Computational Design in Distributed Teamwork

Using digital and non-digital tools in architectural design competitions

Livanur Erbil Altintas[1], Altug Kasali[2], Fehmi Dogan[3]
[1,3] Izmir Institute of Technology [2] Izmir University of Economics
[1,3] {livanurerbil|fehmidogan}@gmail.com [2] altugkasali@gmail.com

This paper reports a case involving computational practices in design process with an aim to understand the role of digital and non-digital tools in the design process. Following an ethnographic approach, we aimed at understanding the nature of the interactions among team participants which are human and non-human in a distributed system. We focused on computational practices in design process and we aimed to understand the role of digital and non-digital tools in the design process. Tools have remarkable role in a distributed system in the sense of propagation of knowledge. It was observed that form exploration by digital tools may not controlled as much as sketching.

Keywords: *Collaboration, Distributed Cognition, Computational Design, Visual Calculation*

INTRODUCTION

Design is a distributed cognitive system consisting of interaction, computation, generation, communication, synthesis, and manipulation of tasks (Lyon, 2005, Lyon, 2011, Cross, 2006) by a series of human and non-human agents. In the architectural design processes, designing and construction stages involve various participants from different disciplines who contribute to solutions to design problems in light of their responsibilities shaped by their areas of expertise. In such a system, one question that stands out relates to the nature of collaboration among participants from different disciplines and with different expertise. According to Cuff (1992), an architect or a designer can have a key role in design decisions but almost every step in practice is associated with collaborative work. Architects can be an expert in many subjects such as space design, aesthetics, site planning, function, structure, mechanical systems, graphic communication, but it is obvious that they need experts from other disciplines (Cuff, 1992). Recently, various digital technologies are used to increase participation and collaboration to increase efficiency in problem solution. The research focuses on architectural practices as a distributed cognitive system to understand the complex project generating mechanisms mentioned above. Analytically, researching architectural project production mechanisms as a system requires a holistic view. The system that comprises humans, objects, and tools can be framed as a distributed cognitive system following the work of Ed Hutchins (Hutchins, 1995). Distributed cognition assumes that any task can be distributed across different parts of a system (Hutchins, 2014). Collaboration, in this framework, involves not only human participants but also other parties such as tools and representations. In this research we have employed the distributed cognition framework in order to account the interactions among human and non-human components of a system.The research focuses on the activities of a team of architects using computational design tools in the design process leading to a proposal for a competition. Within the case the team has employed the computational tools in developing solutions for secondary systems or components, rather than generating the overall form of the design. Given the constraints of the competition program, the computational tools were observed to be instrumental in increasing the number of alternatives, in problem solution, and in streamlining time-consuming tasks. The paper tries to explore the mechanisms, online and offline, in developing particular solutions which required a level of coordination between participants of the cognitive system. The next section introduces the idea of distributed cognition in the design context we have observed through ethnographic field methods.

DISTRIBUTED COGNITION

People interact with other people, artifacts, technologies, tools, surfaces, and the things that are represented to the others. People also interact with their environments as being 'embedded' to coordinate their internal cognitive tasks with external tools (Kirsh, 2008). People's cognitive activities results from interactions with external cognitive artifacts and with other people's activities in a task that are determined by social and cultural contexts and physical environment that they are positioned in (Suchman, 1987, Hutchins, 1995). Distributed cognition discovers and explains the principles of coordination, externalization, representation, and interaction (Hutchins, 1995). Distributed cognition frames the cognitive process of human and non-human mechanisms that are participate in a task (Hutchins, 2004). Hutchins (1995a) describes computation observed in the activity of the larger system as "computation realized through the creation, transformation, and propagation of representational states" . According to Hutchins (1995), to understand navigation system, we need to understand information processing system within the organization. In this research, we assume that an architectural design team as a system consists of human and non-human participants. Each of the participants contribute to the system and information is propagated between them. We, now, turn our attention to computing practices in the distributed cognitive system observed in this research.

COMPUTATIONAL DESIGN AND VISUAL CALCULATION

It is claimed that the computational thinking changes the way we think (Bundy, 2007). Computation has always been a part of design (Carpo, 2011), however the adaptation of computational design tools has been a relatively new phenomenon. Design always proceeds in the form of a calculation and can be explained with algorithms and computation (Stiny and Gün, 2012, Pask, 1963). Computations involve numbers as well as shapes. Stiny (2006) states that design and calculating is equal and puts forward that in designing designers "are actually calculating in a visual sort of way, whether you know or not, and the real central issue, at least for most of my work, is to try to figure out how calculating includes design" (Stiny and Gün, 2012). Visualizing design ideas, according to Stiny (2006), is a form of calculation through which ideas are represented and tested. According to Stiny (2012), designer thinks with eye and seeing is the most interesting part of the design process. The formulation offered by Stiny is a form of "reflective practice"; an interactional process in design which follows "seeing-drawing-seeing" defined by Schön (1991), designer reads the situation and interprets again while drawing in the design process. Schön emphasizes sketching as a valuable tool for representation and exploration (Schön, 1991). Concerning the research presented here, the emerging question whether or not the tools employed in computational design replaces the conventional sketching practices involving representation and exploration. Stiny (2006) states that visual rules are also used for calculation for formal operations in design. However, designers generally unaware of possible alternatives of their actions while designing (Visser, 2010). Within this perspective, it is a valuable research question to think about the relationship between sketching activity and computing activity.

METHODS AND THE CASES

The research is a qualitative study which consists of ethnographic observations of a professional architectural team. The selected office uses digital technology and computing in their design process. Data collection included the following stages. First, we conducted in-situ observations of two competition projects from beginning to end for a month. Ethnographic observations were used to understand groups and people in their everyday professional lives (Emerson et al., 1995). Second, we conducted semi-structured interviews with significant team participants. The semi-structured interviews were face-to-face to provide a way to explore feelings, opinions and behaviors (Sommer and Sommer, 1997). The interviews helped in figuring out the teams' information processing systems in the process, in understanding the communication strategies and knowledge representation techniques of the teams, and in providing a lens to understand participants' descriptions of a situation. Data analysis included the following three phases: description, analysis, and interpretation of culture-sharing group (Creswell, 2007). In the first phase, all the data has been indexed in a time line to understand the phases of the team's design process. Data types have been coded in the time line as sketch, photograph, field notes, meeting minutes, video records, audio records, screenshots, and e-mails. Grounded theory involves analytic attention and provides a procedure for developing categories of information which is called open coding (Strauss and Corbin, 1990). In the open coding phase, the text is examined and emergent categories are identified by the researcher. Categories that are listed according to selected phenomena, are interconnected as axial coding to create categories. Creating relationships between the categories and building a 'story' which connects categories is called as selective coding (Strauss and Corbin, 1990).In this research, the focused team consisted of one architect team leader (TL), two architect job captains (JC1, JC2), and seven intern architects. The team participated in two different architectural competitions at the same time. Competition A was about a youth center design, Competition B was about a municipal service building. In the team, one job captain (JC2) and one intern (IA1) were more interested and capable of computational design tools then the other participants.

FORM EXPLORATION BY TOOLS

Computational design tools are seen as expediting the design problem solving, especially the tools enable searching a wide range of solutions, and visualizing the data to establish collaboration among participants (Olsen and Namara, 2014). Exploration of the form was carried out by the job captain and the team leader in the design process of the competition A's meetings. The competitive nature of the design process has adapted all the team participants' motivation about idea generation. In the meetings of the competition A, the team developed sets of rules in the sense of settlement on the project site, design and size of the lodge units that were given in the competition requirements, and other aspects such as creation of 'legibility' of the lodges array. The team leader was meticulous about 'legibility' issue to express the idea to the jury:

00:16:13 TL: *Now, without a rule, when we get random, it will look as if we just put it randomly. No architect can read that. He or she can't interpret if he/she can't read. Let me tell you so... He should understand it the moment he looks at it.*

The job captain developed multiple alternatives while sketching (figure 1) and calculating the rules and then put the information into the software as 2D drawings (table 1). While sketching the lodge units' organization, the job captain was searching the criteria to have a pattern and image, developed in the meetings (figure 1). Rules were aimed to create legible but at the same time, aimed to express randomness. The job captain about suspicions about randomness:

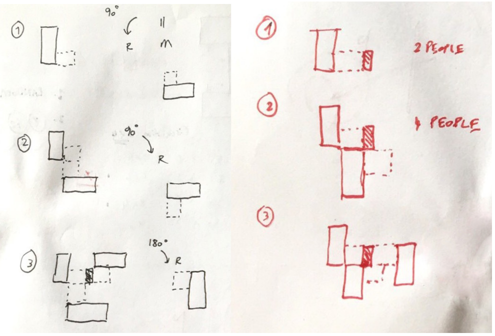


Figure 1
Sketches about the rules, developed by the job captain

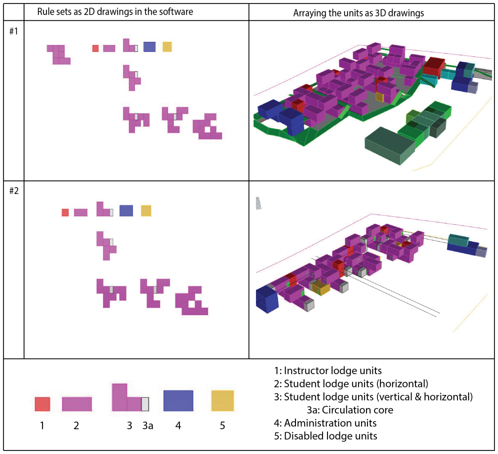


Table 1
(left) 2D drawings of the rule applied units design; (right) 3D drawings of the alternatives: red module

00:00:49 JC: *I really did it with random mirror rotation. Second, it also makes more sense to do so. We're going to multiply this, or I don't know, there will be a unit of 8. I think you put 8, 6, so you made a unit according to the rule. Because it seems to be a little fuller, we made 8 and we did 6, we made 10. Maybe, 3 units, perhaps, disintegrate. Because if we do the same thing all the time, will it get boring? I am thinking about that...*

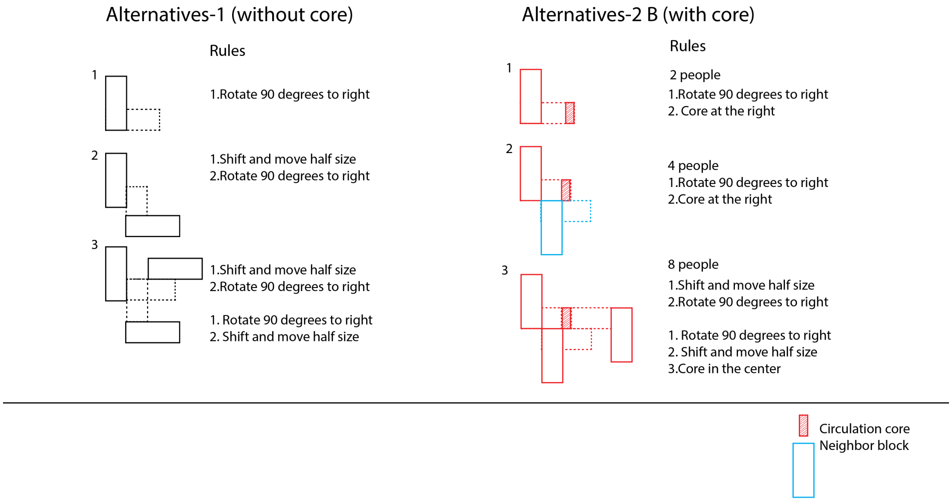
The following figure (figure 2) illustrates and explains the rules that the job captain developed during the design process of the units:

In the following table, some of the alternatives of the units are presented (table1). One of the interesting points that the job captain followed was the steps he took: calculation, drawing, and seeing while sketching. Similar steps were followed in the digital tool, calculation, drawing as 2D, and then to see the created alternative, the job captain applied the rules in the site organization by arraying the units according to rules.

SOLVING DESIGN PROBLEMS BY TOOLS

In this section, we take a look into a part of the competition B where the team leader and the intern architect worked in collaboration in finding façade scheme. In competition B, the team leader dominated the design project more than the project A. Rather than creating multiple alternatives, the team leader described a façade design to the code developer who is also an intern architect (IA1), and then IA1 applied the design idea on Grasshopper by developing rules. In the façade design, wooden elements arrays on the upper floors of the building façade design solutions were applied. The team leader sketched the idea as description of numbers and shapes, then the code developer re-sketched to understand trying to find out exceptional situations such as wooden elements arrays on the corner points of the façade. Afterwards, the code developer developed algorithms and applied on Grasshopper software. In Figure 3, the team leader's sketches are represented. Wooden elements were designed in order to take natural light by filtering into the interior space surrounding of the glass façade. In the sketches (figure 3) the team leader defines the size and proportions of the wooden façade elements. Presumptive perspective and sectional sketches were also developed by the team leader in the meetings. At this point, the team leader gave exact dimensions for the façade elements but façade view and effect were not defined precisely.

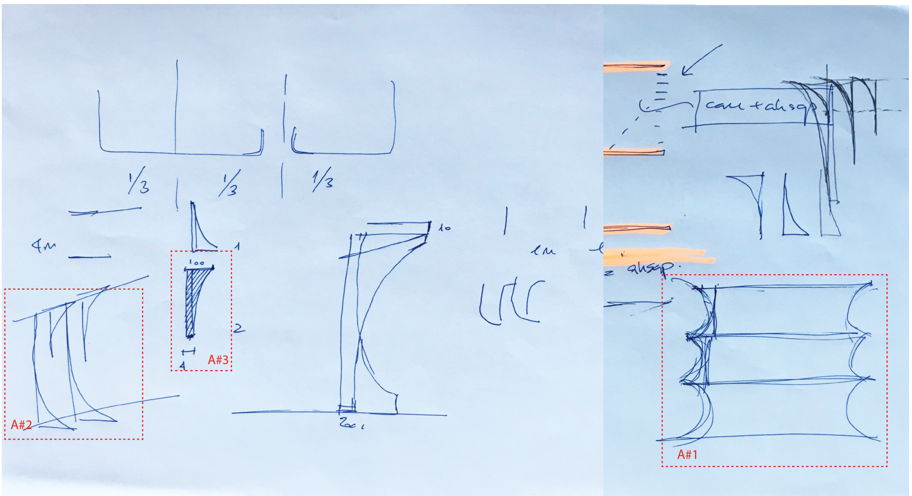
Figure 2
The illustrations and explanations of the rules that developed by the job captain



However, the code developer re-sketched before entering the data (figure 4). Sketches and notes (figure 4) were about the arrangement of the wooden elements on the façade. The sketches represent the elements' dimensions and radius calculations, and the notes are about the wooden elements' placement on the corner of the façade.

In the semi-structured interviews, IA1 explained the rules that applied for the façade works. Once identifying three different types of the wooden elements that would surround the façade, the gaps between the elements were determined. Following that, different ranges and forms were specified as rules, IA1 tested multiple alternatives of the config-

Figure 3
Façade sketches; details and dimensions, developed by the team leader



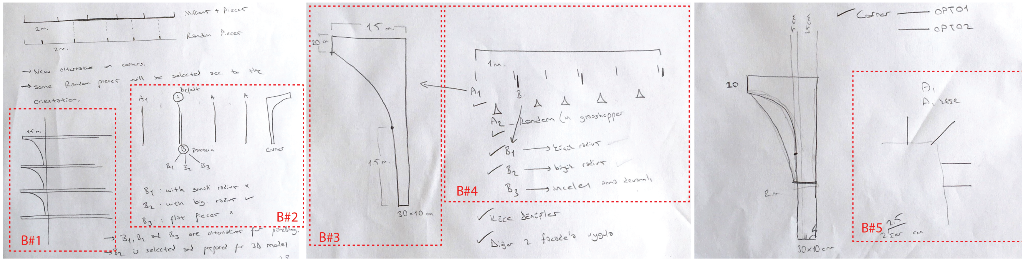


Figure 4
Sketches by the IA1 on the arrangement of wooden elements

uration of the elements on Grasshopper. Within the limits of the rules that are defined by the designer, in the meantime within the flexibility of the rules, the arrangement of the timber elements was calculated through Random-Selection option of Grasshopper. The code developer (IA1) states:

Table 2
Illustrations of Team Leader's and Code Developer's sketches

	Team Leader's sketches illustrations		Code Developer's sketches illustrations
A#1	Section of the building: showing the wooden elements arch effect	B#1	1.5 m; Partial section of the building: showing the wooden elements arch effect and size
A#2	Perspective sketch of the wooeden elements: -variation of the elements	B#2	A1 A A A; Creating variatons for the wooden elements; B pattern; B1 B2 B3; Corner; B1: with small radious; B2: with big radious; B3: flat pieces
	100; 4; A wooden element sectional drawing: - top lenght - bottom lenght	B#3	150 cm; 20 cm; 1.5 m; 30x10 cm; A wooden element sectional drawing: - top lenght (revised) - bottom lenght (revised) - thickness -arch bending point
		B#4	A1 B1; 1 m; Array of the wooden elements; A1; A2-Random in Grasshopper; B1-Small Radious; B2- Large Radious; B3- getting thin but continous
		B#5	A1; Array of the wooden elements/ corner position; A1: Corner

00:03:25 IA1: .... *We also had another situation: randomness. Because of every façade has a different azimuth, for instance we said lets have much more random percentage of elements. We had 50% here, I mean once in two meters, right in the middle of the elements, we had Random in Grasshopper. We applied that in every floor. In this façade, we had 30% because; this façade has sunlight more than the others...*

The rules about the arrangement of the timber elements on the façade, the dimensions of each part, were calculated on paper and sketched, later all the data about the rules were entered into the digital tool (figure 5). On the base of the calculations on paper, IA1 calculated arrangement and percentage of the elements. Random option of Grasshopper calculated possible alternatives. After developing limited alternatives, the team leader decided to develop the façade shown in figures 5 and 6.

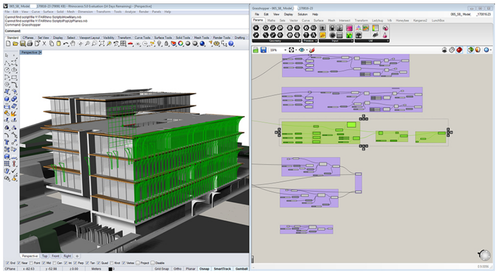


Figure 5
Application of the wooden elements of the façade on Grasshopper by Ihe code developer

Figure 6
The last version of the façade

DISCUSSIONS AND CONCLUSIONS

This research establishes the significance of design exploration by tools in the sense of knowledge propagation between the design object and the designer/s in the design process. The emerging question in this research that computational design tools are explorative as much as conventional sketching or the defined algorithm limits the variety of design solutions. Our claim is that in design process which compromises exploration by sketching and computation may turn into a hybrid practice that complement and feed each other. In both of the competition processes, sketching was one of the main tools that were used in the system as an information propagation tool among the team participants other than computational design tools and other software. Moreover, sketching was also a mediator tool between the code developer and the computational tool; first, the code developer was sketching for each problem then, inputting a set of the rules to the digital tool. Sketching also was an exploration tool in the competition A and manipulations of the shapes were done in order to apply exceptional situations such as disabled users' units design. However, when manipulations applied in a rule-based shape design, in this case we cannot talk about calculation in the design (Stiny, 2006). On the other hand, in the competition B, the code developer was aware of the exceptional situations for the rule, such as wooden elements' array on the corner of the building. However, in the beginning of the algorithm creation, the code developer applied the solution of every situation in the rule.Yu et al. (2015) claim computational design is a dynamic and rule-based process. Computational thinking relies on setting and organizing the rules for development of forms but, designer does not obviously define final form (Poulsgaard and Malafouris, 2013). In competition B, the code developer while defining the parameters including exceptional situations may not be aware of the final form. In competitional design, final design solution does not need to be precise, the random results of computational design can solve a non-fundamental formal design problem. However, in the design process, the interruption of the seeing-drawing-seeing process between designer and design object can lead to an end design arbitrarily.In the design process of the competitions, even though all the team participants were knowledgeable about computational design software, only expert participants took the responsibilities. Shifts in the roles were possible within team participants in the design process but the use of computational tools made it harder for teammates to shift roles. It seems that computational expertise requires a more specialized expertise. While looking for solutions to the architectural design problem, the use of computational design tools by two of the team participants might have influenced the architectural design approach. Moreover, architects who had knowledge of computational design technologies did not use the computational tools in the design process, they applied the computational design approach which they had internalized while using non-digital tools as rule-based sketches to find a solution to the design problem. Moreover, computational tools enable creating multiple alternatives, which one might think would be an opportunity in creating multiple alternatives, but code developers were more concerned with streamlining the process because of time constraints.

REFERENCES

Bundy, A. 2007, 'Computational Thinking is Pervasive', *Journal of Scientific and Practical Computing*, 1(2), pp. 67-69

Carpo, M. 2011, *The Alphabet and The Algorithm*, MIT Press, Cambridge,MA

Cinici, S., Akipek, F. and Yazar, T. 2008, 'Computational Design, Parametric Modelling and Architectural Education', *ARKİTEKT*, 518, pp. 16-23
Creswell, J 2007, *Qualitative Inquiry and Research Design: Choosing Among Five Approaches*, Sage, Thousand Oaks, CA
Cross, N 2006, 'Understanding Design Cognition', in Cross, N (eds) 2006, *Designerly Ways of Knowing*, Springer, pp. 77-93
Cuff, D 1992, *Architecture: The Story of Practice*, MIT Press, Cambridge, Massachusetts
Emerson, R.M., Fretz, R.I. and Shaw, L.L. 1995, *Writing Ethnographic Fieldnotes*, The University Of Chicago Press, Chicago
Hutchins, E. 1995, *Cognition in the WİId*, MIT press, Cambridge
Hutchins, E. 2014, 'The cultural ecosystem of human cognition', *Philosophical Psychology*, 27(1), pp. 34-49
Hutchins, E. 2006, 'The distributed cognition perspective on human interaction', in Enfield, N and Levinson, S (eds) 2006, *Roots of Human Sociality: Culture, Cognition and Interaction*, Bloomsbury Academic, Oxford, pp. 375-398
Hutchins, E. 1991, 'The social organization of distributed cognition', in Resnick, L, B, L, John, M and Teasley, S (eds) 1991, *Perspectives on Socially Shared Cognition*, American Psychological Association, pp. 283-307
Hutchins, E. 2004, 'Distributed Cognition', in Smelsner, J and Baltes, PB (eds) 2004, *International Encyclopedia of the Social & Behavioral Sciences*, Elseiver, Oxford, pp. 2068-2072
Kirsh, D. 2008, 'Distributed cognition: A methodological note', in Dror, IE and Harnad, S (eds) 2008, *Cognition Distributed: How cognitive technology extends our minds*, John Benjamins Publishing, Amsterdam
Lyon, E. 2005, 'Autopoiesis and Digital Design Theory: CAD Systems as Cognitive Instruments', *International Journal of Architectural Computing*, 3 , pp. 317-334
Lyon, E. 2011, 'Emergence and Convergence of Knowledge in Building Production: Knowledge-Based Design and Digital Manufacturing', in Kocatürk, T and Medjdoub, B (eds) 2011, *Distributed Intelligence in Design*, Wiley-Blackwell, pp. 71-98
Menges, A and Ahlquist, S 2011, *Computational Design Thinking: Computation Design Thinking*, AD Reader, London
Norman, DA 1991, 'Cognitive artifacts', in John, MC (eds) 1991, *Designing interaction*, Cambridge University Press, pp. 17-38
Olsen, C. and Namara, S. 2014, *Collaborations in Architecture and Engineering*, Taylor & Francis, NY
Pask, G. 1962 'The Conception of a Shape and the Evolution of a Design', *The conference on design methods*, pp. 153-167
Poulsgaard, KS and Malafouris, L 2013, 'Models, Mathematics and Materials in Digital Architecture', in Cowley, S and Vallée-Tourangeau, F (eds) 2013, *Cognition Beyond the Brain: Computation, Interactivity and Human Artifice*, Springer London
Schön, D. 1991, *The Reflective Practitioner: How Professionals Think in Action*, Ashgate Publishing Limited, Great Britain
Sommer, B and Sommer, R 1997, 'Interviews', in Sommer, B and Sommer, R (eds) 1997, *A Practical Guide to Behavioral Research: Tools and Techniques*, Oxford University Press, New York
Stiny, G 2006, *Shape: Talking about Seeing and Doing*, MIT Press, Cambridge
Stiny, G and Gün, OY 2012, 'An Open Conversation With George Stiny About Calculating and Design', *Dosya: Computational Design*, 29, pp. 6-11
Strauss, A and Corbin, J 1990, *Basics of qualitative research: Grounded theory procedures and techniques.*, Sage, Newbury Park, CA
Suchman, LA 1987, *Plans and situated actions: the problem of human-machine communication*, Cambridge University Press
Terzidis, K 2006, *Algorithmic Architecture*, Elsevier, Oxford-UK
Visser, W. 2010, 'Scho⊠n: Design as a reflective practice', *Art + Design Psychology*, 2, pp. 21-25
Yu, R., Gero, J. and Gu, N. 2015, 'Architects', *International Journal of Architectural Computing*, 13(1), pp. 83-101
Özkar, M. 2005 'Lesson 1 in Design Computing Does not Have to be with Computers', *Proceedings of the 23rd eCAADe Conference*, pp. 21-24