Tangible Computing for Establishing Generative Algorithms

A Case Study with Cellular Automata

Emad Al-Qattan[1], Wei Yan[2], Philip Galanter[3]
[1,2,3] Texas A&M University
[1,2,3] {emadkkqattan|wyan|Galanter}@tamu.edu

The work presented in this paper investigates the potential of tangible interaction to setup algorithmic rules for creating computational models. The research proposes a workflow that allows designers to create complex geometric patterns through their physical interaction with design objects. The method aims to address the challenges of designers implementing algorithms for computational modeling. The experiments included in this work are prototype-based, which link a digital environment with an artifact - the physical representation of a digital model that is integrated with a Physical Computing System. The digital-physical workflow is tested through enabling users to physically setup the rules of a Cellular Automata algorithm. The experiments demonstrate the possibility of utilizing tangible interaction to setup the initial cell state and the rules of a CA algorithm to generate complex geometric patterns.

Keywords: *Physical Computing, Tangible User-Interface, Cellular Automata*

INTRODUCTION

The work presented in this paper explores the potential of tangible interaction to setup algorithmic rules for computational models. Algorithms enable designers to generate complex geometric compositions that are imbedded with design intents (Woodbury 2010, Aish 2005). Algorithms for computational modeling are commonly created through declarative (graph-based) and/or imperative (text-based) programming methods (Appleby and VandeKopple 1997, Davis 2013). Both programming paradigms require designers to explicitly implement mathematical functions and geometrical algorithms to create computational models (Stavric and Marina 2011). However, it is found to be challenging for designers to conceptualize forms mathematically and algorithmically (Woodbury 2010). Research claims that, designers lacking such skills are limited in their creative design process and in their ability to communicate design intents digitally (Kępczyńska -Walczak 2014).

In contrast, the proposed workflow aims to provide designers with a method to setup algorithmic rules through their interaction with physical design objects. The experiments done for this research link an artifact with a virtual modeling environment (Figure 1). The artifact is the Tangible User-interface (TUI), which is composed of (1) a physical model and (2) a physical computing system. The former is a repre-

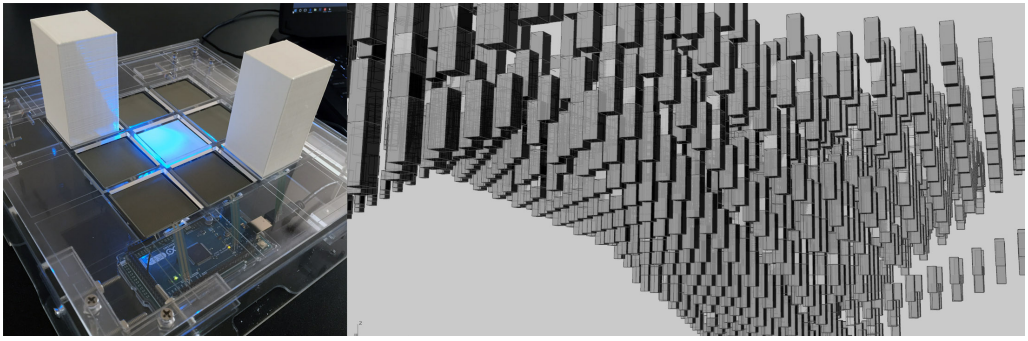


Figure 1
TUI, an artifact representing Cellular Automata cells and grid (left) linked to a parametric modeling tool to generate geometric patterns (right).

sentation of the digital model, which designers will interact with and manipulate; and the latter, is the platform used to translate designers' analog inputs into digital data using a set of sensors. The TUI will assist in capturing design intents into the virtual environment, and it is tested through enabling designers to setup the rules for a Cellular Automata (CA) algorithm to digitally generate three dimensional geometric compositions.

BACKGROUND

Research has shown that computer programming generally is challenging for users, because of their difficulty to (1) comprehend its abstract notions, (2) construct algorithms, and (3) envision the algorithms' applications in real world situations (Lahtinen et al. 2005). In architecture, research states that programming applications present an additional level of complexity for designers as they are required to learn and implement algorithms in a short period of time without having the proper training and basics of computer programming (Austin and Qattan 2016, Muller and Kidd 2014). Most importantly, designers find it difficult to translate intuitive design knowledge explicitly into digital models (Monedero 2000).

Research suggests that integrated environments (digital-physical) provide designers with an intuitive approach for interacting with digital environments (Gannon 2014, Al-Qattan et al. 2016). Physical design representations in a digital-physical workflow provide designers with familiar objects, which make the process of interaction with digital models intuitive and direct (Dourish 2001). Additionally, manipulating physical design representations takes the advantage of designers' haptic skills in the digital design process (Ishii 2008). However, examples of integrated environments have shown to be (1) limited to geometry manipulation; and (2) created to work with predefined programming workflows, which the designer has setup prior to linking the artifact to the digital model (Al-Qattan et al. 2017). Current examples of such works demonstrate an additional level of complexity that is added to the overall design process, i.e. the designer in this case must create the computer algorithm and the circuitry for the physical computing system (Vermillion 2014).

For this research, the objective is to provide an approach where the designer will utilize the physical representation to setup the computer algorithm. The manipulation of physical representations will be translated into algorithmic rules, representing design intents, which will then be used to generate the geometric composition in the virtual environment.

Related works

Research by Zuckerman et al. (2005), Klemmer et al. (2006), and Horn and Jacob (2007) show a few of the examples that utilize tangible interaction with physical artifacts that are used in classrooms to teach students the basics of computer programming and to

assist them in constructing algorithms. Moreover, McNerney provides an extended overview of the development of such works using tangible interaction, which explores the feasibility of functional programming in the physical environment to assist teachers and students in conducting experiments (2004). Such works may not necessarily link artifacts to a digital modeling environments for design purposes, yet the method maybe applicable in the context of architecture to assist designers in establishing computational models. An earlier example of utilizing TUIs to establish digital parametric models is shown in the work done by Al-Qattan et al., which enables designers to create design object relationships through manipulating physical geometry (2017). The TUIs developed for their work can deduce the relationship created between physical objects in a mathematical equation form, and set it up as a parametric constraint in the digital model. The work shows the potential of tangible interaction to establish computational models that represent a design intent.

Cellular Automata

Cellular Automata (CA) is a generative system that has been largely explored as a design method (Cruz et al. 2016). CA consists of an infinite lattice with each cell having two possible states, Alive or Dead. The eight neighbors surrounding a cell will determine its future in later generations in the system's evolution. Since von Neumann (1963) introduced CA, it has been extensively researched and developed for several applications across the fields of art and science. Much of CA's popularity is due to Conway's Game of Life, which produces emergent and complex patterns that resemble the dynamics of living organisms (Gardner 1970, Krawczyk 2002). The game is represented by a two-dimensional lattice and implements a simple set of rules to determine the life and death of a cell in its evolution (Frazer 1995).

For purposes of the work presented in this paper, which focuses on tangible interaction for generating algorithmic rules, CA provides a good example for testing because the characteristics of the generated geometry can be interpreted as spatial configurations for creating architecture (O'Sullivan and Torrens 2001, Herr 2008). Herr additionally mentions that (1) three-dimensional CA geometric configurations suggest building forms, urban layouts, etc.; (2) CA depends on "procedural rule-based logic", which can be associated to the rules that govern architectural form and function; and (3) a temporal dimension can be associated with design development in an architectural design process (2008). Frazer mentions that, CA is straightforward to utilize in a design context, where simple rules can rapidly generate complex geometric outputs (1995). Frazer additionally provides an early example of utilizing artifacts to create a self-organizing system that is based on CA rules. Currently, the Grasshopper plug-in Rabbit [1] provides designers with an approach to construct such evolutionary systems in the digital environment. However, the process of generating and elaborating on CA rules remains as a complicated process (Araghi and Stouffs 2015).

Therefore, the objective of the work is to assist in setting up algorithmic rules, using CA as an example, through artifacts for computational models. The work will present a novel method using a TUI to create a generative algorithm for parametric design. In addition, it contributes to the existing works investigating CA in the context of architecture.

RESEARCH QUESTION

The proposed work aims to address the current limitations of implementing computer algorithms in the context of design through answering the following questions: *What are the types of physical interaction that could be captured and digitally interpreted into algorithmic rules for creating parametric models?* A prototype is developed to test the proposed digital-physical workflow. A CA algorithm is chosen for this work, because of its application in architecture study, and its clear algorithmic grammar; e.g. cells are Alive or Dead based on the number of their neighbors. Designers will setup and alter the rules of the algorithm by manipulating the physical objects in the artifact.

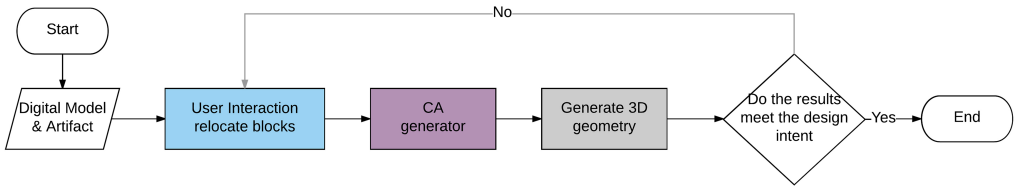


Figure 2
Prototype workflow, which includes three main parts: user interaction (blue), algorithm (purple), and modeling environment (gray).

RESEARCH METHOD

For this work prototype is developed, and it consists of three main parts: a visual programming environment, a CA generator, and a physical representation of the CA base grid and cells (Figure 2). The prototype is used to conduct two experiment, where each explores an alternative visual programming workflow to translate analog data into the digital environment. Experiment 1 tests the artifact to generate initial cell configurations ("Seeds"); and Experiment 2 tests the artifact to setup the rules for CA by determining the surviving cells through counting the number of neighbors. The prototype in this research focuses on providing a TUI for designers to relate both digital and physical models together. The benchmark for evaluating the experiments is the correctness of the system's inputs and outputs, i.e. user interaction, and generated rules and geometry.

The tools used for this research are categorized into two groups: software and hardware. Software tools include: Rhino (geometry modeler), Grasshopper (visual programming environment and a plug-in for Rhino), Firefly (data communication between the artifact and Grasshopper [2]), Rabbit (CA components and plug-in for Grasshopper). Hardware tools include: Arduino UNO (open source microcontroller with a single integrated circuit), and eight pressure sensitive resistors.

PROTOTYPING

For this work two experiments were done to explore the potential of utilizing tangible interaction and artifacts to setup algorithmic rules for a CA algorithm. The artifact for this work included a 3 by 3 square grid resembling a single cell neighborhood and a total of 3 blocks representing the live neighbor cells. The pressure sensors of the physical computing system are integrated with the physical CA grid. Each sensor is directly linked to its corresponding square in the Rhino model (Figure 3). The sensors in this prototype will be used to indicate cell configuration (location of the blocks on the grid) and to count the number of neighboring cells (number of blocks) for each born cell. The eight highlighted squares of the grid in Figure 3 (left) show the Rhino cells that are linked to the sensors in the artifact. The ninth cell (center square) in both the Rhino model and the artifact is the initial cell which will be generated.

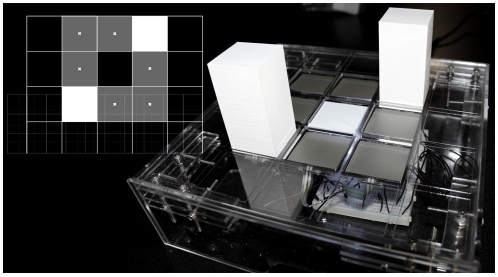
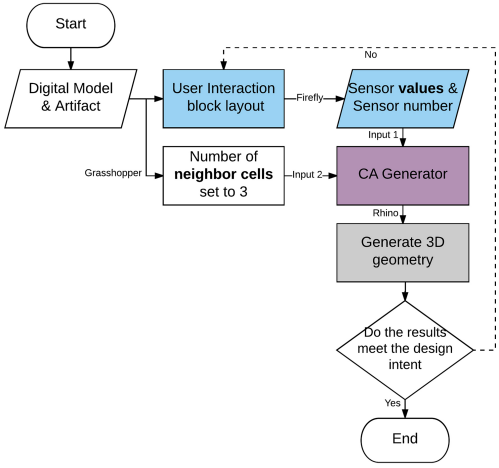


Figure 3
Artifact and digital model representing a single cell neighborhood.

The inputs for the CA algorithm in both experiments are the (1) block configuration for setting up the Seed, and (2) the number of neighbor cells for determining the center cell's state in the next generation: live or dead. The former, is set by having the designer placing the blocks at different locations using the eight cells of the artifact; and the latter, by adding or removing blocks from the artifact. The initial assumption of the work is that, the artifact will be used to provide these inputs and generate the geometric composition; as the designer manipulates these

inputs, the geometric composition will respond and the overall form will change.

Figure 4
Workflow for Experiment 1.



Experiment 1 - Creating Initial States of CA Through the TUI

For this test, the inputs are setup as follows: block configuration will be set using the artifact, and the number of neighbors is set (fixed) in Grasshopper to 3. Figure 4 shows the workflow for this experiment. In the graph, the three main components are user interaction, algorithm setup, and modeling environment. The objective is to test the artifact in providing the CA algorithm with the different cell configurations that will guide the evolution of the geometric composition in the digital model.

The Rabbit plug-in used in the visual programming tool Grasshopper is a CA generator for creating 3D geometric configurations and models. The number of cell neighbors are set for Rabbit within the Grasshopper program graph. The born and surviving parameters are set as follows: if a cell has 2 neighbors it is born, and if a living cell has a minimum of 2 neighbors it survives, otherwise it dies, in the next generation. The cells' initial configuration, will be obtained from the sensors. The physical computing system will detect which sensors are used (holding the blocks) and send that values to the corresponding parameters in the visual program in Grasshopper (Figure 5).

The values obtained from the sensors provide two types of inputs: the weight of the block and its location on the grid. This information is used to indicate which cells of the grid are used and their locations to create the custom configuration input, which will then be used as a Seed input for the Rabbit CA generator. In the case the designer relocates the blocks on the grid and the configuration will update.

Figure 5
The different cell configurations set by the designer as the initial state input for the CA generator.

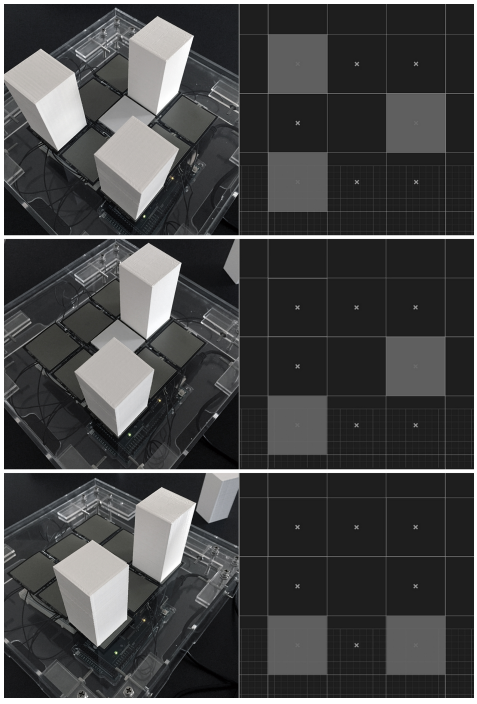


Figure 6 shows a number of the different models produced by changing the Seed, the geometric pattern's configuration is determined by the physical blocks' layout as set by the designer, and any further changes made to the blocks' layout on the grid will change the overall geometric configuration in real-time. The link created between the artifact and Rhino through

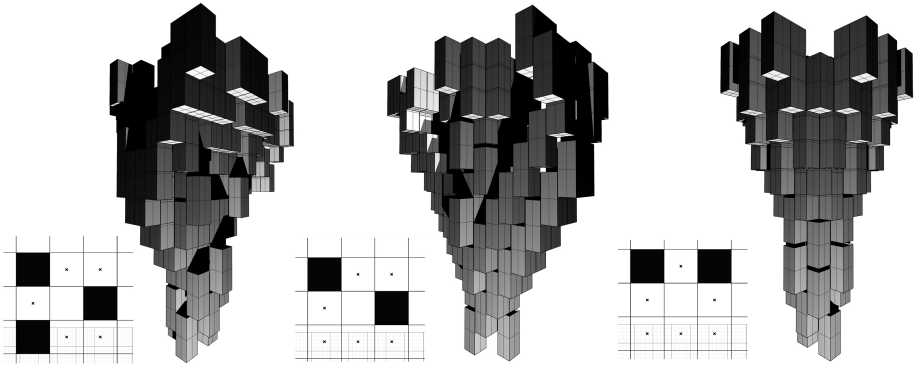


Figure 6
The three geometric models generated by changing the initial states of CA. Vertically, each level of blocks in the models is a generation of the CA process.

the plug-in Firefly enables the designers to manipulate the overall form, and provides them with a level of intuitive control over a complex geometric configuration.

In this experiment, the geometric growth is recorded over a 10 second time-lapse. Each of the generated forms are based on the blocks' configuration in the artifact. Figure 6 shows both the geometric composition generated in the Rhino scene and the initial block layout for each.

Experiment 2 - Setting Up CA Rules Using the TUI

The workflow for this experiment is similar to the previous, having all three main components. Yet, in this experiment the work will test the artifact to set the born and surviving cell rules of the CA generator, by providing the number of the cell's neighbors (Figure 7). The experiment will use the artifact to determine the number of a cell's neighbors by counting the blocks placed on the grid. As for the initial cell configuration, the experiment will utilize a random cell layout configuration defined in Grasshopper.

A 15 by 15 square grid is generated with the Seed set by a random selection of points using Grasshopper (Figure 8, left image). The rules of the evolution are set through the artifact, and tested for two conditions: (1) a cell is born and surviving if it has 1 neighbor (Figure 8, middle image), and (2) a cell is born and surviving if it has 2 neighbors (Figure 8, right image).

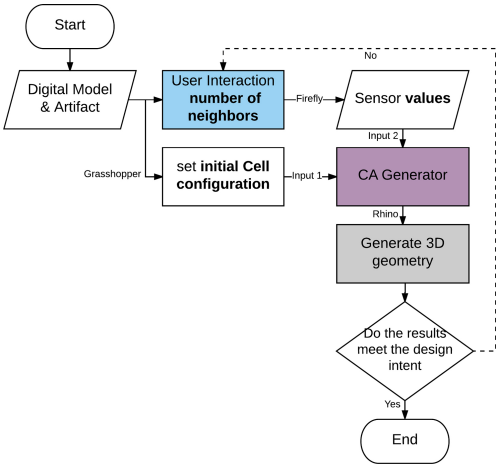


Figure 7
Workflow for experiment 2

The designer in this experiment defines the number of neighbors by adding or removing the blocks from the physical grid, and each sensor will count for one neighbor. The location of the blocks on the grid will not affect the rule. Figure 8 shows that the increase of the number of neighbors will affect the evolution process. In the case of three neighbors required for born/surviving cells (placing all three blocks on the grid), the evolution will only produce two generations, and then all cells die for the specific initial state. If further flexibility is required by the designer in the

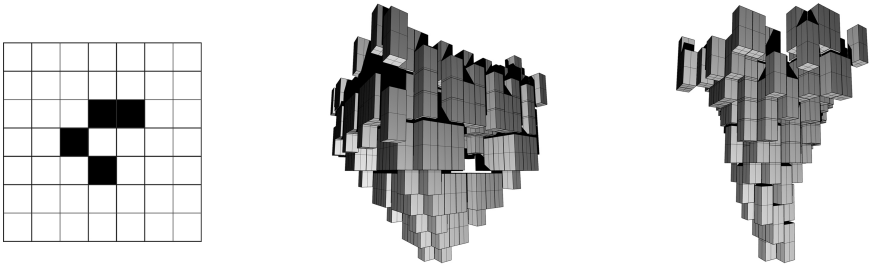


Figure 8
Left, a partial view of the 15 by 15 grid with random cell configuration; middle geometric configuration having 1 neighbor; and right geometric configuration having 2 neighbors.

process of setting up the rules using the artifact, e.g. a cell is born if it has 1 neighbor, and survives if it has 2 or 3 neighbors, this process will involve establishing the rule in Grasshopper using a conditional statement, so that the designer can interact with the artifact to manipulate the number of neighbors to meet these conditions.

DISCUSSION

The proposed method has shown the possibility to setup input parameters using the artifact: CA Seeds and number of neighbors for CA rules. The work has shown to be limited to setting up one rule per experiment, and if the designer requires to setup multiple rules in a single workflow, then a procedure must be created in the visual programming environment, e.g. in the case the designer wants to setup simultaneously the Seed and the number of neighbors for the CA rules. Furthermore, the 9-square grid used for this example limits the model's geometric evolution when compared to using the CA's infinite grid.

However, the proposed method can provide a useful tool for experimenting with CA through intuitive physical interaction between a designer and an artifact. It may also assist in educating students or designers about the principles of generative algorithms. Research claims that educators must adapt to innovative ways of design thinking in response to the development of the architectural practice (Karle and Kelly 2011). Research also indicates that there is a need to develop educational models that are more "implicit, tacit, and intuitive" to engage designers progressively in digital workflows, which generally require "explicit information which do not mostly overlap with the implicit realms of design knowledge" (Aşut and Meijer 2016). Designers using the method explained in this paper will have a hands-on experience with an intuitive interface to associate between digital models and physical artifacts for generating complex growth systems.

CONCLUSION

The present work is part of an ongoing research that investigates the possibilities of TUIs to create computational models and design object relationships. The work aims to provide a method that extends the capabilities of digital-physical workflows. Previous work developed for this topic explores tangible interaction to generate constraints for parametric models (Al-Qattan et al. 2017). The work presented in this paper expands on the topic by enabling tangible interaction to setup algorithmic rules.

Establishing design intents in computational models is usually achieved through text-based (e.g. Python, C#, API, etc.) and graph-based (Grasshopper, Dynamo, etc.) computer programming tools. The proposed method suggests an alternative approach to the previous two, through tangible interaction. Additionally, integrating generative design systems and physical artifacts for creating parametric models is a novel approach that expands the role of tangible interaction to manage and control geometric complexity in a digital design process. The experiments have demonstrated the possibility through tangible

interaction a designer can setup algorithmic rules. Further development of the work will include setting up other generative algorithms using TUIs for design purposes.

REFERENCES

Aish, R 2005 'From intuition to precision', *eCAADe 2005*, Portugal, pp. 10-14

Al-Qattan, E, Galanter, P and Yan, W 2016 'Developing a tangible user interface for parametric and BIM applications using physical computing systems', *eCAADe 2016*, Oulu, Finland, pp. 621-630

Al-Qattan, E, Yan, W and Galanter, P 2017 'Establishing parametric relationships for design objects through tangible interaction', *CAADRIA 2017*, Suzhou, China, pp. 147-156

Appleby, D and VandeKopple, J 1997, *Programming languages: paradigm and practice*, McGraw-Hill, Massachusetts

Araghi, SK and Stouffs, R 2015, 'Exploring cellular automata for high density residential building form generation', *Automation in Construction*, 49, pp. 152-162

Austin, M and QATTAN, W 2016 ''I'm a visual thinker.': rethinking algorithmic education for architectural design', *CAADRIA 2016*, Melbourne , p. 829–838

Aşut, S and Meijer, W 2016 'FlexiMold: Teaching Numeric Control through a Hybrid Device', *eCAADe 2016*, Oulu, pp. 321-328

Cruz, C, Karakiewicz, J and Kirley, M 2016 'Towards the implementation of a composite cellular automata model for the exploration of design space', *CAADRIA 2016*, Hong Kong, pp. 187-196

Davis, D 2013, *Modelled on software engineering: flexible parametric models in the practice of architecture*, Ph.D. Thesis, RMIT

Dourish, P 2001, *Where the action is: the foundations of embodied interaction*, MIT Press, Cambridge, MA

Frazer, J 1995, *An evolutionary architecture: Themes VII*, Architectural Association Publications, London

Gannon, M 2014 'Reverberating across the divide: bridging virtual and physical contexts in digital design and fabrication', *ACADIA 2014*, California, pp. 357-364

Gardner, M 1970, 'Mathematical games: The fantastic combinations of John Conway's new solitaire game "life"', *Scientific American*, 223, pp. 120-123

Herr, CM 2008, *From form generators to automated diagrams: using cellular automata to support architectural designing*, Ph.D. Thesis, The University of Hong Kong

Horn, MS and Jacob, RJK 2007 'Designing Tangible Programming Languages for Classroom Use', *TEI 2007*, Louisiana, pp. 159-162

Ishii, H 2008 'Tangible bits: beyond pixels', *TEI 2008*, Germany, pp. 15-25

Karle, D and Kelly, BM 2011 'Parametric Thinking', *ACADIA Regional 2011*, Nebraska, pp. 109-113

Klemmer, SR, Hartmann, B and Takayama, L 2006 'How bodies matter: five themes for interaction design', *ACM DIS 2006*, Pennsylvania, pp. 140-149

Krawczyk, RJ 2002 'Architectural Interpretation of Cellular Automata', *GA 2002*, Milan, pp. 7.1-7.8

Lahtinen, E, Ala-Mutka, K and Järvinen, H-M 2005 'A study of the difficulties of novice programmers', *ITiCSE 2005*, Portugal, pp. 14-18

McNerney, TS 2004, 'From turtles to Tangible Programming Bricks: explorations in physical language design', *Personal and Ubiquitous Computing*, 8(5), p. 326–337

Monedero, J 2000, 'Parametric design: a review and some experiences', *Automation in Construction*, 9(4), pp. 369-377

Muller, CL and Kidd, C 2014, 'Debugging geographers: teaching programming to non-computer scientists', *Journal of Geography in Higher Education*, 38(2), pp. 175-192

von Neumann, J 1963, 'The General and Logical Theory of Automata', in Taub, AH (eds) 1963, *John von Neumann: Collected Works*, Pergamon Press LTD, England, pp. 288-326

O'Sullivan, D and Torrens, PM 2000 'Cellular models of urban systems', *ACRI 2000*, Karlsruhe, pp. 108-116

Stavric, M and Marina, O 2011, 'Parametric modeling for advanced architecture', *International Journal of Applied mathematics and informatics*, 5(1), pp. 9-16

Vermillion, J 2014 'Physical computing without the computing: small responsive prototypes', *SIGRADI 2014*, Montevideo, pp. 643-646

Kępczyńska - Walczak, A 2014, 'The act of design - beyond the digital', *ARCHITECTURAE et ARTIBUS*, 6(1), pp. 24-28

Woodbury, R 2010, *Elements of parametric design*, Routledge , USA

Zuckerman, O, Arida, S and Resnick, M 2005 'Extending Tangible Interfaces for Education: Digital Montessori-inspired Manipulatives', *CHI 2005*, Oregon, pp. 859-868

[1] https://morphocode.com/rabbit/

[2] http://www.fireflyexperiments.com