DIGITAL VS. TRADITIONAL DESIGN PROCESS

Igor Svetel[1]
Innovation Center, Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11000 Belgrade, isvetel@mas.bg.ac.rs

Tatjana Kosić
Innovation Center, Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11000 Belgrade, tkosic@mas.bg.ac.rs

Milica Pejanović
Faculty of Architecture, University of Belgrade, Bulevar Kralja Aleksandra 73/II, pmilical@arh.bg.ac.rs

ABSTRACT

The design process is a theme that has long been a preoccupation for architects. Today, when digital technologies play irreplaceable part in the architectural design it would be expected that the design process had improved in relation to traditional principles. But starting from sketching, through 3D modelling, to BIM applications, design process is based on the traditional principles that architects applied before the emergence of digital technologies. Digital sketching is focused more on the reproduction of the sketch appearance through the digital tracing of the hand movement and imitation of the look of the pen on paper than on the study of the process of designing an architectural object through sketching. The creation of 3D models is also aimed at generating models that are used as a substitute for traditional mock-up or photorealistic visualization of the building. BIM models are primarily focused on the development of project documentation. The focus on imitation of traditional representations by the use of digital technologies also makes it difficult to use computer applications to simulate various aspects of the functioning of the building, such as energy consumption, natural lighting, ventilation, etc. The paper gives an overview of research in the field of modelling computer-aided architectural design process inspired by advances in system theory and artificial intelligence that was conducted during the 1980s and 1990s and compares that research with computational architectural design applications that are currently present on the market. The analysis shows that many important principles of architectural design by computer, developed at that time, are now forgotten. Today's applications allow only basic problem solving techniques and do not allow exploration and analysis of variants that represent the true nature of the architectural design process. Because of the inability of earlier computers to process a large amount of information, these systems were experimental, but the research conducted at that time briefly illustrated in this paper, shows that it is possible to develop a digital design process that is in line with what architects really do.

Keywords: digital design, design process, CAAD, model

[1] Corresponding author

INTRODUCTION

The design process is a theme that has long been a preoccupation for architects. Inspired by the oldest known written record on architecture, Vitruvius' "De architectura", architects saw the design process as a prescription of the steps leading to a good architecture. Only in the middle of the 20th century, inspired by the development of system theory and computer science, research into the systematic design process has begun, resulting in the seminal book "Design Methods" (Jones 1970) and the development of a new design research discipline. At that time design was seen as the process of devising "courses of action aimed at changing existing situations into preferred ones" (Simon 1969). The later developments in the field of artificial intelligence have influenced the emphasis on the role of human thinking and representation in the design process. Thus, the design process around the 1980s was seen as "the human mind activity that leads to the description of the desired change in the system of the built environment, as well as the description of the artefact, the means for achieving the change" (Petrović 1977) or as "interlocking processes of perceptions, cognition and notations" (Schön 1988). During the 1990s, the interest of scientific circles for the importance of society in the process of thinking influenced the theory of the design process, so design was seen as "partly … an exercise in propaganda, for it is through his work that a designer can most easily explore and simultaneously disseminate his ideas" (Lawson 1990).

Based on above mentioned theories, numerous computer systems have been developed that have implemented these principles. It is important to note that whole research into the systematic design process stared in order to overcome the problems with the traditional design process. Therefore, no computer system developed at the time attempted to simulate the traditional design process with the help of a computer but to create a completely new work environment that would allow a radically new digital design process. The capabilities of the computers from that period did not allow the manipulation of large amounts of data, nor the execution of demanding calculations necessary for complex simulations. Computer applications that implemented the design process theory were just experimental. It would be expected that with today's power of computers, these concepts, at that time experimental, will be fully implemented in commercial AEC applications. Unfortunately, very few of these principles can be found today in commercial applications, and previous research on the definition of a new digital design process has fallen into darkness. This paper aims to shed light on the forgotten research and draw attention to the still current problem of the digital design process.

PROBLEM SOLVING

The result of the first wave of systemic research in the design process was to see it as a problem-solving process. The process is presented as the tree like structure (Figure 1) in which each branching node corresponds to one problem state, and each branch corresponds to one rule that transforms the problem states, i.e. to one decision made by the designer. In order to explicitly enumerate possible solutions to the problem, it is necessary to "travel" through the search tree, namely, starting from a problematic situation and by using the rules, to make all the transformations necessary to find the solution. There are a number of problem solving methods. The simplest method is trial-and-error. It is based on the process of synthesizing a complete solution and checking whether it meets required conditions. Synthesis mechanism can be exhaustive or random. In the first case, the procedure systematically examines all potential solutions, while in the second one potential solutions are randomly created until the desired number of final solutions is obtained.

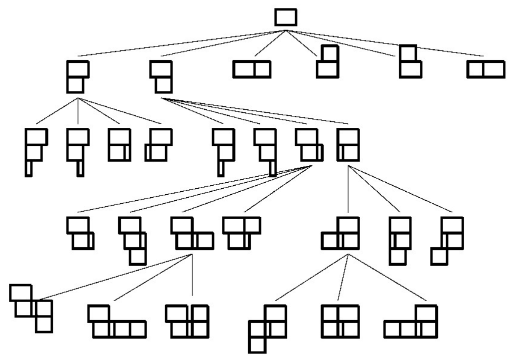

Figure 1: Example of the decision tree in generate-and-test process of layout synthesis. The solution is obtained through step by step process in which partial solutions are generated and evaluated against criteria. Only those partial solutions that satisfy criteria are further expanded toward final solution.

Generate-and-test is a frequently used heuristic method of solving problems. The principle of its work is based on the synthesis of the partial solution and checking whether such solution meets the given project conditions. Testing a partial solution has a dual role: 1) verifies that the partial solution corresponds to the constraints, and 2) determines the next design action. The process of selecting the next action can be more or less "intelligent" depending on the extent to which the information obtained during the synthesis is used to predict most promising next design step. The means-ends analysis is an extension of the generate-and-test method. Its three basic components are: 1) the set of predefined project actions (means), 2) the set of predefined project goals (ends), and 3) the set of decision rules. The basis of the method is the process of detecting the differences between the current state of the project and the objectives that the project should satisfy. When this difference is determined, the rules are applied to find an action that can eliminate existing difference. The planning method aims to reduce the search process for the solution by specifying in advance the sequence of project actions that lead from the current state to the target state of the project. The most commonly used planning method is the hierarchical decomposition process, which allows the complex problem to break into smaller parts, which can be solved independently of each other. The planning method is most often used in combination with other methods of problem solving, whereby planning is used to decompose problems into smaller entities (functional zones) and determine their mutual relations, while one of the above mentioned methods is used to synthesize variants of the partial solutions.

KNOWLEDGE BASED DESIGN

Very often it is almost impossible to define explicit procedures for the evaluation of partial solutions. As a result of this problem a new discipline of heuristic reasoning has been developed. Instead of using rigorous mathematical evaluation, heuristic reasoning uses simple rules based on experience like rule of thumb, educated guess, intuitive judgment, or common sense in order to find a satisficing solution instead of an optimal one. The ability of computer to serve as a unique medium for representing design knowledge and for carrying out design activities has inspired numerous researchers to develop knowledge based design systems (Coyne et al. 1990). These systems are most often based on two structures for computer based knowledge representation: rules and frames. The rules consist of a left-hand or IF part which determines the conditions under which the rule can be applied and the right or THEN sides determining actions to be executed if the left side of the rule is satisfied. Systems consisting of a working memory containing facts about the problem and the rules interpreter are called production systems. Production systems are suitable for use in design systems because this formalization corresponds to the character of specialist knowledge that is organized around specific design situations and solutions for them. Another widely applied representation of knowledge is the frame structure. This representation allows description of the properties of objects by defining object-property-value triplets. This representation is natural in the design process where it is customary for objects to be defined through their physical and formal properties. By combining this representation with the concepts of inheritance and polymorphism, a very flexible structure for defining design knowledge is obtained.

Most knowledge based design systems are imagined to function as a project consultants who simulate the team work of human experts. An example is the ICADS system (Myers and Pohl 1992). The following components determine the architecture of the system: 1) a set of knowledge bases and databases on which the system relies on the work, 2) a graphic editor that creates a symbolic representation of the solution based on the designer's drawings, 3) a pre-designed module that allows the designer to replace the drawing by the objects that are automatically synthesized on the basis of the project specification, 4) a collection of "intelligent project tools" implemented as autonomous systems (agents) based on specific expertise, and 5) blackboard structure that allows communication between agents and which resolves conflicts that arise between agents. The designer either draws his solution or use predesigned modules to construct building layout. Once the designer performs the input of a project element (e.g. room), the resulting partial solution is evaluated by "intelligent design tools" e.g. independent programs that simulate the work of specialists in the field of building physics, structural stability, material costs and accessibility of individual spaces. Each of the computer simulated experts gives their opinion on the solution. If there is a discrepancy, a separate program attempts to resolve this situation by seeking a compromise solution and then sending the solution back to computer simulated experts. If the experts disagree with the compromise, the final decision is left to the human designer. Similar system is SEED (Flemming et al. 1993) where attention is focused on the conceptual design phase. A system is conceived as a collection of individual modules where each module embodies a design phase. Each module supports the entire work at a single stage, that is, it enables the problem definition, the synthesis and evaluation of alternative solutions to the problem that the module solves, and also allows the storage of synthesized solutions in a common database for later reuse, either in the context of the same project or in the context of another project in which case these data become a prototype of the solution. The program also includes a prototype system for "two-way" determination of the parameters of the solution. This approach allows direct evaluation of performance values depending on attribute changes, as well as changing attribute according to the given performance changes. Instead of developing a single design system, other researches focused on the types of project situations that arise in architectural design and the development of appropriate tools adapted to these situations (Schmitt, 1990). The basic types of design are:

routine, innovative and creative. Routine design is related to the adjustment of design model parameters, innovative to the process of model adaptation and combination by adding new parameters, and creative for the model development process. Some researches questioned established sequence of problem solving steps. The Topdown system (Mitchell et al. 1990) has been developed as a tool that allows the implementation of the "top-down" design strategy, unlike other systems that were limited to the "bottom-up" strategy, i.e. from the components to the final solution. The particularity of the system is reflected in the possibility that, if desired, the designer can “climb” to a higher level of abstraction, and in it performs parametric transformations of the object, whereby these changes automatically reflect on the complete detailed elaboration of objects that are at lower hierarchical levels and which inherit values of parameters from that object. The Distributed Design System (DDS) implemented a "whirling" sequence of problem solving steps without predefined direction in which they have to be taken and allowed start and finish at any point of the process (Petrović and Svetel 1999). DDS is modelled as an "open society" of autonomous, cooperative design agents. Each agent represents a particular design tool that enables transformation of the evolving design from one state to another or the presentation of selected parts of the information that describes the current design state. A knowledge base that supports the process and a data model that describes the evolving design are both distributed, enabling simultaneous modelling of different design solutions, and broad representation of design variety.

DIGITAL DESIGN PROCESS IN CONTEMPORARY APPLICATIONS

Over the centuries, the architectural design process relied on sketches and mock-ups as the primary medium for work and even in the era of digital design architects still emphasize the significance of traditional drawing by pointing that pencil and hand lead design thought (Belardi 2014). This view is supported by researches in cognitive science. Creation of external structures is a key feature of the overall thinking process (Clark and Chalmers 1998). The actions that a person performs during the creation of external structures are not aimed at achieving a particular goal. Instead, people use these actions to change their environment in order to gain a better understanding of the situation they are in. These actions, called epistemic actions, improve cognition by reducing the volume of internal mental processing of information (Kirsh and Maglio 1994). Primary functions of epistemic actions are the reduction of memory, number of steps and errors during mental processing of information. The performance of epistemic actions depends on an external artefact that enables understanding by expanding the existing situation with new representations.

But when architect are faced with new digital design process they tend to mix up the traditional representations like sketches that serve as external structures that drive thinking process with project documents that represent the final result of their work. As a result they do not understand role of digital models that are core components of all digital design applications and that represent digital counterpart of external structures (Svetel and Kosić 2017). Consequently, although it has long been concluded that digital design is not the implementation of traditional processes with the help of computers (Oxman 2006), today's applications continue to simulate the traditional design process and the quality of each application is evaluated to the extent that it supports a traditional document-based approach to design, rather than how much innovation it brings to the digital design process.

Preoccupation with final design documents results in the fact that most modern design applications are focused on one solution instead of exploration of alternative solutions as was the case in the past research. Freehand drawing programs are focused more on the reproduction of the sketch appearance through the digital tracing of the hand movement and imitation of the look of the pen on paper than on the sketching process. The whole process is oriented towards creation of a single digital drawing and not to the exploration of alternatives through sketching. 3D modelling applications have commands that make it possible to efficiently create and modify single 3D model. Constructive solid geometry technique that lies behind majority 3D applications treat geometry models as decision trees, but nodes in this formalism represent geometric bodies and branches performed geometric transformations. The process is focused on the process of generating a single geometric model, not on the logic of the object being modelled. BIM applications are primarily focused on the development of final project documentation, which is why most of the commands are aimed at effectively creating a single building model from which such documents can be obtained. This purpose is enhanced through the interface of these applications, which is based on the paradigm of traditional paper documentation through the use of interface elements such as layouts, sections, etc.

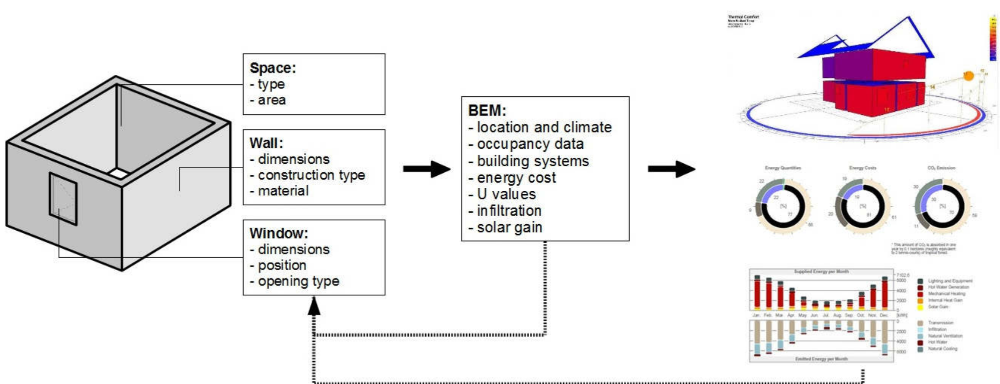

Figure 2: Example of the one-way flow of information from BIM to BEM and finally to simulation results. There is no feedback to create a building model enriched with information.

If we compare the past research with today's systems for architectural design with the help of computers we come to the conclusion that it is currently possible to use only the simplest trial-and-error method. All commercial programs for the analysis of architectural solutions require the entire building model for their work. In addition, it is most often necessary to convert the building model into a specific model that the simulation program uses and only then perform the performance analysis. If, as an example, we take the process of energy consumption simulation, whether using a BIM application or a 3D modeller, it is necessary to convert the complex geometric model of a building into a simple 3D geometric model that applications use to simulate energy consumption (Svetel et al. 2017). These simple models are then combined with other information like materials' properties, weather data, location data, occupation data, etc. to form Building Energy Model (BEM) that simulation applications use to produce their results. Most often the designer gets results in the form of diagrams and charts, and the mapping of the results is performed against simplified BEM geometry. The flow of information is one-way, from BIM or 3D model, through BEM model to the final simulation results (Figure 2).

Contemporary BIM applications such as ARCHICAD and Revit have achieved a high level of sophistication, but their orientation toward a traditional document based design process prevents the exploratory approach to design. As a solution to this problem, applications that support the so-called generative design have been developed. Grasshopper and Dynamo are visual programming languages, meaning that designer does not need to write code but is using visual objects and relations to create algorithms. Since both Grasshopper and Dynamo support development of objects in general programming languages like C#, VB.net and Python it is possible to develop any kind of algorithm. These applications basically deal with parameters that describe geometry and are widely used for form hunting in architecture. Extension toward performance based design (Oxman 2007) can be achieved by connecting geometry parameters with information that describes building functionality. Even then, when using existing building simulation applications that require complete solutions as their input, the design process does not surpass simplest trial-and-error method.

CONCLUSIONS

Today we are witnessing a great advancement in computer architectural design applications that enable the development of very complex and detailed building models, the development of complete project documentation and the execution of complex analysis of building performances. But when we compare these achievements with research done during 1980s and 1990s we can conclude that today's applications allow only basic problem solving techniques such as trial-and-error and do not allow exploration and analysis of variants that represent the true nature of the architectural design process. Because of the inability of earlier computers to process a large amount of information, these systems were experimental, but the research conducted at that time briefly illustrated in this paper, shows that it is possible to develop a digital design process that is in line with what architects really do. In order for digital tools to find the right place in the design, it is necessary to find the true role for digital models in the design and to explore the processes that this new medium defines. As long as architects rely on computer imitation of the traditional document based design process, or are uncritically dependent on the performance of existing computer systems, they will not be able to conceive a new digital-based design process. It is time to combine today's power of computers with principles that have been previously established and to set new guidelines for the development of computer tools for architectural design. There is no need to treat the inability of managing the alternatives and gradual development of the solution as features of digital design, but as shortcomings of applications. It is necessary to redefine the typology of digital models (Oxman 2006) to include an analysis of the limitations of existing applications and to expand the typology with the digital models developed in forgotten researches and accordingly to redefine meaning of the digital design process.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia under grant TR-36038. It is a part of the project 'Development of the method for the production of MEP design and construction documents compatible with BIM process and related standards.' The project director is Dr. Igor Svetel.

REFERENCES

- Belardi, P. 2014. *Why Architects Still Draw.* Cambridge: The MIT Press.
- Clark, A., and D.J., Chalmers. 1998. "The extended mind." *Analysis* 58, no.1 (January): 7–19.
- Coyne, R.D., Rosenman M.A., Radford A.D., Balachandran M., and J.S., Gero. 1990. *Knowledge-Based Design Systems*, Reading: Addison-Wesley.
- Flemming, U., Coyne, R.F., and R., Woodbury. 1993. "SEED: A Software Environment to Support the Early Phases in Building Design." In *Proceedings ARECDAO'93 Conference*, 111-122. Barcelona: Institut de Tecnologia de la Construcció de Catalunya ITEC.
- Jones, John Christopher. 1970. *Design Methods: seeds of human futures*. London: John Wiley & Sons Ltd.
- Kirsh, D., and P., Maglio. 1994. "On Distinguishing Epistemic from Pragmatic Action." *Cognitive Science* 18, no.4: 513-549.
- Lawson, Bryan. 1990. *How Designers Think: The Design Process Demystified.* (2nd. ed.) Oxford: Butterworth Architecture.
- Mitchell, W.J., Liggett, R.S., and M. Tan. 1990. "Top-Down Knowledge-Based Design." in *The Electronic Design Studio,* edited by McCullogh, M., Mitchell, NJ.J., and P., Purcell. 137-148. Cambridge: MIT Press.
- Oxman, R. 2006. "Theory and design in the first digital age." *Design Studies* 27, no.3: 229-265.
- Oxman, R., Roey, H., and A.B. Shoham. 2007. "Performative Design in Architecture." In *Predicting the Future: 25th eCAADe Conference Proceedings*, 227-234. Frankfurt am Main: Faculty of Architecture and Civil engineering, FH Wiesbaden.
- Myers, L., and J., Pohl. 1992. "ICADS expert design advisor: an aid to reflective thinking." *Knowledge-Based Systems* 5, no.1: 41-54.
- Petrović, Ivan, and Igor Svetel. 1999. "From Number Cruncher to Digital Being: the Changing Role of Computer in CAAD." In *Architectural Computing from Turing to 2000: 17th eCAADe Conference Proceedings,* 33-39. Liverpool: University of Liverpool.
- Petrović, Ivan. 1977. *On Problems and Methods of Design*. Belgrade: Architectural Faculty. (in Serbian)
- Schön, Donald A. 1988. "Designing: Rules, types and worlds." *Design Studies* 9, no. 3: 181-190.
- Simon, Herbert A. 1969. *The Sciences of the Artificial*. Cambridge: MIT Press.
- Svetel, Igor, Jarić, Marko, and Nikola Budimir. 2017. "Expectations, Reality and Perspectives in Using BIM for the Green Building Design." *Structural Integrity and Life* 17, no.3: 229–234.
- Svetel, Igor, and Tatjana Kosić. 2017. "Digital and Architecture: Still Not a Perfect Match." In *ON ARCHITECTURE Conference Proceedings*, 175-182. Belgrade: Strand.