


Computational Design Thinking through Cellular Automata: Reflections from Design Studios

Pinar Calisir Adem 

Yeditepe University, Department of Architecture, Istanbul, Turkey, (Corresponding author)

Gulen Cagdas 

Istanbul Technical University, Department of Architecture, Istanbul, Turkey,

Received: October 26th 2020, Revised: November 7th 2020, Accepted: November 11th 2020

Refer: Calisir Adem, P., Cagdas, G., (2020), Computational Design Thinking through Cellular Automata: Reflections from Design Studios, Journal of Design Studio, V.2, N.2, pp 71-83,

P. Calisir Adem ORCID: 0000-0001-6139-6289, G. Cagdas ORCID: 0000-0001-8853-4207,
DOI: 10.46474/jds.816833 <https://doi.org/10.46474/jds.816833>

Abstract: The current technologies have created a shift from Computer Aided Design to Computational Design in architecture. Computational design allows inquiries into what can be implicit knowledge in traditional design thinking, enables the definition of the mechanisms of design process and formulations of design knowledge and representation, and defines generative and evaluative knowledge. The purpose of this study is to discuss how Cellular Automata can be utilized in design studios to develop computational design thinking, through the examination of Frazer's and Herr's studio works. After finding matching concepts and comparisons of Cellular Automata methods used in two design studios, the concept of 'computation' in Cellular Automaton studies and contributions of using this generative method in design studio will be discussed. In other words, this study will examine the content of Computational Design Thinking through Cellular Automata applications and their contributions to design education. As a result, since Cellular Automata methods are exploratory processes. They enhance seeing, reaching the whole from the parts, noticing the relationships and patterns between the parts and re-inventing them during and after the generative process. For these reasons, Cellular Automata have an important role in the development of computational design thinking in design studios with different concepts and setups.

Keywords: Cellular Automata, Computational Design Thinking, Architectural Design Studio.

1. Introduction

Technology is developing rapidly and "computational design" is becoming an important part of the architectural practice and education. Also, new modes of thinking and production methods based on computation are forming new agendas for the design studios. The computational design methods, unlike the conventional ones, can enable us to design with novel techniques and to look at the design process from different perspectives. As often stated, for the development of computational design thinking in architectural education, the use of computational design methods has been

more important than the use of digital tools. In this context, the purpose of this study is to discuss how Cellular Automata, which is a generative design method, can contribute to the development of computational design thinking in design studios, based on the experiences of Frazer (1995) and Herr (2008), who have continued design studios for many years within this context.

The current technologies have created a shift from Computer Aided Design (CAD) to Computational Design in architecture. The relationship between architectural design and

computation was a topic of research among designers (Antoni Gaudi, Frei Otto, Buckminster Fuller, Cedric Price), even before computers became so common. At the same time, computational design methods had been applying to design problems in terms of ideas and logic without using a computer. However in recent years, with the development of Information and Communication Technologies, computation has had significant effects on the perception and realization of architectural form, space and structure. These effects have led to improvements in the intended use of the digital tools. In this context, there are terminological and process-based distinctions between computational design methods and CAD. As Oxman stated, “while principles, theories and methods of CAD have been basically based on imitating paper-based design, the novel concepts of digital design models are re-introducing a different medium of conceptualization, replacing paper-based media” (Oxman, 2008, p. 106). Thus, computing and computerization (CAD) concepts are different from each other (Ahlquist & Menges, 2011; Terzidis, 2003). Computation increases the number and the content of information by revealing dependencies among data sets. On the other hand, computerization or computer automation compiles the sets of data and tries to process as much as data as possible. Thus, the differentiation between CAD and computational design methods arise from the algorithmic structure of the design action. While CAD aims to create an object, the computational design, which can also benefit from computer technologies, starts the process with properties of small parts and relationships between units of the design context. It is based on formation rules and reveals the information that generates the form as a dynamic system (Ahlquist & Menges, 2011).

The computational design thinking is formed depending on various developments in areas, such as system theory, cybernetics, morphogenesis, and biology (Ahlquist & Menges, 2011). It is based on a perception that enables designers to evaluate the purpose of design not through the final product, but

through the generation process. In this manner, computational design methods are systems that produce forms using a set of stages (algorithms) that describes the formation process of form, space and structure with or without computer support. These stages contain processes ranging from the analysis and interpretation of design data, such as user, topography, climate, program; arranging the design stages and determining the formation rules; evaluating the resulting products in terms of various performance criteria, and materializing the optimum solution.

The computational design allows inquiries into what can be implicit knowledge in traditional design thinking by enabling the definition of the mechanisms of design processes. By doing so, it formulates and defines generative and evaluative design knowledge (Singh, 2012). On the other hand, the development of computational design leads emergence of a new domain of knowledge and a conceptual vocabulary. Thus, it can be stated that the computational design thinking causes a paradigm shift in design mediums, knowledge and theory, and models and methodologies. “The characteristics of topological form, the transformational evolution of spatial structure, non-hierarchical organization, complex, hyper-connective spatial conditions have become more prominent” with this paradigm shift in architecture. “Continuous, hierarchical, topological, materiality, structure and formation” are also very important concepts replacing with “discrete, intricate, typological, space, form and representation” (Oxman, 2008, p. 103). First, new modes of design thinking and designer-authored computational processes can enrich the individuality of design processes and support creativity and innovations in program, spatial qualities and materiality by breaking standard building typologies. They can help to define design based on a process rather than a final product and understand data and forces that shape design formation. Also, computational design processes can provide a different orientation to design discovery and creativity. Instead of a holistic, conventional problem definition that includes contextual,

functional and typological programs, definitions of computational models, such as "animation", "parametric design" or "generative design", can serve as a starting point for new modes of design exploration. Thus, it is very important to develop computational design thinking in design studios. Emergence, self-organization, complexity, parametric dependencies, seeing the design process as a system, feedback, evolution of forms and designer-authored formation rules are some basic concepts identifying with computational design thinking. In studios, conducting discussions and approaching the solution of design problems through these concepts can improve the computational design thinking.

In this context, the purpose of this paper is to discuss how Cellular Automata can be utilized in design studios to develop computational design thinking, through Frazer's and Herr's studio works. Cellular Automata is a generative method which involves a pre-formalized formation process in which the final products will emerge. Since the user, program and context-related data can be included in the formation process and the designer gains a new position as a developer of the formation tool, the most comprehensive way that can enhance computational design thinking can be the generative design rather than the performance-base or parametric design. The Cellular Automata is chosen among the generative design methods, because, it is a method that expresses the complex creativity arising from simplicity in the formation process, and highly complex structures can arise with a very few elements and rules.

In this study, the studio works of Herr (2008) and Frazer (1995) will be examined and discussed within the scope of design purposes and concepts that they used. Herr and Frazer were chosen, because they tried to explore generative systems, particularly Cellular Automata, in a systematic series through workshops and studios. For each studio process, a conceptual map will be prepared to deeply understand specific concepts related to design studios, and two studios will be compared in

terms of their Cellular Automata processes. After finding matching concepts and comparisons of Cellular Automata methods used in two design studios, the concept of 'computation' in Cellular Automata studies and the contributions of using this generative method in design studios will be discussed. In other words, the content of Computational Design Thinking through Cellular Automata applications and its contribution to design education will be elaborated. In this line, this paper will try to answer these research questions:

1. Is there a match between the concepts used in Cellular Automata studios and the concepts associated with computational design thinking in the literature?
2. What is the nature of computation in the process followed in Cellular Automata studios and how is this applied in architectural formation in design studios?
3. What are the contributions of Cellular Automata and what aspects of these models can be improved for enhancing Computational Design Thinking in design processes?

2-Generative Design Techniques

Computational design tries to solve design problems by defining specific steps, thanks to the algorithmic methods they use (Terzidis, 2003). Oxman (2008) divides computational design methods in 3 categories based on the process following by designers: the formation models, the performance models and the generative models. The formation models contain animation and parametric design approaches. The performance models suggest simulations for analysis, synthesis and generation of forms. Similarly, the generative models comprise formal methodologies consisting of rules and procedures to generate designs. Shape grammars, mathematical models, topological features, evolutionary algorithms (Frazer, 1995), mapping and morphism are algorithmic processes that reveal unpredictable, distinctive, formal behaviors and properties of space that can be used in architectural design. These algorithmic procedures lead to various generative design processes.

Design is a process in which the acquisition of knowledge and the production of forms evolve together. In a generative design process, the emergence of forms enables designers to look at the design problem from a new perspective and enriches the design process and design outcomes. Also, the generative design must be able to produce complexity, be based on the relationship with the environment, have the ability to fit the changing environment, and produce unpredictable, novel relationships or outcomes (McCormack et al., 2004). Generative design methods enable architects to explore a large design space. In addition, generative design systems are one of the main computational methods that can be used to enable automation in design (Gu, et al., 2010). There are 5 basic generative design algorithms frequently used in the literature: Shape Grammars, L-Systems, Genetic Algorithms, Cellular Automata and Agent-Based Systems. There are differences and similarities between these algorithms and different generative models are suitable for solving different design problems (Gu, et al., 2010).

Shape Grammar is used to produce a design language or a set. Shape grammar can be both a generative and a descriptive method. It can be used as a design tool to produce different design languages or as a formal analysis tool revealing the formation rules of a specific design. Shape grammars produce two and three dimensional compositions in terms of spatial arrangements.

L-System is a mathematical algorithm based on the recursive rewriting of a string with generation rules. The concept of a string is a symbolic expression of form in architecture. L-Systems create repetitive patterns. It can be used in an urban or architectural design while aim is to produce fractal-based forms.

Genetic Algorithms are inspired by evolutionary processes found in nature. They search for different solutions in the search space, using evolutionary methods to find optimized solutions based on a fitness function. Usually, it can be used for optimization in design.

Agent-Based Systems are often utilized for implementations in social or collective behaviours. Acts of agents are autonomous and independent, but they can also interact and communicate with each other to compete or collaborate, and collectively achieve specific goals. Local interactions of unsophisticated agents cause collective behaviours leading global patterns to emerge.

Cellular Automaton (pl. cellular automata, abbrev. CA) is context-sensitive based on its nature. Its generative process is controlled by the states of the neighbouring cells. Design constraints are implemented from bottom-up and the local behaviours of each cell affects the whole. As a result, the outcomes of Cellular Automata are often complex and difficult to predict. Although Cellular Automata depend on a finite set of elements and rules, an appropriate rule and initial configuration can create a setup that is perceived as unlimited. Thus, they enable emergent properties of the formation process and help designers to see and interpret new relations between cells. Since the cells can be represented by graphic symbols, the system allows abstraction in the design process. Because of the context-sensitive nature of Cellular Automaton, it allows many different factors to interact within the defined limits. Interaction makes the model suitable for systems focusing on the behavior of the elements. It provides editing after the generation process in terms of transformation of forms or representing cells with various functional forms. Also, emergent form can be the result of the desired functionality. As a result, for this study, Cellular Automaton is chosen for exploring computational design thinking in design studios due to its simple setup, abstract nature and rich set of results.

3-The Nature of Cellular Automata

The Cellular Automaton enables the cell population on a grid to be formed depending on certain rules based on their relationship with neighbouring cells in a consecutive time period. The method can be run by determining neighbouring typologies and different rules in one, two or three dimensional spaces. By

establishing appropriate rules and an initial setting, the generative power of Cellular Automata can make possible to see all the potentials that may occur in the set of design results. Also, defining the appropriate rules can help manipulate the parameters affecting the design problem in terms of context or functionality. The Cellular Automaton is a generative design process, because the relations in the system are defined by the smallest units (cells) and affect the global form, reveal emerging behavior and unpredictable results. Associated cells come together depending on the formation rules over generations, to produce global shape. The ability of Cellular Automata to produce symbolic structures, the arrangement of their grids, cell formation rules and cell states in accordance with different contexts enable the methods to transform in accordance with the nature of the design problem. As a result, Cellular Automaton is shaped by using local knowledge, depending on the simple relationships of the cells and has the potential to produce complex forms.

The Cellular Automata were discovered by John von Neumann (1951) in the 1940's and developed by Stanislaw Ulam. Cellular Automaton is a mathematical method in which simulation models of complex structures can be formed as a result of simple structures following simple rules (Krawczyk, 2002). Because of its simplicity and complex productivity, the traditional Cellular Automata are used for the purpose of exploring and modelling many complex phenomena in nature. The method can be used to simulate complex formations, such as urban development and sprawl, fire and disease spread and traffic flow. The structure of simple computer programs can be reduced to visual and mathematical systems based on rules, and the processes in nature can be simulated by these mathematical models (Wolfram, 2002). Mathematical model based on computer technologies can make abstractions and simplify the relationship between parameters while looking at different kinds of complex systems in nature. In this context, Cellular Automaton has been applied in many disciplines due to its analysis of

complexity based on its simple constructs. Wolfram (2002) states that no matter how simple the rules are, systems created by Cellular Automata cause a certain level of complexity and randomness. Fully repetitive patterns, repetitive-intertwined patterns, irregular-random structures, and finally, a complex structure that appears to be irregular in general with local patterns may occur in the global form of the Cellular Automata.

In sum, structures in nature that are perceived as complex and incomprehensible can actually be formed with the Cellular Automata with by following rules, and repetition of these rules over generations can create emergent properties. When we discuss the Cellular Automata within the scope of a design problem, it is necessary to proceed the design process by examining each factor/parameter/determinant and their relations revealing the design problem. If designer understands the nature of the design problem with the nature of Cellular Automata, associations can be build and context-sensitive nature of the model can be triggered. Five elements are needed to construct a simple Cellular Automata method. These elements are a grid/lattice, a neighbourhood, cell states, time and transition rules.

Grid/Lattice: In the Cellular Automata, each cell should be located on a specific space in the grid. The grid can be one, two or three-dimensional (Figure 1). Also, the grid or the lattice can consist of regular or irregular geometries (Figure 2).

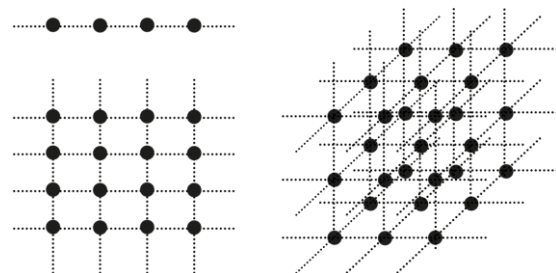


Figure 1: One, two and three-dimensional grids.

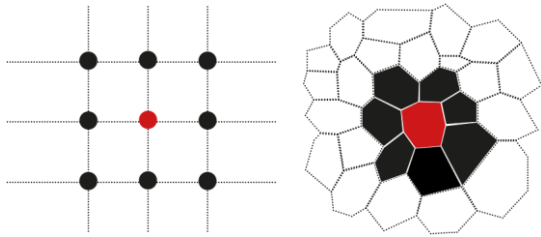


Figure 2: Regular and irregular two-dimensional grids.

Grids help designers locate neighbours by giving the location of each cell within the system and make cells aware of their neighbours.

Neighbourhood: The group of cells interacting with each other in the grid, constitutes the neighbourhood. The simplest one-dimensional automaton has a neighbourhood unit consisting of three cells, while the simplest two-dimensional automaton has a neighbourhood unit consisting of five cells (Figure 3).

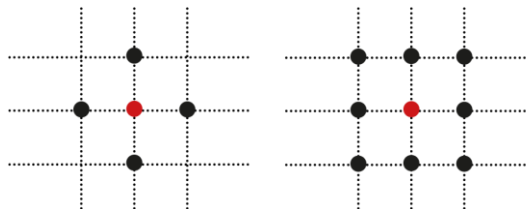


Figure 3: Left: Von Neuman neighbourhood. Right: Moore neighbourhood.

Cell States: Cell states are values, which the cells convey over consecutive time stages. These values are based on local interactions and information exchange with neighbours and may vary depending on transition rules. The cell state can be formed by a binary definition (on / off || 0/1 || live / dead), can be defined by symbolic colors, or expressed with numbers. Therefore, the Cellular Automata can also be seen as data systems that carry information about the configuration of cells (Coates et al, 1996).

Time: In the Cellular Automata, time is sequential and discrete. As time passes, the state

of the cell is updated in each time unit and the system updates itself accordingly. During the generation process, each new neighbourhood unit is called "generation" in parallel with the time unit.

Transition Rules: In the Cellular Automata, the transition rules are determined by the designer. The rules, reflecting the system accurately, should depend on the neighbourhood relations and determine the new state of the cells in the next generation. Rules are applied to all cells in the system synchronously and provide general character of the system. One of the best known examples of Cellular Automata is "Game of Life". There are four basic rules in the Life simulation (Figure 4). Two of these rules determine the state of death, one of them determines the survival status and the other determines the birth status.

Death: If the living cell has less than two living neighbours at $\{t\}$ moment, that cell dies from loneliness at $\{t + 1\}$.

Death: If the living cell has more than three living neighbours at $\{t\}$ moment, that cell dies from overpopulation at $\{t + 1\}$.

Survival: If a living cell has two or three living neighbours at $\{t\}$ moment, that cell will survive at $\{t + 1\}$.

Birth: If the dead cell has exactly three living neighbours at the moment of $\{t\}$, that cell becomes alive at $\{t + 1\}$, that is, it is born.

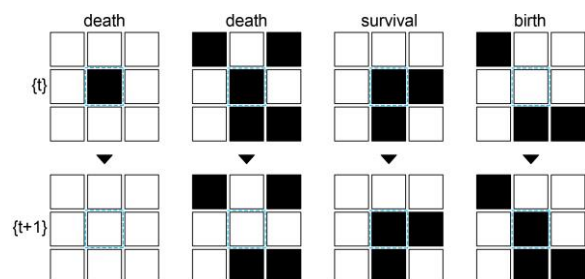


Figure 4: Game of Life rules.

In different Cellular Automata studies on urban and architectural scale, designers mostly divide the system into sub-parts, reveal and reconstruct the relationship between the parts and create repetitive formation rules in the design process.

In these studies, Cellular Automata are utilized as generative and analysis tools in different architectural contexts. In order to make the Cellular Automaton easier to produce the intended form in architectural scale, it is necessary to customize the transition rules, assign multiple and functional cell states, determine the rules for geometric transformations, and make quantitative and qualitative analyzes for the architectural problem, before the model is constructed. Due to the suitability of the method to the architectural design process and the ease of its editing, examples of applications are also encountered in design studios. In the next section, John Frazer's (1995) and Christiane M. Herr's (2008) studio experiences will be examined and discussed in terms of how Cellular Automata might be enhanced the computational design thinking in studios.

4-Cellular Automata in Design Studios

John Frazer's studio, Diploma Unit 11 in the Architectural Association (AA), continued from 1990 to 1995 to build experimental interactive models, generative and evolutionary systems, and explore architecture as an artificial way of life (Frazer, 2005). The produced architecture was accepted as a part of the natural system, exhibited as 'metabolism' acting like the mechanisms to which it was formed: in exchange with the environment, responsive to feedback and evolutionary in its own right



Image 1: "Universal Constructor", Unit 11, Gordon Pask and Julia Frazer, 1990 (Frazer, 1995).

(Ahlquist & Menges, 2011). One of the important Cellular Automata-based

experiments of this group was the project called "Universal Constructor". The system was implemented as a Cellular Automata hardware system controlled by a main computer that also functions as a human-computer interface (Herr and Kvan, 2007).

Universal Constructor (Image 1) was made up of 3-dimensional identical cubes in an array. A cube shape was chosen because of its universality and potential for abstraction. Each cube had 8 LEDs to display 256 different cell states. Thus, 8-bit code could be used to match states of the cells with any form, structure or environmental determinants. Cubes covered 12*12*12 units in a volume called as "the logic space". Each cube had a defining state and also the ability to represent any other states. Messages could be transferred between cells by streaming data in a serial form by going one stack cell down and up. The array of the cells used as an input or output device. For an input device, the exact configuration, location and identifying code of every cell could be deduced by the controlling processor questioning each location and provoking for a possible neighbourhood above. For an output device, each cell could represent 256 messages with the 8 LEDs: one flashing light expressed "take me away" and two flashing lights expressed "add a cube on top". This 3-dimensional hardware Cellular Automaton was a tool for the explanation and demonstration of a radically new design process and it was understood as an expression of logic in space in architectural terms (Frazer, 1995). Various different experiments were conducted with this 3-dimensional hardware. One of these works was done by Stefan Seemüller in 1991 and called as "Evolving Sequence". In this experiment (Image 2), an evolutionary model also added to the system in order to test output results against the rule. In this model, the process started with definitions of the initial configuration of cubes, states and rules for generation and testing. Rules applied a series of loops in a chaotic string where each step in the evolutionary development was continuously read and tested against the rules, and at the end, emerging patterns appeared. Later, the system upgraded

with 16-bit state patterns which produce a high level of complexity and unexpected, also logical forms.

Cellular Automata and the resulting form could be seen. One automaton controlled the evolution of structure and the other controlled the

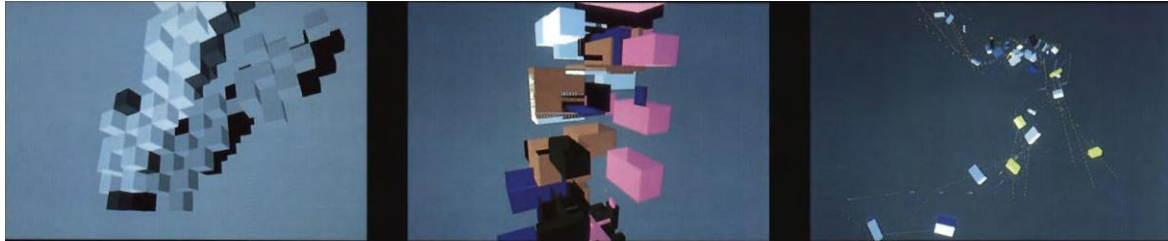


Image 2: Evolving Sequence from the Universal Constructor by Stefan Seemüller, 1991 (From John Frazer, An Evolutionary Architecture, Architectural Association publications, Themes VII, copyright John Frazer and the Architectural Association 1995, p.46).

In the below experiment (Image 3), self-organized cubes behaved like a 3-dimensional Cellular Automaton, at the same time, a positive and negative feedback loops existed between adjacent cells, causing interaction or in other words, flow of information. The interaction between cells relied on their comparison of the features of their growth, and their effort to duplicate successful behavior.

In the third experiment, which was called “Hierarchical Cellular Automata” (Image 4), two different Cellular Automata were merged to create 3-dimensional output. In this study, the coexistence of two different constructions of

environment. The interaction between formations created flowing, intersecting surfaces as outputs. In addition, this study is an example of how abstract forms could be transformed into concrete forms during the form-finding processes.

Frazer (1995), produces a generative design model based on abstract expressions and computer-human interaction, without being tied to a real architectural space or an environment. These abstract expressions and rules can be utilized for exploration of architectural spaces, patterns, and contain concepts of computational design thinking, such as “rule, state, loop, hierarchy, feedback, interaction, development

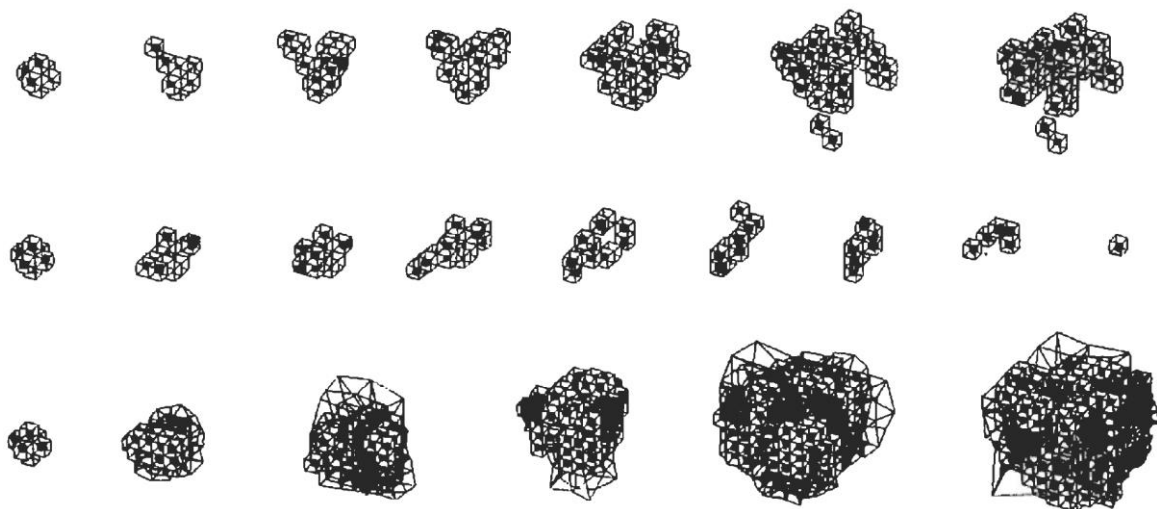


Image 3: Three-Dimensional Self-Organizing Constructor by Ichiro Nagasaka, 1991 (From John Frazer, An Evolutionary Architecture, Architectural Association publications, Themes VII, copyright John Frazer and the Architectural Association 1995. p. 50).

of form, complexity, self-organization” (Figure 5).

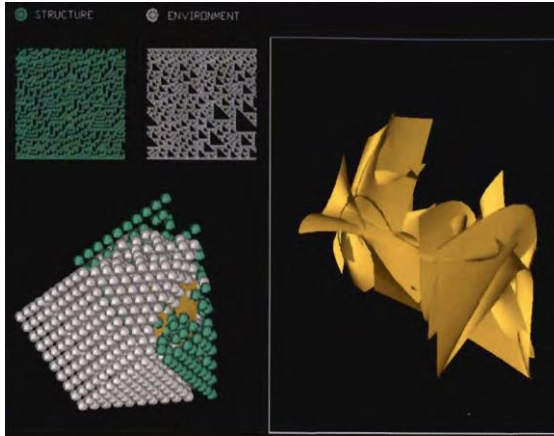


Image 4: Hierarchical Cellular Automata by Mani Rastogi, 1994 (From John Frazer, An Evolutionary Architecture, Architectural Association publications, Themes VII, copyright John Frazer and the Architectural Association 1995. p. 89).

Christiane M. Herr (2008) explores Cellular Automata as a design support in architectural design, based on series of studio experiments and workshops. Herr, focuses on how Cellular Automata can be adopted in the architectural design context, especially in the early conceptual design process. In this regard, Herr’s studios differentiate from Frazer’s ones. Frazer searches for a more general theory, while Herr

concentrates initial steps of the architectural process, although her approach can be applied also in further steps in architectural design.

Herr, developed object-based extended Cellular Automata by improving cell and neighbourhood shapes and definitions in terms of architectural design process. Also, the context of the model had an actual program or an environment, and merged top-down and bottom-up dynamics. According to Schön (1983), in conceptual design processes, designers make use of design support to assist in the reflective conversation with the situation they are engaged in. For this reason, conventional conceptual design support usually provides representations in the form of sketches and models, which allow for different interpretations and visual discoveries. Used as exclusive, self-sufficient tools, Cellular Automata are unlikely to integrate well into the architectural design process, nor generate desirable outcomes. The extended design process model maintains the basic activities of framing, moving and reflecting or evaluation as characteristics of Schön’s (1983) design process, and offers Cellular Automata-based generative processes as optional design moves. Additionally, such Cellular Automata-supported design moves provide nested evaluation loops as each element (Cellular Automata cell) is capable of automated

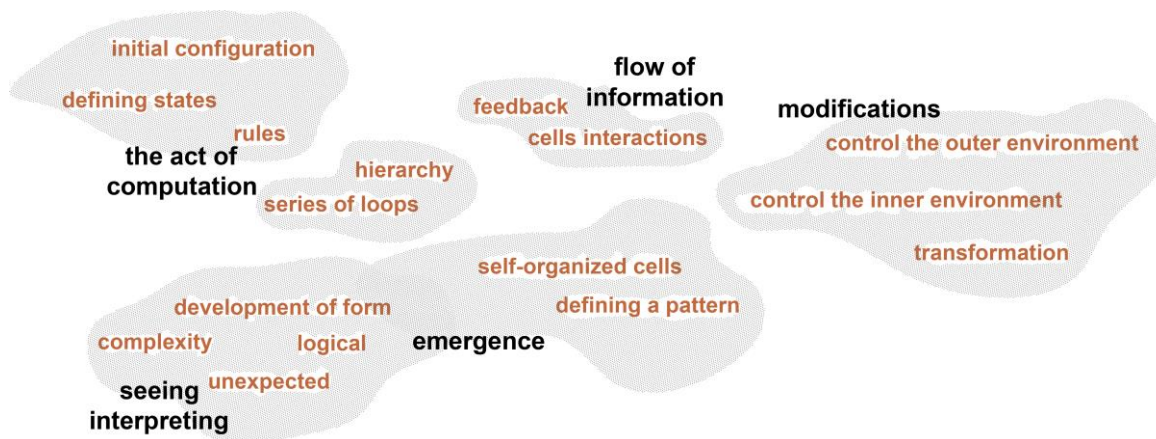


Figure 5: Conceptual map of Frazer’s design studios: orange colors express specific concepts, black ones express generic concepts of computational design thinking.

evaluation of numerical constraints. While Cellular Automata are still employed as generative mechanisms with potentially surprising results, this process allows designers more immediate control in determining the direction of the design process. In this context, Herr (2008) created a Cellular Automata software. With executive studio experiments and enhanced software properties, Herr made Cellular Automata more appropriate for the designers' demands. In this process, while the first two experiments were the first trials of the software, the last experiments tried to respond to different demands in the architectural design process with more interesting transformations.

The earlier studio tests of Herr showed that students wanted to experiment with the Cellular Automata software without chasing bias/goal-directed results. In addition, the functions and rule compositions found too detailed and disruptive. Thus, for the Tofu Cubes and Soap Bubble studio (Image 5), conducted with Thomas Fisher in 2006, the level of generative automation of functions tried to increase with

less required parameter settings. On the other hand, the increased automation and the generation of more complex and unexpected forms could provide richer design outcomes. In sum, the aim of this studio was to build a Cellular Automata software with few input, but more complex and unexpected outcomes based on the generative process. Also, in the studio two different processes were used. The first is the Tofu Automata Generator which worked as bottom up, and the second is the Soap Bubble Truss Co-Rationaliser which functioned as a top-down. In these processes, Herr (2008), added some functions performing to adapt results, such as multiplication, attraction, repellent, leveling, scaling or deleting. While the processes were designed in two different approaches in this study, it was revealed that the students used both generative systems in a bottom-up manner. Geometrical translations of forms and the inverted volumes based on parametric properties of shapes were some studies emerged at the end of the design process. In these works, fewer initial parameters created intricate and complex behaviours. Also,

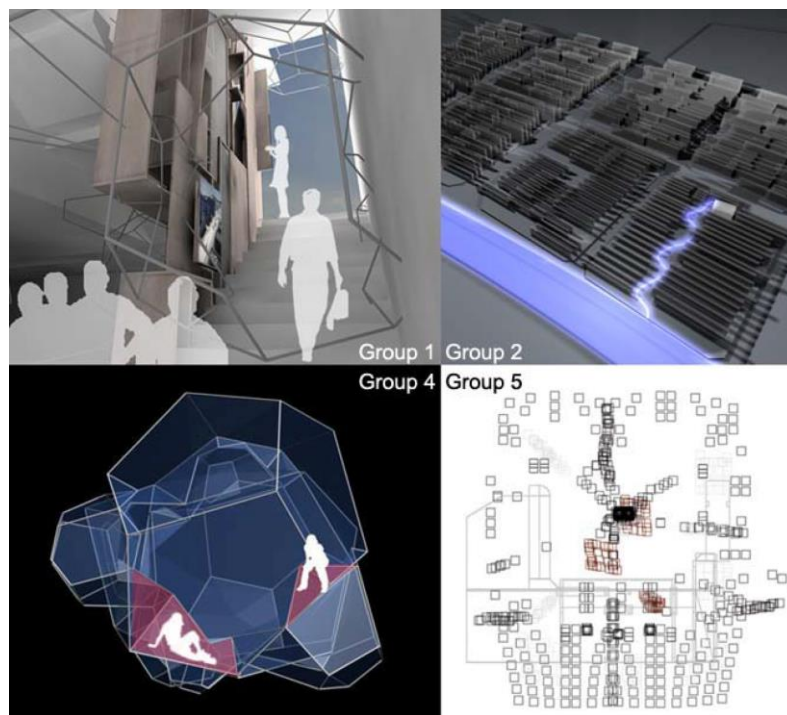


Image 5: Modified results from Tofu Cubes and Soap Bubble studio (Herr, 2008, p. 99).

the processes focused on configurations and relationships of elements rather than the 3-dimensional forms. The generative processes were used for the initial concepts and further modifications were applied.

The next experiment, called “KCRC Urban Automata”, was created for a specific urban design problem (Image 6). Therefore, multiple cell states were used to present functions of urban elements, such as living, working, recreation and landscape. The Cellular Automata generated the complex organization of urban elements based on rules defining the characteristics of urban space. The system, also permitted manual interventions throughout the generative process. The results represented not the actual urban forms, but abstractions of urban models. Therefore, the system was open to further modifications for detailed urban arrangements.

based diagrammatic representations of design programs (Image 7). Thus, it can be also called as a generative model of “automated diagrams”. The model assisted thinking and reflections in terms of functional relationships of a design task. Automated diagrams were not limited to representing forms, but also allowed for various interpretations and mappings. To build a rule set for Alrogram, the context and building program should be considered and functional relations should be revealed. By doing so, one could generate a rule set and proceed to the generative phase. Alrogram could achieve to produce different types of spatial relations such as hybrid or intersected spaces. In addition, these spatial qualities also allowed ambiguity which could enhance the creativity of designers during the early design phases.



Image 6: Abstract outcomes of KCRC Urban Automata: different colors representing the density of various functions (Herr, 2008, p. 106).

“Alrogram”, the last model from Herr (2008), was an implementation determined aspects of representation and interpretation as main research points. It provided a support during the conceptual design phase by producing rule-

Herr (2008), develops a generative model to be used in the early stages of design with abstract expressions, try to relate a real architectural space and environment with each other. In this process, the concepts of computational thinking, such as bottom-up process, context-

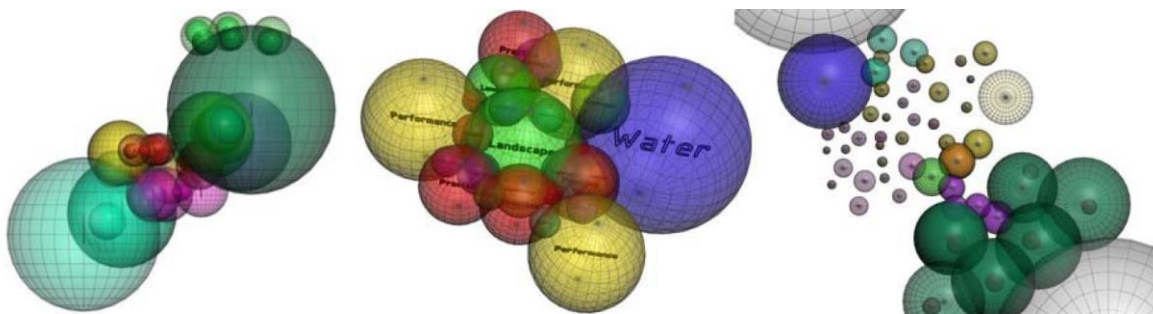


Image 7: Different diagrams generated by Alrogram model (Herr, 2008, p. 115).

sensitive, if/then logic, ambiguity were mainly discussed (Figure 6).

transformations, generated space, evolution of form, patterns, self-organization are frequently mentioned concepts and result products are

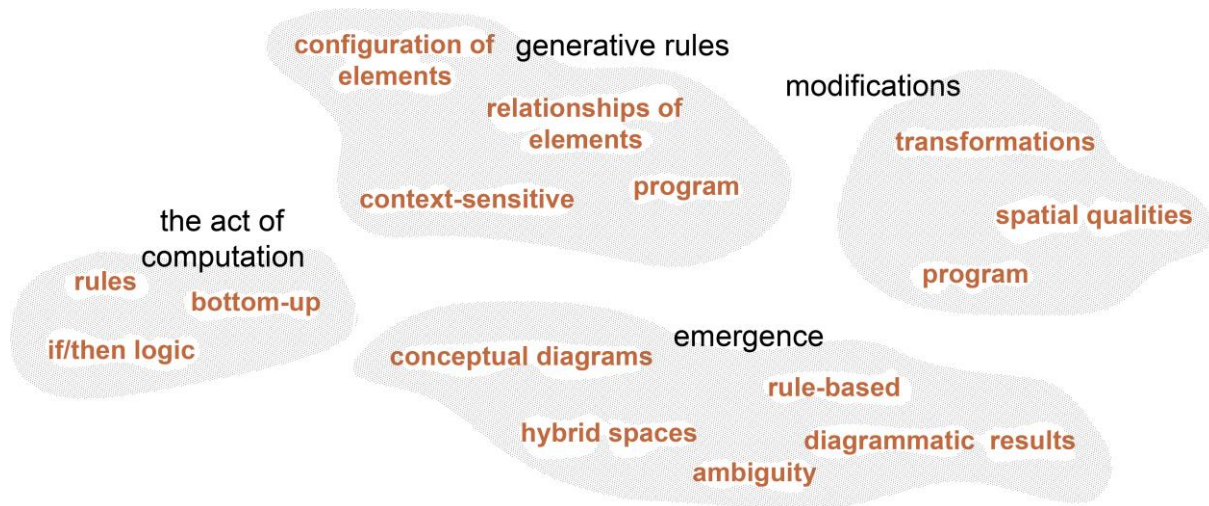


Figure 6: Conceptual map of Herr's design studios: orange colors express specific concepts, black ones express generic concepts of computational design thinking.

5-Discussion and Conclusion

Experiments in Frazer's and Herr's studios enhance Cellular Automata in two different ways. Frazer explores unusual formations and the potentials of computer-human interaction through the computational aspects of Cellular Automata. On the other hand, Herr has a more practical purpose and is trying to find a model of the Cellular Automata that can be utilized by designers in the early stages of design, by solving a program and context-related design problems with a conceptual diagrams. Although their aims are different, the two design studios actually follow similar processes. First of all, studios consider design as a system and evaluate it as a whole in terms of parameters, relations, rules, operations and final products (Sönmez & Ataş, 2016). The generative rules produced in these studios are constructed from abstract relations related to the program, form or context. Thus, the studios contribute to understanding of the basic concepts of computational design thinking in the context of architectural design. A system, emergence, parametric relations, morphological

discussed and evaluated through them. In both design studios, after understanding the nature of computation, post-operations or modifications, such as interpretation and manipulation have been applied, and the coherence of the resulting forms to the architectural program and the context can be discussed. Potentials and restrictive features of computation are discovered, and accordingly, the role of the designer in studios is changed from only a tool-maker into a designer of the post-operations.

In this study, first of all, the concept of computational design thinking and the generative systems in architecture were introduced and the conceptual background related to the subjects have been drawn. After, processes and concepts of Herr's and Frazer's Cellular Automata studios were examined in terms of Computational Design Thinking. Since Cellular Automata produce abstract results, they are very effective in understanding computational design thinking in the architectural design process. They enable the discussion of the computation-related concepts

by producing abstract fictions. They increase the possibilities for the exploration of concepts such as emergence, ambiguity, seeing through resulting forms. As a result, since Cellular Automata are exploratory processes, they provide to see, to reach the whole from the parts, to notice the relationships and patterns between the parts and to re-invent them during and after the generative processes. For these reasons, Cellular Automaton has an important role in the development of computational design thinking in design studios with different concepts and setups.

References:

- Ahlquist, S. & Menges, A. (2011). *Computational design thinking*. London: John Wiley and Sons.
- Coates, P., & Healy, N., & Lamb, C. & Voon, W.L. (1996). The use of cellular automata to explore bottom up architectonic rules. *Eurographics Conference*. Imperial College of Science and Technology, London.
- Frazer, J. (1995). *An evolutionary architecture*. Architectural Association publications, Themes VII.
- Frazer, J. (2005). Computing without computers. *Architectural Design*, 75(2), 34-43.
- Gu, N., & Singh, V., & Merrick, K. (2010). A framework to integrate generative design techniques for enhancing design automation. In Dave, B., & Li, A. L., & Park, H. J. (eds.), *New Frontiers: Proceedings of the 15th International Conference on Computer-Aided Architectural Design Research in Asia CAADRIA 2010*, 127-136. Association for Research in CAADRIA, Hong Kong.
- Herr, C.M., & Kvan, T. (2007). Adapting cellular automata to support the architectural design process. *Automation in Construction*. Volume 16, Issue 1, 2007, 61-69.
- Herr, C. (2008). From form generators to automated diagrams: using cellular automata to support architectural design. (PhD Thesis).
- Department of Architecture: The University of Hong Kong.
- Krawczyk, R. J. (2002). Architectural interpretation of cellular automata. *International Conference On Generative Art*. Milan, Italy, December, 2002.
- Oxman, R. (2008). Digital architecture as a challenge for design pedagogy: theory, knowledge, models and medium. *Design Studies*. Design Studies 29 (2008), Elsevier Ltd. 99-120.
- Schön, D. A. (1983). *The reflective practitioner: how professionals think in action*. New York: Basic Books.
- Singh, V. (2012). Towards an integrated generative design framework. *Design Studies*. Vol.33, No.2, March 2012, Elsevier Ltd. 185-207.
- Sönmez, N. O., & Ataş, Z. (2016). An alternative design computing and education approach through the concept of computespace. *Yapı*, 419. (October 2016). 142- 146.
- Terzidis, C. (2003). *Expressive form: a conceptual approach to computational design*. Taylor & Francis.
- Von Neumann, J. (1951). The general and logical theory of automata. *Cerebral Mechanisms in Behavior: The Hixon Symposium*. New York: John Wiley&Sons.
- Wolfram, S. (2002). *A new kind of science*. Champaign: Wolfram Media