Creative use of Architectural Precedents in Design Education: A Framework for a Computational Model

Hakan Anay
Middle East Technical University, Department of Architecture, Ankara-Turkey

Present study primarily aims to outline a theoretical framework for developing a computational model towards the creative use of architectural precedents in architectural design education. It departs from a short summary of the critical/formalist approach as the model that we adopt for the studio education, and goes on with a discussion on paradigms or precedents as the containers of knowledge and as the primary focus of the studio model as important elements of architectural design education. Prior to conclusion the study sets three important problem areas concerning the model, from the computational point of view, and investigates them while trying to outline a framework towards a computational model. As a conclusion it proposes two main areas that we can utilize computers for the creative use of architectural precedents in architectural design education, and discusses a possible studio implementation of the proposed framework as the next stage of the study.

Keywords: *Architectural Design Theory; Design Methodology; Architectural Design Education; Computer Aided Design*

Introduction

The main aim of the present study is to outline a theoretical framework for developing a computational model towards the creative use of architectural precedents in architectural design education. This theoretical framework will not only provide a basis for the development of models for architectural design studio, but also it will make possible the development of strategies and computational tools related with these models.

The design studio is an essential organization in architectural education where the students of architecture gain foundational design knowledge, skills in modes of making, learn thinking architecturally, and solving architectural problems. However, what we call design studio is an umbrella term which covers a broad range of studio implementations/models that follow different studio traditions and approaches. Through their tradition, these approaches/models are established upon various epistemological foundations. A framework or model targeting the creative use of computers and computation in architectural design education must be established upon the same epistemology, and be primarily capable of fulfilling the needs of the tradition that we follow, or the approach that we adopt for the studio education.

The present study takes the critical/formalist approach as its basis for outlining the theoretical framework. It is beyond the scope of this paper is to make an in depth discussion of the critical/formalist

approach. However, a short summary on the methodology is inevitable. Methodologically, the approach has two overlapping operational main layers: the first layer concerns the critical/formalist analysis of the paradigms or precedents and the second layer concerns the constructive/formalist transformation.

The first layer concerns the investigation of the existing architectural works themselves, since a holistic, critical, and in depth examination of the works of architecture and a comprehensive understanding of them (as paradigms or precedents) is one of the important sources for learning how to design, and for the production of architectural knowledge. The "knowledge" of these paradigms or precedents is a prerequisite for solving new problems. As such, we do not start from a tabula rasa but build our own work upon what was already known and what was already achieved so far.

The second layer concerns the evolutionary/creative search process. The evolutionary/creative search process begins with an "approximate configuration" or "solution in principle" as a starting point. "Solution in principle" can be derived from the precedents (exemplars or paradigms), (re)conceptualized, (re)configured, or (re)combined (or "mutated") for the new conditions and for the new problems. Once the starting point is established, the process continues by making, reading and criticism (evaluation), and remaking cycles (Anay, 2005). This process consists of indeterministic alternative creation (by blind search and mutation) and critical selection of the alternatives.

Paradigms as the Containers of Knowledge

The concept of "paradigm" has a particular importance for the present purposes. The term has two related but distinct meanings. In its first conception, paradigm equals to the term "disciplinary matrix" which is a global entity that embraces "all the shared commitments of a scientific group…" (Kuhn, 1977). In its second conception, paradigm refers to a particular sort of commitment which is a subset of the "disciplinary matrix" (Kuhn, 1977). Although there are many of such commitments, Kuhn emphasizes three of them as primarily important for the central cognitive operation of the group. These three commitments are "symbolic generalizations," "models" and "exemplars" among which the last one is particularly important for the present study. In this last conception, paradigms equal to exemplars, which can be described as "concrete problems with their solutions" (Kuhn, 1977). From the educational point of view, exemplars have a pedagogical importance. Kuhn states that "…students of physics regularly report that they have read through a chapter of their text, understood it perfectly, but nonetheless had difficulty solving the problems at the end of the chapter. Almost invariably their difficulty is in setting up the appropriate equations, in relating the words and examples given in the text to the particular problems they are asked to solve. Ordinarily, also, those difficulties dissolve in the same way. The student discovers a way to see his problem as like a problem he has already encountered. Once that likeness and analogy has been seen, only manipulative difficulties remain" (1977).

We may say that prior to any attempt for solving a new problem one needs the knowledge of already known solutions to problems of similar type. In fact this type of thinking is essential to all learning and problem solving, and eminently applicable to architectural education and architectural problem solving. Each architectural work can be seen as a solution to a particular problem or set of problems, and in this sense can be interpreted as an exemplar or paradigm. But it must be underlined that, as an artifact to be "known," an architectural work must be subject to a careful and in depth examination and an interpretative and critical analysis. Only as such it can serve as an exemplar or paradigm.

Towards a (Critical/Formalist) Computational Model

When confronted with an architectural problem, prior knowledge of "good" paradigms provides an insuperable foundation and required preconceptions for developing an "approximate configuration" or "solution in principle" as a starting point for architectural investigation. Beginning the investigation with the knowledge of existing solutions and an "approximate configuration" or "solution in principle" helps us to understand the problem, organize the constraints, the program and the requirements and put them in correct places to test and develop the initial proposal (Anay 2006).

But the analysis and understanding of the paradigms or precedents and use and adaptation of them for new designs is a challenge for the studio setting since it requires background (or contextual) knowledge which the students of architecture yet do not possess. The potentialities of the computers and computation can be used for supporting design education primarily at this point and further for the evolutionary/creative search process. However, a comprehensive investigation of this issue is beyond the limitations of the present paper. Instead, from the computational point of view, this study selects three essential problem areas to discuss.

First question is what exists in the paradigms or precedents themselves to be transferred to new works? Architecture is open to "mindless" imitation or repetition of the organizational schemas, forms and programs taken from the historical precedents. Contrarily, an architectural object must be taken as an "open work" which is subject to almost infinite reading and interpretation, for understanding it, deriving design knowledge from it, and to be used as a starting point for architectural investigation. To use Karl Popper's terminology, every product of human mind (such as symphonies, pictures, works of architecture) possesses a "thought content" inherent in it (1978). This is the "objective" content that resides in the object independent from our experience of it or its creator's intentions. However, thought content is not the essence of the work, since a work may yield "thought contents" which nobody has thought so far (including its creator), and perhaps nobody will ever think about. The essential point is, first, one must be able to read and understand a work, and second, one must have the ability to decipher and grasp its "thought content."

Directly related with the first, second question is how we can represent the works and the "thought content" contained in them computationally, so that we can understand and interpret the existing works and also be able to build our work upon what was already known with the help of the computers? It must be underlined that even if they were prepared by the help of the computer software, the technical drawings, or the visual images of an architectural work have a little use for our purpose. We need a system that will serve for making schematic and conceptual and also holistic computational representations (or models) of the works. Furthermore such a system must be adaptable not only to the computers, and computational problem solving but also architects, architectural conventions and real-life architectural problem solving. Such a system is roughly formulated by Marvin Minsky in the field of artificial intelligence. In his influential essay "A Framework for Representing Knowledge," Minsky departs from describing a theory of artificial intelligence and thinking (1974). The essence of the theory is given as follows: "When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a Frame. This is a remembered framework to be adapted to fit reality by changing details as necessary" (Minsky, 1974). The general idea of frame (and its modified version frame-system) is essentially same as the idea of "paradigm" since it has its epistemological roots in the "paradigms" of Kuhn. For Minsky, a "frame" is a data structure for representing a typical (or already known) situation such as a solution to a particular problem or an object. Typically, several kinds of information can be attached to a frame. Minsky argues

that a frame is typically "a network of nodes and relations," where the "'top levels' of a frame are fixed, and represent things that are always true about the supposed situation." There are many slots or terminals at the lower levels that must be filled by specific instances or data. Each slot or terminal can specify conditions its assignments meet where these assignments are usually sub-frames. The slots can be assigned "to be a person, an object of sufficient value, or a pointer to a sub-frame of a certain type." These terminals are already filled with default values which can be interpreted as assumptions or expectations subject to displacement with new items that fit better to the confronted situation or problem (Minsky, 1974).

However, in its present conception, the idea of frame must be reconsidered for the requirements of the critical/formalist (computational) model. Typically, a frame is a holistic and complex system of nodes and relations. Beyond these characteristics, a frame must be an indeterministic system, having "plastic" relations between its nodes and sub-frames. It must be open to outer affects and interventions, responsive to environmental influences, and in this sense adaptable and evolvable to new conditions and requirements while keeping its integrity and main organizational scheme intact. However, it must be underlined that this is only a rough outline of a frame-system and this issue demands a comprehensive discussion and investigation.

Final question is how can we use "thought contents" (computationally) for adapting them for new problems or for the creation of new works? The basic method of computational problem solving is algorithmic method. This method is not suitable for architectural problems since it is only good for solving problems that have a well-defined starting point, a feasible algorithmic definition and a well-defined solution. So the method of search must be heuristic, pertaining to trial-and-error. In heuristic search, trial-and-error process does not depart from a tabula rasa, or from any random conjectural point. Typically, heuristic search utilizes already existing, contextual knowledge (Rechtin, 1991), or knowledge of past solutions, and proceeds in this way. This is almost inevitable and useful from the educational point of view if we reconsider the discussions on paradigms earlier in this paper. One of the basics of architectural design education must be to make students learn what was already known by analysing the precedents or paradigms, and from their mistakes as well as from their "good" moves while trying to develop their works upon the precedents. With this respect, heuristic search may provide a controlled process and environment and help them gaining already existing or contextual knowledge. However, at this point it must be noted that as far as the creativity is concerned, this interpretation of heuristic search is only partially true. Typically, every heuristic search begins and proceeds with already existing, contextual knowledge. But creativity sets out where we try to go beyond what was already known and what was paradigmatic.

These questions and the answers given to them lead to us to another question: how to bring these together for developing a model for design education? As a basis for the conclusion a summary of a study from the field of artificial intelligence that can be used analogically as a starting point for this purpose. In 1955, Arthur Samuel wrote a computer program that uses heuristic methods for learning playing checkers (Samuel, 1959). The program played checkers by making trial moves and learned from its mistakes and good moves by evaluating these trials. It stored its experience as generalizations and later projected this experience to new games. But the program's excellent ability to learn was achieved as follows: While playing with itself for learning, the program "played" two identities. Its first identity called alpha, was a rapidly mutating pioneer, preferring to engage in new and risky moves, and its second identity, called beta was a conservative preferring to be tied to guaranteed and best moves achieved so far (Dennett, 1995). If a better move was found, the program either stored it as a successful move or replaced less successful move with the better one.

The tension between these two identities –that is the conservative and the pioneer- was essential, and made the learning (and creativity in play) possible, to the degree that the program even beat its creators. If we reformulate, the program on the one hand utilized already existing (contextual) knowledge, on the other it tried to change this knowledge to find new (and better) solutions.

A Tentative Conclusion

In "Of Clocks and Clouds," Popper argues that "… we use, and build, computers because they can do many things which we cannot do; just as I use a pen or pencil when I wish to tot up a sum I cannot do in my head" (1972) From this point of view, computers can be seen as devices for enhancing and improving human thinking. At the same time they are the enabling devices for externalizing our thoughts so that they can be criticized and developed. In architectural design education, we must utilize computers not as hermetically sealed intelligent design machines, but as intelligent devices for assisting human thinking and creativity, and as devices for helping learning and extending learning abilities. With these issues and with the arguments made earlier in this paper at mind, we may say that there are two main areas that we can utilize computers for the creative use of architectural precedents in architectural design education.

First, they may help students to represent the existing architectural works themselves in terms of frame-systems, and store these in a relational database. By the help of the computers these representations can be later retrieved, analysed, interpreted and adapted to be used in a heuristic search. If required, the computer must provide an interface between the frame-systems, the database and the conventions of architectural design.

Second, the computer may play the role of an intelligent design partner. While "knowing" the good paradigms or exemplars, a computational tool may derive and set rules and constraints and actively engage in the design process by proposing already successful solutions (playing the role of the beta in Samuel's case). The students have the responsibility to analyse and understand what was proposed by the computer and to try changing and adapting these paradigms (playing the role of the alpha) by creating multiple alternatives by blind (or intentional) trials. The computational tool may further help students to learn evaluating their blind (or intentional) trials and for selecting and eliminating alternatives. The whole process, the set of tentative solutions and the relations between them can be stored as an evolutionary process, where at any time any point can be accessed, rewound, replayed, modified, in short, under full control. Once a solution is achieved, it may be stored with relation to its precedents or by replacing less successful solution.

The next stage of the present study will be to (re)formulate a studio model based on the proposed framework, which will target the second year architectural design studio. It must be noted that there is no ready made software that suits the needs of the proposed model. This is a problem to be addressed. However, preparation of fully operational software from scratch prior to studio implementation is impossible to carry out. Rather, such software is being developed by coding small modules to be tested in the studio setting for their suitability for various tasks and to be revised by using the feedback. This requires a self sufficient studio setting which has its own objectives and goals that is capable of running successfully without the aid of the computational tools, but flexible enough to allow the integration of these tools to various tasks and stages when needed. (Also see forthcoming "World Three Hermeneutics: In search for a (better) Model of Architectural Design Studio Teaching/Learning")

References

Anay H.: 2005, Towards a Reconsideration of Computer Modeling in the Idea-Creation and Development Stages of the Architectural Design Process in New

Design Paradigms Conference Proceedings of IASDR, 2005
Anay H.: 2006, On the Relevance of Karl Popper's Epistemology for Architectural Education in Built Environment and Information Technologies Proceedings of CIB PGRC 2006
Campbell D.T.: 1974, Evolutionary Epistemology in P.A. Schilpp (ed.), The Philosophy of Karl Popper, Open Court Publishing, La Salle Illinois, pp. 413-463
Dennett D.C.: 1995, The Computer that Learned to Play Checkers in Darwin's Dangerous Idea, The Penguin Press, pp. 207-211
Kuhn T.: 1977, Second Thoughts on Paradigms in The Essential Tension, The University of Chicago Press, Chicago and London, pp.293-319.
Minsky M.: 1974, A Framework for Representing Knowledge, MIT-AI Laboratory Memo 306
Popper R.K.: 1972, Of Clocks and Clouds in Objective Knowledge, Oxford University Press, Oxford
Popper R.K.: 1978, Three Worlds, The Tanner Lecture on Human Values delivered at the University of Michigan
Rechtin E.: 1991, Systems Architecting, Prentice Hall, New Jersey
Samuel, A.L.: 1959, Some Studies in Machine Learning Using the Game of Checkers, IBM Journal of Research and Development, Vol.3. No.3