April. 2016. Vol. 8. No. 03 ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

COMPUTATION AS A SKILL FOR DEVELOPING DIGITAL DESIGN TOOLS IN ARCHITECTURE EDUCATION

[1]SERDAR AŞUT, [2]ARZU ERDEM

[1]PhD Candidate, Department of Informatics, Istanbul Technical University
[2]Prof. Dr., Department of Architecture, Istanbul Technical University
Email: [1]serdarasut@gmail.com , [2]erdemarz@gmail.com

ABSTRACT

This paper discusses the relationships between design cognition and tool use, particularly in architecture education. It claims that the rise in the use of Computer Aided Design (CAD) applications requires the reformation of curricula in order to enable the students to comprehend and customize the nature of CAD applications. In this sense, computation is proposed as a necessary core skill which needs to be integrated into the education. Thus, the students will be able to develop personal digital design tools which can capture their own unique ideations.

Keywords: Architecture Education, Design Tools, Computer Aided Design, Computation, Design Cognition

1. INTRODUCTION

The aim of this papers is to put forward a critical discourse within architecture education and claim that it is necessary to reform the teaching by introducing computational thinking as a core skill for architects. In this regard, computational thinking is coined as the set of skills which are necessary to develop architects' own personalized digital design tools. The importance of personalization of the tools is described through the relations between design cognition and tool use, particularly within teaching.

Design is profoundly engaged with tool use. Design education is mostly about teaching how to think through engagements with the tools. Advanced digital design tools, namely Computer Aided Design (CAD) applications, have been facilitating new methods and theories in the last few decades. The practices of architecture are mostly led by these new tools, methods and theories being introduced. Various CAD skills become primary requirements for architects as seen in every single job advertisement. Along with the practicalities which are offered by the CAD applications, a tool-driven practice involves problematics related with design cognition. It is necessary to review these problematics and develop critical approaches in order to introduce the necessary knowledge and skills into teaching. However, the integration of CAD into architecture education is far behind the speed of its integration into the practice. This paper discusses why it is important to enable this integration by providing a conceptual road map.

2. THE ELUSIVE STUDIO CURRICULUM AS THE CASE STUDY

Bamberger and Schön (1983) describes design as a reflective conversation with materials and define the "talking back"s in design process. Design, in this form, is a reflection-in-action (Schön, 1984). Following this form of reflections and actions, design studio of architecture schools are environments where students can experience design process through reflective conversations. The reflector actors of this conversation are several; principally the student himself/herself, other students in the studio, the studio tutor, and design tools. Within the discussion in this article, the role of design tools in the reflective conversation constitutes the main focus.

Design studio is an environment in which the student learns through practicing designerly behaviors. Here the tutor does not literally "teach" design. Rather, tutor's role is to provide the environment of evaluation of the expected designerly behavior. As defined by Yürekli and

ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

Yürekli (2004), rather than design teaching, design evaluation is the main concern of the studio. Here, the student is expected to investigate the design problem space and develop ideas, tools and methods for generating the solution space. Project based and problem oriented desk critics, juries and group discussions are there in order to evaluate these investigations and ideas. Therefore, design studio is a platform of evaluations on hands-on practices.

Norman's (1990) definition of the seven stages of action, which is presented in ***Table 1***, can naturally relate with the actions which are taken at the studio as an environment for continuous evaluation. Here, the seven stages that are taken sequentially are in a continuous loop. However, there exist continuous re-executions of the previous actions instead of a linear flow. For example, according to the evaluation of the outcome, designer may need to review and update his/her design decisions, tools or methods. And all of these actions follow the reflections which occur during the conversations between the designer and his/her tools, which enable him/her to evaluate his/her very own implicit decisions. Designers need qualitative and imprecise external visualizations to interact with their mental images and they need to be in continuous interaction with them to make design decisions (Dorta, Pérez, & Lesage, 2008).

Schön (1984) emphasizes the importance of representations in design process by defining architectural design as a making of representations of things to be built. Following this definition, it is possible to classify representation -of the problem, data, decisions, form, space, etc- as an important genre of design tools. What a student experiences in the studio is how to have a continuous conversation with design representations and to execute actions according to the reflections. Here, reflection has two intentions; the maker's spontaneous reflection and the reflection of the materials back to the maker (Bamberger and Schön, 1983).

Table 1. The seven stages of action (Norman, 1990).

#	The Stage of Action	Designer's Actions
1	Forming the Goal	Investigate the problem, determine what to do.
2	Forming the Intention	Determine how to do and the tools and methods needed.
3	Specifying the Action	Determine the design decisions.
4	Executing the Action	Execute the decisions through design tools and methods.
5	Perceiving the State of the World	Review the process and current state.
6	Interpreting the State of the World	Criticize the process and current state.
7	Evaluating the Outcome	Evaluate the outcome and update the decisions.

What is experienced at the studio is how to communicate with design representations. Representation is the language of designing. This is a language which is used to analyze, depict and evaluate the design. Students learn how to use this language through hands-on practices. However, a pedagogical approach which aims at teaching mere the universal methods of using a particular representational language, such as technical drawings, is not sufficient for educating a designer. Even though such universal methods are necessary, they do not constitute the origins of designerly behaviors. Therefore, such an approach should not be the main and only concern of design education. On the other hand, learning what is universally common in practice is comparatively easier to learn, not only at school but also during the professional practice. Instead, the curricula need to be able to encourage each student to generate his/her own language and gain personal, unique and profound designerly behaviors. To this end, it is necessary to allow, enable and encourage the students to develop personal means of representing their own ideations.

ISSN2305-8269
International Journal of Engineering and Applied Sciences

www.eaas-journal.org

3. DESIGN TOOL DESIGN

Chard (2005), referring to drawing instruments, discusses the notion of transparency between a thought and its projection on to a picture plane and mentions that the facility to draw and the capability of the tools must have some sympathy with the intention. Related to this notion, one can argue that the qualifications of the tool are significant in terms of processing the intentions. Tool, as being an interface, is able to implicate its content to the problem and solution spaces and therefore it is capable of transforming the intentions of the user. One can argue that the more the interface (or more interfaces) takes part in the activity, the more the user will lose originality. The solution to this is through user's capability of using the tools as his/her extension of the body. Kotnik (2010), while discussing the interaction of the body with its environment, mentions that every tool mediates this interaction because of its specific usage, thereby influencing the perception of the user and his way of thinking. A case where the way of thinking is in its purest -or as purest as possible- form is sketching.

The importance of sketching in design process is well defined by Do (2002) as being a natural way for people to explain and understand complex ideas and to perform visual and spatial reasoning. This is because sketching is an exploratory and generative process which does not consider precision. Even though sketching is mostly used in the early phases of design, it is a procedure that can be introduced in any phase for various purposes some of which are decision making, form finding, process and outcome evaluation, perception or data input. Distinct from other types of representational media, sketches, which can be worked on 2D or 3D drawings, diagrams or physical models, are free from constraints. This is mostly because sketches, by their very nature, do not necessarily have to follow any universal language or method. On the contrary, they are based on a personal and spontaneous language for each designer himself/herself.

Developing a personal, unique and spontaneous design language depends on the ability of manipulating and customizing the tools and tool spaces. This manipulation starts during the investigation of the problem space. Thinking about the processes of developing design ideas is an integral part of the process of design itself (Füssler, 2008). Design process starts with investigating the design problem and certainly includes determining the tools and methods to be used. A problem exists when a goal must be achieved and the solution is not immediately obvious, and problem solving often involves attempting different ways to solve the problem (Dunbar, 1999). This is indeed the usual case for a designer since design problems are ill-defined. As Dunbar (1999) puts forward, one of the key elements in solving a problem is finding a good way of representing the problem. Therefore, there is an indefinite space of problem representations in which the designer needs to generate his/her own personal representational sets to be utilized for the problem discovery and solution formulation. Schön (1983) points out that the dialectic between problem discovery and solution formulation is key to design education.

Dearden (2006) introduces the concept of "material utterance", which includes the making or modification of material artifacts in conversation. The domain of making and modification bases upon the qualities of the speaker, audience, material and genre. For expecting a student to develop a personal design language, tool making and method modification should be considered as a design problem as well. Reflexive behavior that constraints itself with what is handy within tools and methods is not what a designer would need. Such a behavior would obviously generate optimized design solutions as tool and method make up the process of problem solving. On the other hand, dealing with the problematic of tools and methods helps the designer to gain a more profound understanding of the design problem. Problem representation is a critical tool which influences problem solving. The development of design ideas can be enhanced by designing with self-designed tools (Füssler, 2008). Therefore, an approach of tool design becomes a crucial challenge for design education.

4. COMPUTATION FOR TOOL DESIGN

Thinking within digital tools, the existence of a personal, unique and spontaneous language becomes questionable. Within computer-aided design (CAD) practices, the definition and context of designerly intelligence sometimes become ambiguous. Because, CAD applications are usually able to constrain the design methods and are usually very hard to customize. Even if it is possible to customize the tool, it practically becomes difficult to realize such an intention as long as the understanding of the nature of the tool is missing – particularly from the point of view of a student. Dorta et al. (2008), while discussing the relation between computers and ideations, mention that the mediation of menus and commands is distracting and often carries so many pre-conceptions about how the design process should be that they affect decision-making. This is because tool, in any case, can deterministically direct the way designer thinks and behaves. This determinism becomes more apparent within CAD applications, because they are not only physical but also psychological tools, which are fairly defined by Vygotsky (1986). The risk is that; the way each application is constructed can weaken the perceptual interpretation space of the student.

In order to build a sympathy between the unique interpretation space of the user and the capabilities of a CAD application, one needs to comprehend the very nature and the origins of the CAD systems. This may only become possible through a specific cognitive state where the term computation is descriptive. This cognitive state includes a particular way of thinking which is systemic, holistic and semantic, i.e. computational, rather than being mere depictive. Just learning to operate a CAD tool or a programming language or gaining skills in using the computer as an electronic drawing board does not necessarily make an architect think computationally about design. It is precisely the degree of awareness of the computational background and its intentional use that can be seen as the defining characteristics of digital design and at the same time as a possibility to mark the threshold between digital and non-digital design (Kotnik, 2010).

One of the most important aspects of computation is that it is mostly engaged with the origins that lay beneath digital media as a way of thinking. Such a way of thinking differs from traditional ways of design thinking where representational tools are there principally for depictions. Within depictive thinking, which is fundamentally artistic and spatial, the prior focus is to generate the forms and shapes that will afterward generate the spaces. However, computational thinking requires modeling the problem, determining the decisions, collecting the data and building up the relations between them. These particularly require mathematical and semantic knowledge and skills in addition to artistic and spatial ones. The computational designer mainly needs the skills in analytical geometry because as Kotnik (2010) mentions, it enables the functioning of CAD-software through the translation of geometric operations into computational functions.

As Cross (2001) argues, design discipline introduces not only new tools for old methods, but also new methods. The emergence and wide spread of CAD applications indicate such a necessity for computational methods. Lack of these methods will sustain designers' inability against digital media and let them to practice as mere CAD operators. However, computational models provide the knowledge and skills that are necessary for creating one's own unique tools and customizing the existing ones. By using these skills and knowledge, designers will be able to create their own personal ways of doing.

Computational thinking is crucial for both being able to realize a unique design process instead of an optimized one, and generating educational models that help students to comprehend the origins of design which is engaged with tool use. In order to provide the latter, the understanding of design teaching as a practice of introducing the tools should improve to a new approach which encourages tool design. Tool design, in this manner, refers to the endeavor to capture the elusive origins of each design.

5. CONCLUSION

> *"...do not think of curriculum as a 'thing,' as a syllabus or a course of study. Instead, think of it as a symbolic, material, and human environment that is ongoingly reconstructed. This process of design involves not only the technical, but the aesthetic, ethical, and political if it is to be fully responsive at both the social and personal levels (Apple, 2000)."*

Design education needs to be reformed towards capturing the skills for computational thinking in order to help and encourage the future practitioners to develop their own personalized digital design tools. To this end, a dramatic change in the curricula is necessary. This change will include introducing new courses as well as transforming the existing ones towards the objective.

Courses which focus on mathematical, analytical and semantical thinking plays a crucial role in constructing the proposed curricula. However commonly, like mentioned by Özcan and Akarun (2001), mathematics courses often are offered in the freshman year, as part of a common scientific core and the students do not consider the subject as a part of their creative development, and regard it as a boring compulsory course; a monotonous repetition of high school subjects. It is crucial to change this attitude and to equip the curricula with such courses which provide the principal knowledge that are necessary for understanding the very nature of digital practices and therefore should be considered as an important training which supports the studio. Thus, the student will be able to comprehend what is running behind the computer screen and not only masterly use it but also to manipulate it.

On the other hand, as computational workflows require particular ways of thinking, introducing computation courses will not be enough to provide the necessary pedagogical basis. Rather than a minor field of study, it needs to be the driving approach of the teaching practiced in the studio. Our experiences indicate that it may even be more efficient not to have separate computation courses at all, since the students usually have a tendency to consider separate courses as minor aspects which do not have practical use in design. However, introducing computational thinking into the studio will more profoundly help the students to consider it as a natural and principal way of doing.

The lack of such a pedagogical approach and reformed curriculum will graduate designers who are hardly more than CAD practitioners. This refers to a future of design in which most of the designers basically repeat what is valid in the market or what is trendy; while what is valid and trendy is defined by the software vendors.

6. REFERENCES

1. Apple, M. W. (2000). *Official Knowledge: Democratic Education in a Conservative Age*. Psychology Press.
2. Bamberger, J., & Schön, D. A. (1983). Learning as Reflective Conversation with Materials: Notes from Work in Progress. *Art Education*, *36*(2), 68–73.
3. Chard, N. (2005). Drawing Instruments. *Architectural Design*, *75*(4), 22–29.
4. Cross, N. (2001). Designerly Ways of Knowing: Design Discipline Versus Design Science. *Design Studies*, *3*(7), 49–55.
5. Dearden, A. (2006). Design as conversation with digital materials. *Design Studies*, *27*(3), 399–421. Retrieved from http://shura.shu.ac.uk/16/1/fulltext.pdf
6. Do, E. Y. (2002). Drawing marks, acts and reacts: Toward a computational sketching interface for architectural design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *16*(3), 149–171.
7. Dorta, T., Pérez, E., & Lesage, A. (2008). The Ideation Gap: Hybrid tools, Design Flow and Practice. *Design Studies*, *28*(2), 121–141.
8. Dunbar, K. (1999). Problem Solving. In W. BECHTEL & G. GRAHAM (Eds.), *A Companion to Cognitive Science* (pp. 289–298). Blackwell Publishing.
9. Füssler, U. (2008). Design by Tool Design. In *Proceedings of the Advances in Architectural Geometry Conference* (pp. 37–40). Vienna.
10. Kotnik, T. (2010). Digital Architectural Design as Exploration of Computable Functions. *International Journal of Architectural Computing*, *8*(1), 1–16.
11. Norman, D. A. (1990). *The Design of Everyday Things*. New York: Basic Books.
12. Özcan, O., & Akarun, L. (2001). Mathematics and Design Education. *Design Issues*, *17*(3), 26–34.
13. Schön, D. A. (1984). The Architectural Studio as an Exemplar of Education for Reflection-in-Action. *Journal of Architectural Education*, *38*(1), 2–9.
14. Vygotsky, L. (1986). *Thought and Language*. Boston: The MIT Press.
15. Yürekli, H., & Yürekli, F. (2004). Mimari Tasarım Eğitiminde Enformellik (Informality in Architectural Design Education). *İTÜ Dergisi/a*, *3*(1), 53–62.