivity view. In: Proceedings of the TMCE 2008
Summers JD, Shah JJ (2010) Mechanical engineering design complexity metrics: size, coupling, and solvability. J Mech Des 132(2):021004
Summers JD, Eckert C, Goel AK (2013) Function in engineering: benchmarking representations and models. In: International conference on engineering design. The Design Society, Seoul
Summers JD, Miller MG, Mathieson JL, Mocko GM, Summers JD, Mathieson JL, Mocko GM (2014) Manufacturing assembly time estimation using structural complexity metric trained artificial neural networks. J Comput Inf Sci Eng 14(1):11005
Tu JV (1996) Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. J Clin Epidemiol 49(11):1225–1231
Varma D, Trachterberg E (1990) On the estimation of logic complexity for design automation applications. In: International conference on computer design: VLSI in computers and processors, Cambridge
Watts DJ, Strogatz S (1998) Collective dynamics of 'Small-World' Networks. Nature 393(6):440–442
Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural network: the state of the art. Int J Forecast 14(1):35–62
Zuse H (1991) Software complexity: measures and methods. Walter de Gruyter and Co., New York

Identifying Sentiment-Dependent Product Features from Online Reviews

Dedy Suryadi and Harrison M. Kim

Abstract This paper presents a method to correlate relevant product features to the sales rank data. Instead of going through the labor-intensive surveys, online product reviews have become an efficient source to gather consumer preferences. The contribution of the paper is to relate the content of reviews to a product's sales rank that implicitly reflects the motivation behind what drives customers to purchase the product. After using part-of-speech tagging to extract the relevant feature and opinion pairs from the reviews, the extracted data along with the review ratings and price become the variables to explain the sales rank. An experiment is run for wearable technology products to illustrate the methodology and interpret the result. The result indicates that the positive opinion for battery and negative opinion for sleep tracker are significant towards sales rank, while price is not.

Introduction

Products are mainly designed to fulfil customers' needs. Therefore, it is natural to start the product design process by identifying customer needs. By designing product features to the needs appropriately, it is expected that customers are interested and decide to purchase the product.

The voice of customers can be obtained from those who have had experience with a product or a similar one. In particular, it is valuable for product designers to find out what features are appreciated by customers and what features bring disappointment to customers, such that the next generation can be designed better.

Collecting the voice of customers may require a significant amount of effort and money, if it is done by traditional method such as surveys. An alternative way to collect preference data is through publicly available product reviews. Online product reviews refer to the free texts written by consumers, rather than domain experts, which are published in e-commerce websites, such as Amazon.com

D. Suryadi · H.M. Kim (✉)
University of Illinois at Urbana-Champaign, Urbana, IL, USA
e-mail: hmkim@illinois.edu

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0_37

(Liu et al. 2013). In those reviews, customers voluntarily express their opinions about a product, and therefore high authenticity of the content can be expected (Decker and Trusov 2010).

The amount of online product reviews has grown rapidly on the internet, and they have a significant impact on e-commerce (Forman et al. 2008). Online reviews also become an important and more attractive source of information (word-of-mouth) for online shoppers than vendor-released product description (Bickart and Schindler in Archak et al. 2011). According to a report (Sun 2012), 68% of online shoppers check at least four reviews before buying, and nearly 25% of the shoppers check at least eight reviews. It supports the belief that customers often seek others' opinions before making a decision to purchase a product. More interestingly, it is discovered that "the online ratings on a website are significantly correlated with online purchases." (Archak et al. 2011).

This paper presents a method to link product sales rank with the sentiments from the online product reviews to find how the content of the product review affects its sales (rank) which will in turn lead to better product design by reflecting the sentiments. In particular, the products observed are wearable technology products, which are relatively new in the market as it is expected that customers are seeking and sharing more information about new products via online reviews. The rest of the paper is organized as follows: Literature Review summarizes the relevant previous researches, Methodology describes the flow of the proposed method, Data Processing explains the steps to process the sales rank, price, and review data, Regression & Result shows the regression model and the result, followed by Discussions of the results and the Conclusion of the paper.

Literature Review

In understanding online product reviews, there have been previous approaches that showed interesting results. For example, Decker and Trusov identified the effect of reviews on the rating given by the reviewer (Decker and Trusov 2010). However, in their research, the collected review data from the website had been divided into pros and cons. Therefore, it was already clear that the features mentioned in the pros section are deemed positive by the user, and vice versa. The methods applied are Poisson and Negative Binomial regressions. Among the findings, an interesting insight is that when 'price' appears in the pros section, the effect to sales rank is slightly negative. The reason is that most of the time when 'price' appears in the pros section, it comes from a cheap product. A cheap product, however, must not necessarily imply high satisfaction, particularly if the product sacrifices its quality and reliability. Overall, it is concluded that Negative Binomial regression approach can be a promising compromise between simplicity of application and the need of adequately considering opinion heterogeneity. However, the disadvantage of this approach is that using rating as the target variable does not seem very useful. After all, it is the sales that matter.

Tucker and Kim (2011) also utilized the review data from cnet.com to reveal customer preference, in which the reviews have been divided into pros and cons sections. Using the comments from the pros and cons sections, the approach is to predict emerging features and obsolete features. It is argued that emerging product features may be popular in future product design models, while obsolete ones should be excluded from future product designs. This approach, again, does not directly link the feature and opinions to the most important metric to measure a product's success, i.e. sales.

Other research observes the effect of online reviews to the sales rank for books (Chevalier and Mayzlin 2006). However, the dependent variables used are only the numerical aspects of the reviews, such as rating (the number of stars), the review length, number of reviews, etc. An interesting finding is that the fraction of 5-star reviews (i.e., ratio of 5-star reviews to the total number of reviews) affects sales negatively. It is argued that the credibility of the reviews become doubtful as the fraction gets too large. Another finding is that for the case of books, customers appear to read the reviews, as proven by the significance of review length. This strongly suggests that the content and characteristics of the reviews should not be ignored.

Archak et al. (2011) go deeper into the review content in their approach. The target variable used is sales rank, the dependent variables are both numeric and text. The research confirms that not only the numeric aspects matter, but the textual content matters as well. As an example, for digital camera, it is shown that a review containing 'good picture quality' affects the sales rank negatively. It is explained that customers may have higher expectation than simply 'good' picture quality for a camera.

An extensive list of dependent variables is created by Chong et al. (2015) but variables related to textual content of product reviews are excluded. Those dependent variables are the availability of free delivery, discount value (in monetary value), discount rate (in percentage), customer review rating, number of customer reviews, number of answered questions, fraction of positive reviews (percentage of 5 and 4-star customer reviews), fraction of negative reviews (percentage of 2 and 1-star customer reviews), rating of the most helpful review, number of people who found the most helpful review useful, and total number of people that gave feedback (either found the review useful or not) to the most helpful review. Those variables are used to explain the sales rank of products in Amazon.com. As indicated by related research (Chevalier and Mayzlin 2006; Archak et al. 2011), textual content of product reviews contain valuable information to explain the movement of sales rank.

As for the feature and opinion extraction from product reviews, previous research proposes a method that starts with Part-of-Speech tagging (Hu and Liu 2004a, b). It is followed by applying association rule mining to the nouns and noun phrases to select feature candidates. The candidates are further pruned based on compactness and redundancy rule.

The opinion are extracted from the adjectives and they are linked to the nearest noun or noun phrase, then the sentiment for each opinion word is obtained by utilizing WordNet (Hu and Liu 2004a, b). A related research (Liu et al. 2005) applied association mining to mine pattern rules in order to capture product features. An example of the rule obtained is “easy, to, <V> [feature]”, which <V> indicates a verb and [feature] indicates a feature to extract. Nevertheless, they do not relate the feature-opinion with sales rank.

The proposed method intends to find the relationship between the textual content of product reviews and product sales rank. The product example is a relatively new type of products, in contrast to the ones used in previous researches, such as books, CDs, cameras, etc. Furthermore, although all products in the example are grouped as wearable technology products by Amazon, their sales are ranked in different categories. Therefore, a method should be created to make fair comparison across various categories. The details of the methodology follows in the next section.

Methodology

The methodology of the research is summarized on the graph in Fig. 1. It comprises three parts, i.e. pre-processing price and sales rank, processing product reviews, and the regression model.

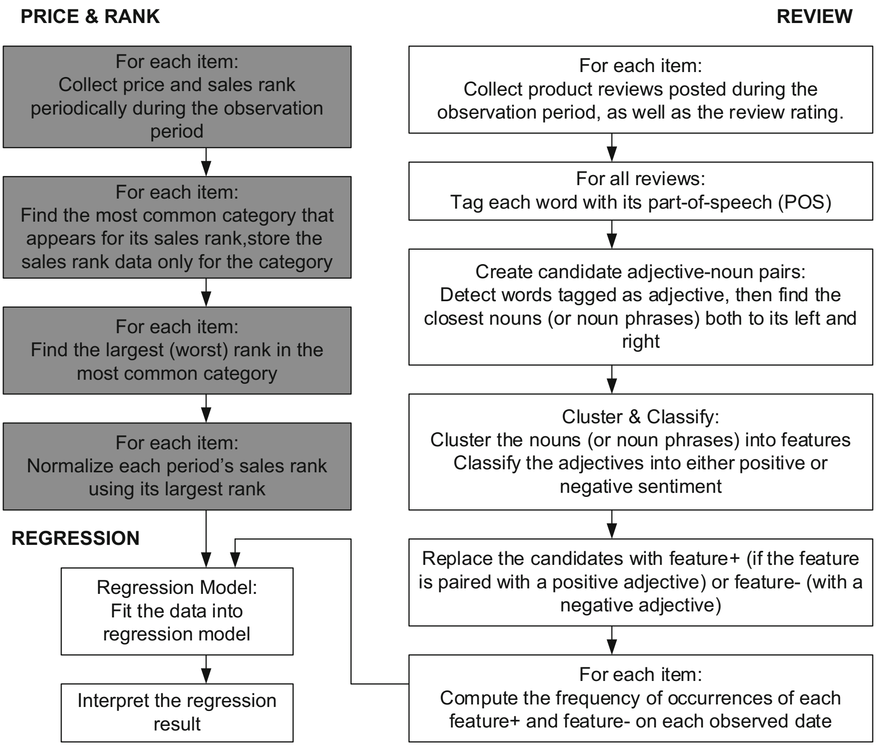


Fig. 1 Research methodology

The purpose of processing price and sales rank is to make the data comparable between items in a fair way. For the product reviews, the data are processed to capture what opinion is expressed towards a feature or the product in general. Afterwards, the regression model is used to draw the relationship between price and product reviews with the sales rank.

Data Processing

The data for this research comes from the publicly accessible product webpages in Amazon.com. In particular, the product category taken as example for this paper is wearable technology. At the beginning of data collection on August 8th, 2015, there were 140 items under the category. For each of those 140 items, its price and sales rank were collected periodically. The time between data collections is between 24 and 48 hours. At the end of data collection on September 30th, 2015, there had been collected 50 data points for each item.

A single data point correspond to a particular date when price and sales rank were recorded. There were three dates when the price and sales rank were not recorded, therefore for each item there are only 50 data points out of 53 days in the period from August 8th to September 30th.

Besides price and sales rank, another important part of the data is the customer review data. In Amazon.com, customers can write a free form review, i.e. there are no specific slots to put either the positive or negative aspects of the product under review. A review data contains the review content itself, as well as the name of the product being reviewed, the date of the review, the URL, and the rating given by the reviewer. An example of a review data is as follows:

Product's Name: UP 24 by Jawbone Activity Tracker - Medium - Onyx (Discontinued by Manufacturer) (item 22)

Date: 2015-09-09

URL: http://www.amazon.com/gp/customer-reviews/R27LEGMLOOGVFB?ASIN=B00GQB1JES

Rating: 2

Review Content: "I got this tracker on super sale for $32, so I can't complain too much. The sleep tracker is nice, not 100% accurate, by probably still my favorite feature. My biggest complaint is the step tracking. It seems that the only steps that count are those "I'm explicitly walking to be walking." I'm a professor who lectures for 150 min each day, constantly walking around, but the tacker doesn't register any of these. In fact, I have the alarm go off after 30 min of inactivity, and it never fails to go off during each class, sometimes while I'm going up and down each row handing out papers. Maybe I just walk funny?"

The example shown above is a quite informative review, in the way that it describes what features the user likes from the product ("sleep tracker"), how the feature works ("not 100% accurate"), what features the user dislikes ("step tracker"). The example also shows the user's sentiment level towards a feature, such as the inaccuracy of sleep tracker is still acceptable ("probably still my favorite feature"), but the wrongly functioned step tracker causes a big disappointment ("My biggest complaint is the step tracking"). However, obviously, not every review states detailed information. For example, some reviews simply mention whether or not the product works. Two of such reviews for the same product are shown below:

Date: 2015-08-27

URL: http://www.amazon.com/gp/customer-reviews/R7MEVP4K4SCV0?ASIN=B00GQB1JES

Rating: 5

Review Content: "Gave as a gift but I have one and enjoy it"

Date: 2015-09-04

URL: http://www.amazon.com/gp/customer-reviews/R25YL7VS906KQN?ASIN=B00GQB1JES

Rating: 1

Review Content: "Crappie product. Stopped working after 90 days."

Visualization of Price and Sales Rank

Price is one of main factors that may affect customer's decision to buy a product. Among the wearable technology items observed in this research, the price of some items shows fluctuating behavior, while others are relatively stable. An example is shown on the graph in Fig. 2, where items 2, 4, and 11 show different price behavior. Day 1 on the graph corresponds to the date August 8th, 2015 and Day 50 corresponds to September 30th, 2015.

Although the price is relatively stable, however, the sales rank could still fluctuates. As shown in Fig. 3, the sales rank movement for item 4 (whose price is relatively stable) fluctuates as much as the other two items.

The Y-axis of the graph in Fig. 3 is the Relative Rank, a metric that would be explained in the next section. This suggests that the fluctuation of an item's sales rank does not solely depend on its price. There are other factors that drive the movement of it. It is hypothesized in this research that product review contributes to the sales rank movement (which is measured by an item's Relative Rank).

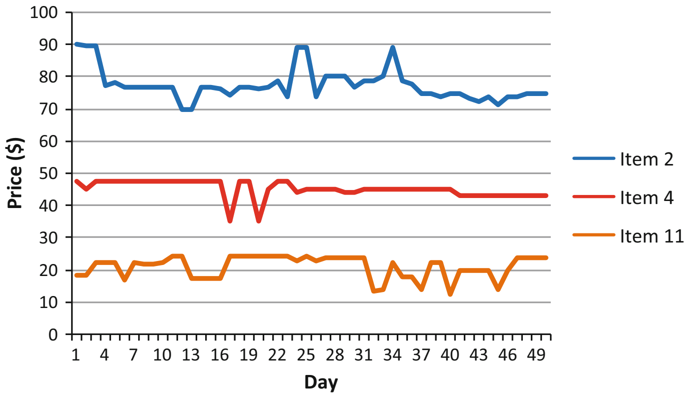


Fig. 2 Observed price for selected items

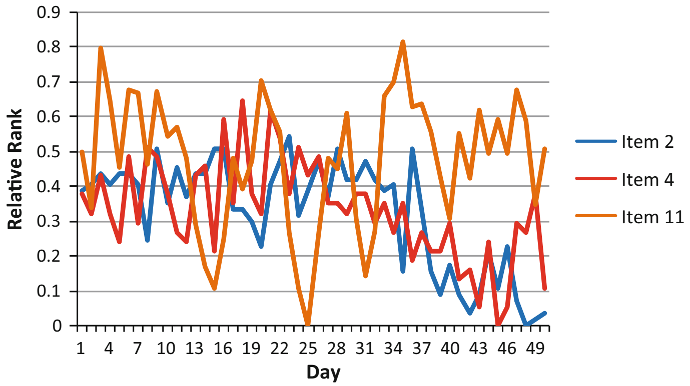


Fig. 3 Observed relative rank for selected items

Pre-processing Sales Rank

Since there are 50 data sets collected periodically for each of the 140 items, the total data set is composed of 7,000 data points. Each data point correspond to the sales rank and price of a particular item on a particular data collection date. For example, on September 9th, 2015, the item 'UP 24 by Jawbone Activity Tracker—Medium —Onyx (Discontinued by Manufacturer' (item 2) was set on the price at $79.00 and its ranking was shown as:

```
Amazon Best Sellers Rank: #1994 in Cell Phones & Accessories (See
Top 100 in Cell Phones & Accessories) #21 in Sports &
Outdoors > Outdoor
```

```
Recreation > Accessories > Electronics > Pedometers    #27    in
Electronics > Outlet   #30   in   Sports   &   Outdoors > Outdoor
Recreation > Accessories > Electronics > Fitness Trackers.
```

For each item, Amazon displays different categories in its Sales Rank section. Moreover, even for the same item, the categories may change from one day to another—although most of the times, they do not.

Since sales rank becomes the important proxy for the actual sales volume (or market share), the sales rank data needs to be pre-processed such that the rank for an item is consistent, i.e. coming from the same category for all dates observed. Moreover, the movement in sales rank for an item should be comparable to that of another item.

First, the sales rank data for an item on a particular day is simplified, such that it only contains the rank number and its corresponding rightmost category, which is the most specific category. For example, the sales rank data for item 2 on September 9th above is processed to become:

```
['#1994',   '#21',   '#27',   '#30',   'CellPhones&Accessories',
'Pedometers', 'Outlet', 'FitnessTrackers']
```

where the first number corresponds to the first category, i.e. #1994 corresponds to 'Cell Phones & Accessories' category respectively.

Afterward, in order to find a consistent category for each item, the frequency of each category is computed. For example, for item 2, all four categories 'Cell Phones & Accessories', 'Pedometers', 'Outlet', 'Fitness Trackers' appear in all 50 observed dates. To break the tie, 'Fitness Trackers' is arbitrarily chosen as the category to represent the sales rank of this item. Another item, 'Misfit Shine—Activity and Sleep Monitor' (item 128) has sales rank in category 'Electronics' in 50 observed dates, 'Fitness & Activity Monitors' in 48 dates, and other categories are even in fewer dates. Therefore, the sales rank data used for item 128 are the ones from 'Electronics' category. Other item, however, may have different most common category and thus be different from item 2. Item 1, for example, uses the sales rank data from 'Monitors' category.

In order to make the sales rank comparable between items having sales data from different categories, it is not the raw sales rank number used, but the rank is normalized relative to its own worst rank. After normalization, each item's relative rank would be in the range of [0, 1), in which the larger the value is the better it is. The normalization is done for each item using the following formula:

$$RelativeRank_{it} = 1 - (Rank_{it}/MaxRank_i) \tag{1}$$

where: RelativeRank_{it} = relative rank of item i on day t, Rank_{it} = actual rank of item i on day t, MaxRank_i = maximum actual rank of item i during the observation period.

Table 1 shows an example for normalizing the actual sales rank data of item 2 into relative rank. During the 50 days of observation period, the worst rank for this item is 57 on day 47, so MaxRank_i = 57 and RelativeRank for that day becomes zero.

Table 1 Conversion of actual rank into relative rank for item 2

Day (t)	$Rank_{it}$	$RelativeRank_{it}$
0	35	0.38596
1	34	0.40351
2	32	0.43860
3	34	0.40351
4	32	0.43860
5	32	0.43860
…	…	…
47	57	0.00000
48	56	0.01754
49	55	0.03509

Processing Product Review

After collecting all the reviews during the observation period, each word is tagged by its part-of-speech (POS). The POS tagger utilized in this research is from Python Natural Language Toolkit (NLTK). The words tagged as adjective (JJ) are collected, since they are the most likely words to be used to express one's opinion about a product. Using the previously shown review for item 22, the JJ-tagged words are presented below as underlined words.

`Review Content:` "`I got this tracker on super sale for $32, so I can't complain too much. The sleep tracker is` <u>`nice`</u>`, not 100%` <u>`accurate`</u>`, by probably still my` <u>`favorite`</u> `feature. My biggest complaint is the step tracking. It seems that the` <u>`only`</u> `steps that count are those "`<u>`I'm`</u> `explicitly walking to be walking." `<u>`I'm`</u> `a professor who lectures for 150 min each day, constantly walking around, but the tacker doesn't register any of these. In fact, I have the alarm go off after 30 min of inactivity, and it never fails to go off during each class, sometimes while` <u>`I'm`</u> `going up and down each row handing out papers.` <u>`Maybe`</u> `I just walk` <u>`funny?`</u>"

Starting from the position of each JJ-tagged word, a search procedure to find the closest nouns is performed both to the left and to the right of the JJ-tagged word's position. The nouns are expected to be in the same sentence with the adjective, so the search is terminated once another adjective, a dot, the start of a review, or the end of a review is encountered. The procedure is described as follows.

Search Procedure to Obtain Feature Candidates

1. Collect all JJ-tagged words.
2. For each JJ-tagged word, do the Left Search procedure:

   a. Move a position to the left, get the word.
   b. Return an empty set { }, if the word is an adjective-tagged word, a period, or the start of a review.

c. Collect the word into Noun set, if the word is a noun or an indication of negative form ("not", "n't", "no"). Go back to step (a).
d. Return Noun set, if Noun set is not empty and the word is not a noun. Otherwise, go back to (a).

3. For each JJ-tagged word, do the Right Search procedure:

   a. Move a position to the right, get the word.
   b. Return an empty set { }, if the word is an adjective-tagged word, or a period.
   c. Collect the word into Noun set, if the word is a noun or an indication of negative form. Go back to step (a).
   d. Return Noun set, if Noun set is not empty and the word is not a noun. Otherwise, go back to (a).

For example, from the first adjective "nice", the first noun encountered to its left is "tracker". The Left Search is continued because the next word is still a noun, i.e. "sleep". Afterwards, the search is stopped because the next word is not a noun. The Right Search for the same adjective "nice", does not found any noun until it is terminated when encountering the next adjective-tagged word, i.e. "accurate". It finds an indication of negative form, which is recorded as a "*NOT". The result of all candidates for item 22's review on September 9th, including the ones that contains only an adjective with no closest nouns, is shown below:

```
22, '"2015-09-09",', [[('sleep', 'NN'), ('tracker', 'NN'), ('nice',
'JJ')], ['*NOT', ('accurate', 'JJ')], [('count', 'NN'), ("i'm", 'JJ')],
[('class', 'NN'), ("i'm", 'JJ')], [('papers', 'NNS'), ('.maybe', 'JJ')],
[('nice', 'JJ')], [('accurate', 'JJ')], [('favorite', 'JJ'), ('fea-
ture', 'NN')], [('only', 'JJ'), ('steps', 'NNS')], [("i'm", 'JJ')],
[("i'm", 'JJ'), ('professor', 'NN')], [("i'm", 'JJ'), ('going', 'NN')],
[('.maybe', 'JJ')], [('funny?', 'JJ')]]]
```

There are nearly 11,000 noun(s) collected, with the most frequent term to appear is "product". The majority of the terms appear very few times, as more than 7000 terms appear only three times or less. Table 2 show some examples of words that appear the most frequently to the words that appear just once.

It can be noticed from the table that some terms directly indicate the product itself in general ("product", "device", "fitbit"), or either a feature or service related to the product ("app", "battery", "customer, service"). Meanwhile, there are other nouns that are not related or too general to be related to any of those aspects ("months", "i", "something").

Based on the observation from the collected nouns, it is required to screen the nouns by selecting informative nouns and then cluster the nouns with similar meaning into the same cluster. The total frequency for all nouns is 98,474 times. Nouns appear less than three times are considered unimportant, so they are discarded. After discarding those terms, the ones left appear 86,547 times (87.9% of all nouns).

It is determined to cluster the nouns into 22 clusters, i.e. Activity, App, Battery, Button, Calorie, Charge, Clip, Color, Design, GPS, Heart, Indicator, Price, Product,

Table 2 Selected nouns sorted by frequency of occurrences in product review

Frequency	Noun(s)
1581	product
1452	band
1399	time
1342	app
1208	device
1102	months
1041	steps
894	watch
862	day
819	days
794	fitbit
769	its
728	battery
…	…
338	battery, life
337	things
335	money
329	i
322	something
317	customer, service
…	…
1	consumers
1	consumption
1	contact, dermatitis

Quality, Screen, Service, Sleep, Sync, Time, Tracker, Water. The clusters are determined based on the features found for these products. Other clusters describe a product in general, or more general aspects of a product (e.g. design, price, quality, service).

A part of the result after clustering the nouns is shown on the Table 3. It displays five most frequent terms for three clusters, i.e. Activity, Battery, and Sleep. In this paper the clustering is done manually based on subjective judgments.

The next task is classifying whether an adjective is more likely to express a positive or a negative sentiment. The classification is done by matching each adjective with the sentiment lexicon (Hu and Liu 2004a; Liu et al. 2005). If an adjective appears in the positive set of the sentiment lexicon, then it is classified as positive, and vice versa. However, if there is an indication of negative form in the adjective-noun pair candidate, then the sentiment is reversed. For example, a candidate [‘`*NOT`’, (‘accurate’, ‘`JJ`’)] is assigned with a Negative sentiment, although “accurate” appears in the positive set of the sentiment lexicon.

After clustering the nouns and classifying the adjectives, each candidate is reviewed. If the noun belongs to any specified cluster, then it would be replaced by

Table 3 Top 5 words for selected clusters (Activity, Battery, Sleep)

Frequency	Term	Cluster
1041	steps	Activity
456	activity	Activity
243	fitness, tracker	Activity
232	activities	Activity
178	activity, tracker	Activity
728	battery	Battery
338	battery, life	Battery
104	batteries	Battery
40	battery, lasts	Battery
22	battery, dies	Battery
573	sleep	Sleep
67	sleep, mode	Sleep
57	sleeping	Sleep
53	sleep, tracker	Sleep
30	sleep, monitor	Sleep

the cluster's name. Then, the adjective attached to the noun is replaced by its sentiment. For the item 22's review above, from all candidates, the result after replacement becomes:

```
[22, "“2015-09-09”,', ['sleep+']]
```

which comes from this particular candidate:

```
[('sleep', 'NN'), ('tracker', 'NN'), ('nice', 'JJ')].
```

Other candidates for that review data only contains adjective (e.g. "not accurate") and there are no nouns found attached to it; or contains both adjective and noun(s) but the adjective is not found in the sentiment database. Therefore, the only feature-opinion pair obtained is 'sleep+'.

Therefore, for every item and its respective dates, whenever a pair of feature and sentiment appears, it is recorded accordingly. An example below shows that in reviews for item 14 on September 9th, 2015, there are positive opinions for design, screen, activity, sync, and tracker:

```
[14, "“2015-09-09”,', ['design+', 'screen+', 'activity+', 'sync+',
'tracker+']]
```

Suppose there are two occurrences of positive opinion for screen, then 'screen+' would appear twice on the list.

Each review also has a rating assigned to it by the reviewer in the range from 1 to 5 stars. These ratings are incorporated into the regression model, because they are a part of the reviews as well.

Regression and Result

The hypothesis is that the movement of an item's sales rank is influenced by significant variables. Among those variables are price, review content and review rating. Thus, the independent variable for the regression model is an item's Relative Rank for a particular observed date, and the dependent variables are its price, the number of reviews, the frequency of each feature-opinion pair that appears on the review on a particular date, as well as its previous date's Relative Rank.

As for the review ratings, the average of those is taken for each date, as well as the maximum and the minimum ratings for the date. These all become dependent variables for the regression model as well. The dates when there were no reviews at all are excluded, such that from the original 7000 data (50 dates for 140 items) there are 4391 data which become the eligible input for the regression model in (2).

$$\begin{aligned} RelativeRank_{it} = {} & b_0 + b_A AverageRating_{it} + b_{M1} MinRating_{it} + b_{M2} MaxRating_{it} \\ & + b_N NumReviews_{it} + b_R RelativeRank_{(t-1)} + b_P Price_{it} \\ & + b_{F+} (F+)_{it} + b_{F-} (F-)_{it} \end{aligned} \tag{2}$$

where: RelativeRank_{it} = relative rank for item *i* on day *t,* $\text{AverageRating}_{it}$ = average review rating given for item *i* on day *t,* MinRating_{it} = the smallest review rating given for item *i* on day *t,* MinRating_{it} = the largest review rating given for item *i* on day *t,* NumReviews_{it} = number of reviews written for item *i* on day *t*$\text{RelativeRank}_{i(t-1)}$ = relative rank for item *i* on day ($t-1$), Price_{it} = price of item *i* on day *t,* $(\text{F+})_{it}$ = frequency of positive opinion about feature *F* of item *i* on day *t,* $(\text{F}-)_{it}$ = frequency of negative opinion about feature *F* of item *i* on day *t.*

It is found out that the statistically significant coefficients ($\alpha = 5\%$) for the regression model are *Relative-1* (previous day's relative rank), *battery*+ (positive opinion about battery), and *sleep*− (negative opinion about sleep tracker), as highlighted in Table 4. The complete linear regression result from the statistical software Minitab is shown in Table 4. It also shows that the R-squared for the model is 67.54%.

Discussions

There are difficulties in the attempt to understand the correct meaning of a discourse, i.e. the product reviews in this research, mainly because there is ambiguity in natural language. The ambiguity appears in the forms of word ambiguity, different representations of similar concepts in human language, and the implicit semantic information in textual data. Thus, it is inherently more difficult than dealing with entirely numerical data (Liu et al. 2007).

Table 4 Linear regression result

Term	Coef	SE Coef	*T*-Value	*P*-Value	VIF
Constant	0.0922	0.0107	8.59	0.000	
averageRating	−0.00503	0.00530	−0.95	0.342	7.70
minRating	0.00337	0.00288	1.17	0.242	4.81
maxRating	0.00008	0.00359	0.02	0.983	3.53
numReviews	−0.000043	0.000232	−0.18	0.854	1.73
Price	−0.000075	0.000042	−1.78	0.075	1.22
Relative-1	0.82011	0.00885	92.63	0.000	1.03
activity+	0.00104	0.00537	0.19	0.846	1.11
activity−	−0.0148	0.0129	−1.15	0.251	1.08
app+	0.00137	0.00545	0.25	0.801	1.06
app−	0.0105	0.0123	0.85	0.395	1.24
battery+	0.02829	0.00958	2.95	0.003	1.05
battery−	−0.0158	0.0161	−0.98	0.329	1.05
button+	0.0335	0.0369	0.91	0.364	1.07
button−	−0.0300	0.0509	−0.59	0.556	1.02
calorie+	−0.0054	0.0197	−0.28	0.783	1.06
calorie−	−0.0225	0.0193	−1.16	0.244	1.10
charge+	0.0223	0.0159	1.41	0.159	1.05
charge−	−0.0292	0.0184	−1.59	0.113	1.01
clip+	−0.0011	0.0120	−0.09	0.925	1.09
clip−	−0.0227	0.0135	−1.69	0.091	1.05
color+	−0.0132	0.0167	−0.79	0.432	1.02
color−	0.0631	0.0570	1.11	0.269	1.07
design+	−0.00685	0.00975	−0.70	0.482	1.04
design−	0.0168	0.0243	0.69	0.490	1.08
gps+	0.0123	0.0314	0.39	0.695	1.04
gps−	−0.0122	0.0891	−0.14	0.891	1.04
heart+	0.0053	0.0140	0.38	0.704	1.10
heart−	0.0350	0.0227	1.54	0.123	1.08
indicator+	−0.0342	0.0553	−0.62	0.536	1.00
indicator−	−0.0444	0.0562	−0.79	0.429	1.04
price+	0.01076	0.00709	1.52	0.129	1.08
price−	−0.0308	0.0189	−1.63	0.103	1.02
product+	−0.00354	0.00323	−1.09	0.274	1.15
product−	−0.01032	0.00588	−1.76	0.079	1.12
quality+	−0.0155	0.0192	−0.81	0.418	1.05
quality−	0.0190	0.0280	0.68	0.498	1.03
screen+	0.0087	0.0116	0.75	0.451	1.08
screen−	0.0052	0.0175	0.30	0.766	1.23
service−	0.0029	0.0106	0.27	0.784	1.07

(continued)

Table 4 (continued)

Term	Coef	SE Coef	*T*-Value	*P*-Value	VIF
service+	−0.00517	0.00628	−0.82	0.410	1.15
sleep+	0.00418	0.00975	0.43	0.668	1.05
sleep−	−0.0341	0.0162	−2.11	0.035	1.10
sync+	−0.0097	0.0220	−0.44	0.659	1.08
sync−	0.0211	0.0329	0.64	0.522	1.06
time+	−0.00161	0.00717	−0.23	0.822	1.10
time−	−0.0128	0.0101	−1.26	0.206	1.05
tracker+	0.00484	0.00840	0.58	0.564	1.06
tracker−	0.0083	0.0165	0.50	0.615	1.08
water+	0.0223	0.0225	0.99	0.321	1.02
water−	−0.0176	0.0199	−0.88	0.376	1.01

Model summary

S	R-sq	R-sq(adj)	R-sq(pred)
0.123371	67.54%	67.16%	66.85%

From the first step where the words in the reviews are tagged with their part-of-speech (POS), ambiguity has arisen in the form of a word that can have more than one possible tag. Considering the inaccuracies from Python NLTK tagger, in the future it may be useful to use other taggers and compare their accuracies. Furthermore, including the context of a word may improve the POS tagger's accuracy as well.

It is also assumed in this paper that an opinion is expressed with an adjective. Although it is true in many cases, in fact there are also opinions which are expressed in verbs, such as "like", "dislike", or "hate". Adding these verbs may increase the recall and precision of opinion detection. As there are more opinions detected, the input would be less sparse than the current condition where there are lots of 0's observed for each date's either *feature*+ or *feature*− frequencies. As more data have been collected, in the future the data will be aggregated, e.g. in a weekly basis, such that it reflects the idea that people read not only a particular date's reviews but also the reviews posted several days before.

It is also assumed that the feature (noun or noun phrase) related to an adjective is the only one closest to the adjective, and there is no other adjective between them. As the result, in a sentence such as "sleep tracker is nice, but not 100% accurate", the feature 'sleep tracker' is only linked to the adjective "nice", but not to "not accurate". In the future, a more sophisticated approach would be required to solve the entity discovery and entity assignment problems, i.e. how to discover entities that are discussed in a sentence and how to assign entities to sentences where entity names are not explicitly mentioned (Ding et al. 2009).

As for the product, wearable technology products are chosen to be observed in this paper because those are relatively new products. Those products are still expected to develop their features. Moreover, the users of those products are typically those who are familiar with technological gadgets. Therefore, it is also expected that the users of this product would be likely to write reviews about the product.

Despite of some drawbacks in the feature-opinion detections and relatively sparse values for the dependent variables, the regression result confirms that sales rank for wearable technology products is significantly influenced by the content of the reviews, its sales rank on the previous day, but interestingly not by the price. A possible reason to explain it might be that the price does not change much during the data collection period, or that the influence is not captured because the dates with no reviews are excluded from the current regression model. Therefore, as stated previously, it might be worthy to aggregate the daily data into weekly data to reveal more meaningful insights.

The result confirms the intuition that negative opinion worsens sales rank, as shown by the sleep tracker's regression coefficient: −0.035 (p-value = 0.044). On the other hand, positive opinion gives an increase towards sales rank, as shown by the battery's regression coefficient: 0.02829 (p-value = 0.003). This result reveals that sleep tracker is more important for customers, compared to other tracking functions offered by these products, i.e. activity tracker and calorie tracker. In particular, bad reviews about the sleep tracker tend to discourage people from buying the product.

Also, based on the result, the review ratings and the number of reviews do not affect the sales rank significantly. The average rating, smallest rating, and largest rating are all not significant. This emphasizes the importance of review content compared to review ratings.

Conclusion

The research confirms the hypothesis that the movement in sales rank does not depend solely on an item's price. In fact, for the case of the products in this paper, the price, the review ratings assigned (average, smallest, largest), and the number of reviews do not significantly influence sales rank, while the positive opinion for battery and negative opinion for sleep tracker are found to have significant effect to sales rank.

For future work, in terms of the model, there might be interesting insight if the products are divided based on their price ranges and their release dates as a newly released product may react differently towards reviews than an old one. Since the R-squared for the current model is 67.54%, more data have been collected in order to strengthen the result in the future. As for the Natural Language Processing part, an improvement for the POS tagger and the development of an automatic method to cluster features need to be done.

References

Archak N, Ghose A, Ipeirotis PG (2011) Deriving the pricing power of product features by mining consumer reviews. Manage Sci 57(8):1485–1509

Chevalier JA, Mayzlin D (2006) The effect of word of mouth on sales: online book reviews. J Market Res 345 Vol XLIII

Chong AYL, Ch'ng E, Liu MJ, Li B (2015) Predicting consumer product demands via Big Data: the roles of online promotional marketing and online reviews. Int J Prod. doi:10.1080/00207543.2015.1066519

Decker R, Trusov M (2010) Estimating aggregate consumer preferences from online product reviews. Int J Res Market 27:293–307

Ding X, Liu B, Zhang L (2009) Entity discovery and assignment for opinion mining applications. In: International conference on knowledge discovery and data mining (KDD'09)

Forman C, Ghose A, Wiesenfeld B (2008) Examining the relationship between reviews and sales: the role of reviewer identity disclosure in electronic markets. Inf Syst Res 19(3):291–313

Hu M, Liu B (2004a) Mining and summarizing customer reviews. In: Proceedings of the ACM SIGKDD international conference on knowledge discovery and data mining (KDD-2004)

Hu M, Liu B (2004b) Mining opinion features in customer reviews. In: AAAI'04 Proceedings of the 19th national conference on artificial intelligence, pp 755–760

Liu B, Hu M, Cheng J (2005) Opinion observer: analyzing and comparing opinions on the web. In: Proceedings of the 14th international world wide web conference (WWW-2005)

Liu Y, Lu WF, Li HT (2007) Knowledge discovery and management for product design through text mining—a case study of online information integration for designers. In: International conference on engineering design, ICED'07

Liu Y, Jin J, Ji P, Harding JA, Fung RYK (2013) Identifying helpful online reviews: a product designer's perspective. Comput Aided Des 45(2013):180–194

Sun M (2012) How does the variance of product ratings matter? Manag Sci 58(4):696–707

Tucker C, Kim HM (2011) Predicting emerging product design trend by mining publicly available customer review data. In: International conference on engineering design, ICED11

Author Index

A
Abdelmohsen, Sherif, 459
Adams, Robin S., 613
Agogué, Marine, 99
Al-Sayed, Kinda, 537
Assem, Ayman, 459

B
Becattini, Niccolò, 555
Boa, Duncan, 575
Burge, Janet E., 633

C
Cagan, Jonathan, 173, 267, 401
Cascini, Gaetano, 555
Chao, Jie, 613
Chen, Chih-Chun, 359
Chen, Tian, 279
Clausner, Timothy, 39
Corney, Jonathan, 209
Crilly, Nathan, 359

D
Dinar, Mahmoud, 193
Dong, Andy, 81
Duarte, José P., 495
Duffy, Alex H.B., 135

E
Ezzat, Hicham, 99

F
Frazer, Keith, 157

G
Galil, Omar M., 115
Garcia, Sara, 229
Gero, John S., 39, 301, 653
Gill, Amaninder, 667
Goel, Ashok K., 157
Goldstein, Molly, 613
Goucher-Lambert, Kosa, 173
Grace, Kazjon, 593
Grealy, Madeleine, 135

H
Hale, Jonathan, 437
Hancock, William, 157
Hay, Laura, 135
He, Yuejun, 23
Heylighen, Ann, 81
Hicks, Ben, 575
Hunter, S.T., 59

I
Ibrahim, Ahmed, 459
Inoue, Shiro, 3

J
Jiang, Hao, 301
Jin, Yan, 379
Justice, Connor, 633

K
Kim, Harrison M., 685
Kirsh, David, 419
Kotovsky, Kenneth, 401
Krstic, Djordje, 249

L
Lai, Song Liang, 321
LeDuc, Philip, 267
Lee, Lina, 39
Le Masson, Pascal, 99
Lu, Dizhou, 379
Luo, Jianxi, 23

J.S. Gero (ed.), *Design Computing and Cognition '16*,
DOI 10.1007/978-3-319-44989-0

M
Maclachlan, Ross, 209
Maher, Mary Lou, 39, 593
Martusevich, Kirill, 115
Mathur, Tanmay, 633
McComb, Christopher, 401
McKay, A.S., 59
McTeague, Chris, 135
Miller, S.R., 59
Montagna, Francesca, 555
Moss, Jarrod, 173
Mueller, Caitlin T., 477

N
Najjar, Nadia, 593
Niblock, Chantelle, 437
Nourian, Saeid, 613

O
O'Shields, Steven, 339

P
Patel, Apurva, 339
Patel, Sayjel V., 477
Penn, Alan, 537
Pidgeon, Laura M., 135
Polimera, Meghna, 193
Purzer, Şenay, 613

R
Rodgers, Paul A., 3
Rogers, Benjamin, 633
Rudolph, Stephan, 513

S
Sarkar, Somwrita, 653
Sass, Larry, 419
Shekhawat, Krishnendra, 495
Sen, Chiradeep, 115
Shah, Jami, 193
Shea, Kristina, 279
Shu, L.H., 321
Smithwick, Daniel, 419
Spencer, Nick, 3
Sri Ram Mohinder, C.V., 667
Starkey, E.M., 59
Summers, Joshua D., 339, 667
Suryadi, Dedy, 685

T
Tahsiri, Mina, 437
Tam, Mark K.M., 477
Tarabishy, Sherif, 459
Taysom, Eloise, 359
Tennant, Andy, 3
Thiagarajan, Ashwinraj, 339
Tonhäuser, Claudia, 513
Tuchez, Christian, 157

V
Vasantha, Gokula, 209
Vasconcelos, Luis A., 359
Vuletic, Tijana, 135

W
Weil, Benoit, 99
Whiting, Mark, 267
Wilson, David, 593
Wodehouse, Andrew, 209

X
Xie, Charles, 613

Y
Yu, Rongrong, 39

Z
Zimmermann, Luca, 279