A teaching strategies model experiment for computational design thinking[1]

RESEARCH AND EXPERIMENTATION

Selin Oktan, Serbülent Vural,
Department of Architecture, Karadeniz Technical University, Turkey
selinoktan@ktu.edu.tr
svural@ktu.edu.tr

Abstract. This study aims to share an educational model experiment for teaching computational thinking with hands-on activities. There is a gap between today's architectural education system and computational thinking. The exercises aim to fill this gap. In this study, conventional and computational design processes are not considered as two opposing poles, but as integrated processes and as a bridge between these processes.
Starting from Gagné's model, the learning process classification is reinterpreted, and the exercise processes are discussed in the titles of reception, expectancy, computation and semantic encoding, responding and creating alternatives. The outcome of this study will be a discussion on the first results, observations, and feedback from the students about the educational model attempted to be created.

Keywords: Computational design; Computational thinking;Conventional design; Parametric model; Material studies

Introduction

The idea of modernism plays an important part in today's architectural education system. However, modernist doctrines fail to discuss and explain computational design processes. Rivka Oxman (2008) says that developing technology is changing architecture evidently, and that the architectural education system must be updated to answer this transformation. The progressive architectural schools are aware that students are becoming more familiar with technology. The discourse, knowledge, practice and research must be redefined by architectural schools within the context of developing technology (Sheil, 2012), which brings computational fabrication techniques, thus increasing the importance of computational thinking. This study focuses on creating an educational model to fill the gap between today's architectural education system and computational thinking processes. The methodology is based on Bauhaus teaching methods and attempts to reconsider them within today's computational design methods. It uses various techniques, such as hands-on activities, learning by doing, and flipped classroom to teach computational thinking to undergraduate students.

Within the scope of the study, conventional and computational methods are not considered as two separate poles, but as integrated processes that support each other. For this reason, each exercise carried out as part of this study starts with a conventional method and is finalised with a computational method. The exercises are based on "hands-on activities". During the semester, four exercises and one final exercise based on various techniques and materials were performed, but only two of them are presented in this paper. One of these exercises focuses on knitting technique and fibre material, while the other concerns the form-finding technique and fabric material. The exercises were carried out with fifth year architectural students. It is important to carry out the computational thinking exercises with undergraduate students in order to make future architects familiar with the developing technology and computational thinking.

The conventional method is the first process a student of architecture learns. Within this study conventional methods, which students are accustomed to, are used as a first step for teaching computational thinking. This is important to underscore the common points between computational methods and conventional methods. A pre-test – post-test design method is conducted to measure the students' change after the exercise series. A questionnaire is administered at the beginning and at the end of the semester. The questionnaire is drawn up within the frame of the Lawshe method. As a part of the Lawshe method, questions are evaluated by six experts and the final form of the questionnaire is produced (Lawshe, 1975). The outcome of this study will be a discussion on the first results, observations, and feedback from the students about the education model attempted to be created.

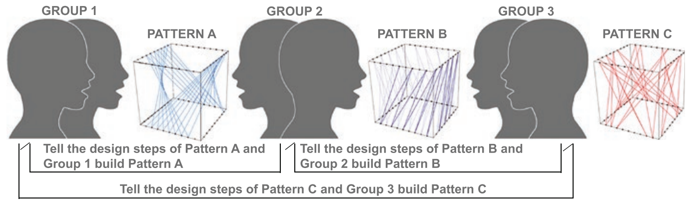
Constructivist approaches for computational thinking

The transition period in architecture, which starts with computer aided design (CAD) methods, continues today with digital architectural design (Oxman, 2008). The digital tool was previously used instead of paper, and only for presentations. However, today it has been transformed into a co-designer and a tool for defining the design process. Clark says that «Conventional CAD is just an electric pencil» (Whitney, 1990). But computational tools have more potential than an electric pencil. Hence the need to improve computational thinking. The concept of computational thinking, which is discussed as a part of this study, describes the design process step by step and with computational relations as in the algorithm.

Developments in the field of technology lead to new research areas in architecture. Similar associations between architectural education and technology occurred in the past. For example, the establishment of the Bauhaus School is directly related to the 2nd Industrial Revolution. It is a foregone result that today's technological transformation will affect architectural education. Within this context hands-on activities, a core method of this study, were used by the Bauhaus School too. In this sense, Bauhaus methods guide towards the model the research is attempting to create to teach computational thinking. Constructivist approaches in education are the common points for both Bauhaus and the exercises carried out within this study. The constructivist approach first focuses on 'knowing how' and then on 'knowing that' (Schoenfeld, 1987). In this aspect it is believed that the constructivist education process is related to computational thinking. The progressive architectural schools in the field of computational design, such as MIT, IAAC, etc., which also adopt constructivist approaches, prefer to carry out their studies with graduate level students. For example, Neil Gershenfeld's famous class called 'How to Make Almost Anything' is only open for graduate level students (Gershenfeld, 2008). This study tries to teach computational thinking at undergraduate level, and aims to do so by learning by doing exercises, which are closely related to hands-on activities. 'Hands-on activities in the learning process make the students focus even more. In this process students face problems about details and try to find a solution. As a result, the students learn how to do it. For this reason, hands-on activities lead to constructivist learning.

ISSN online: 2239-0243 | © 2021 Firenze University Press | http://www.fupress.com/techne
DOI: 10.13128/techne-10701

01 | The conventional process of knitting exercise



Hands-on activities

Learning by doing (Dewey, 1938; Özkar 2007), bringing theory and practice together (Tzonis,2014), are the concepts reconsidered in today's architectural education system after Bauhaus. John Dewey (1938) emphasises that learning by doing in the conventional design process is insufficient, and doing and experiencing must be at the core to personalise the learning process. Dewey's claim gains importance today to bring theory and practice together. It is also important to direct the student's attention to the course content. This study aims at developing new teaching methods. Within this context, Dewey's (1938) determination is taken into consideration and the exercise processes are integrated with hands-on activities.

The hands-on activities were also a part of Bauhaus' education system, and learning by doing was at the core. Bauhaus was aimed at graduate students who are familiar with the new technology. Within this perspective, both this study's aim and Bauhaus School's aim coincide. Both Dewey's (1938) vision about learning by doing and the hands-on activities of the Bauhaus School are considered as previous experiences for this study. The study attempts to integrate such experiences with computational design thinking. The study develops an educational model to improve computational thinking.

A model experiment for teaching computational design thinking

The model to be created for teaching computational thinking consists of two main phases. In the first phase, a design problem is assigned to the students and they are expected to apply the solutions to this problem. This stage is usually implemented on a model scale. The first stage is carried out with conventional methods and the computational relations of the design process are questioned in this stage. In the second stage, a similar process to the first stage is carried out in the digital environment by means of the parametric model. At this stage, students discover that the digital and the conventional design processes are complementary processes.

Gagné (1985) classifies learning conditions of cognitive processes as reception, expectancy, retrieval, selective perception, semantic encoding, responding, reinforcement, retrieval and generalisation. Within this context, starting from Gagné's model, the learning process classification is reinterpreted at the intersection of conventional design approaches and computational thinking:

- reception: defining the design problem;
- expectancy: detailing the design problem;
- computation: defining the computational relations about the design problem; design phases are defined step by step as in the algorithm; the black box of the conventional process thus becomes explicit;-
- semantic encoding: students are told how to create a parametric model for a similar process; the flipped classroom model is applied in this process; within this context, homework is done at school and a lecture is given as homework;
- responding: students recreate the parametric model within the context of computational relations defined by themselves;
- creating alternatives: the parameters of the parametric model are redefined, and the alternative models are created. If the student can create meaningful alternatives that answer the design problem, this shows that the student has successfully learnt the computational modelling process.

Reception, expectancy and computation phases are carried out in the conventional design process; semantic encoding, responding and creating alternatives phases are developed in the computational design process.

Conventional process of the exercises

The design process is shaped by the designer's subjective approach to the problem. In the conventional design process, the designer is not obliged to explain his/her decisions with the related reasons, which is called the black box of design process. On the other hand, in the computational design process the design process itself is designed. For this reason, every step of the design process should be explained. For computational thinking to become part of the design process, the designer must be able to explain even its intuitive steps. Within this context, the computation phase, which is performed in the conventional stage, is the base for the model to be created.

As a part of this study, the exercise processes which focus on various making techniques and materials are discussed in point of learning process classification. Within this context, the first exercise was carried out with fibre material and knitting technique. The first phase of this exercise is reception. In this phase the students were asked to design a pattern in the given 30 cm3 cube volume. The fourteen students worked in pairs. The second phase of the learning process is expectancy. In this phase the details about the design problem were given to the students. At first, the students drew their pattern designs as a perspective view and wrote down their design process step by step, including the computational relations.

The third step of the learning process is computation. As a part of this study, the computation process is performed as writing a kind of G-code manually. G-code, in the digital fabrication process, is a language that provides human – machine collaboration. Every machine has its own coordinate system and travels according to it. In the study, the students were asked to describe every step of the design process of creating their patterns in a way that everyone could understand. The logic of G-code was thus taught. Each group tried to apply the pattern of another group (Fig. 1). Thus, the success of step-by-step description could be measured.

In the computation process, the general approach followed by the students was to tag the points with numbers or letters. A coordinate system was thus created, as in G-code logic. Then, every move made when designing the pattern was explained with this system (Tab. 1)

02 | The final products of conventional design process

Tab. 01 | The rules and images of the patterns

Tab. 02 | The rules and images of tensile structure exercise

A Model Experiment for Teaching Computational Design Thinking

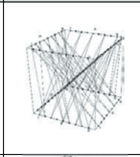
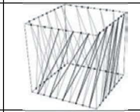
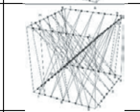
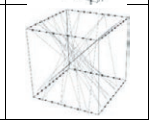
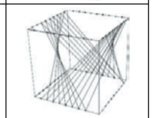
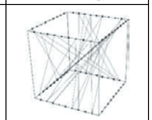
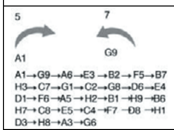
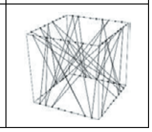
In the process of computing the conventional design process, it can be said that the students used repetitive movements, mathematical expressions or the computational equivalent of design steps. The design rules defined by one group were applied by another group. The final products are the same as the perspective drawings. This shows that the exercise process was successfully accomplished (Fig. 2).

The second exercise carried out as a part of this study was performed using the form finding technique and fabric material. In the reception phase, which is the first phase of the learning process, the design problem was described as designing a tensile structure in the 20 cm^3 cube volume. In the second phase, expectancy, the design problem was detailed. Within the context of the design problem, a five-face closed volume was prepared by dividing the surface into 1 cm grids. The grids were used in the computational phase of the conventional design process. In this phase the tension points were defined on the fabric material and tagged by numbers. The tension points on the fabric and the anchor points on the surfaces were matched. The parameters of the form finding process were thus defined (Tab. 2).

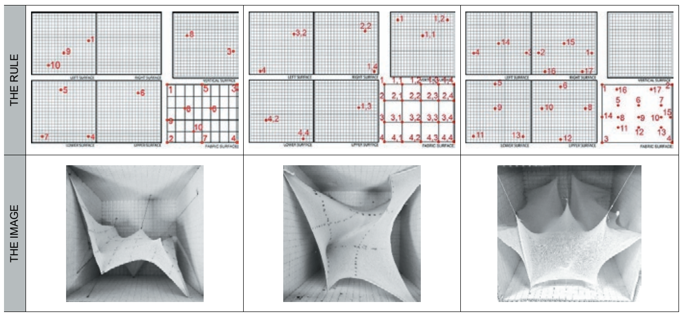
The final conventional phase of the exercise was completed by creating the first answer for the design problem and by applying the answer integrated with the hands-on activities. As a result of the conventional process, the design product which would be modelled parametrically was created. The first stage of the exercises, which are carried out with conventional methods, also involves a preparation for the computational design process. The link between the conventional process, which students are familiar with, and the computational design process is established at this first stage.

Computational process of the exercises

Computation means to formulate problems to represent them as steps or algorithms (Aho, 2012). The computational design process focuses on computational thinking. It is almost completely explicit, unlike the conventional design process. It also can be defined as designing the design process. The second stage of the exercise processes are carried out with the computational design process. In this stage a parametric model is achieved which, based on the information collected, forms the computed conventional design process.

THE RULE	THE IMAGE
On the first edge of A plane, begin with an odd number and in every step add 3 to it, On the fourth edge of B plane, begin with an even number and in every step add 3 to it. A=[1,2,3,4] 1 2 3 4 ⇕ ⇕ ⇕ ⇕ B=[5,6,7,8] 8 7 6 5	
From left to right (for the beginning point); the number of bottom plane + 4 = the number of top plane From right to left; the number of top plane + 4 = the number of bottom plane	
Skip one point in every step. From A9 to B9 odd numbers are paired: (1-1) (1-3) (3-3) (3-5) (5-5) (5-7) (7-7) (7-9) (9-9) From B2 to H1 the empty points are paired.	
In the first round; From A plane to B plane → x + 4 From B plane to A plane → x – 1 In the second round; From B plane to A plane → (x+1), (x+2),…, (x+n) From A plane to B plane → (x+(n-1)) n= the first point of the new round	
Starting from the first point of A edge, in clockwise direction pass from one edge to another and add 1: A - 1 → F - 2 → C - 3 A1 – F2; F2 – C3; C3 – H4; H4 – A2; A2 – F3; …	
If you are on A go to G; If you are on G go to C; If you are on C go to E; If you are on E go to B; If you are on B go to H; If the point is an odd number then; point number + 3 (mode9) If the point is an even number then; point number + 5 (mode9)	
	

| Tab. 01

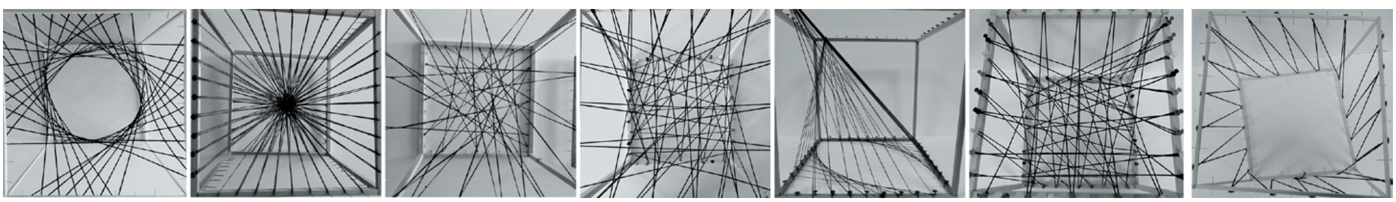


| Tab. 02

Grasshopper, which is a plug-in for Rhinoceros, was used as the parametric model tool. Semantic encoding, responding and creating alternative phases are performed in computational design stage as a continuation of the learning process.

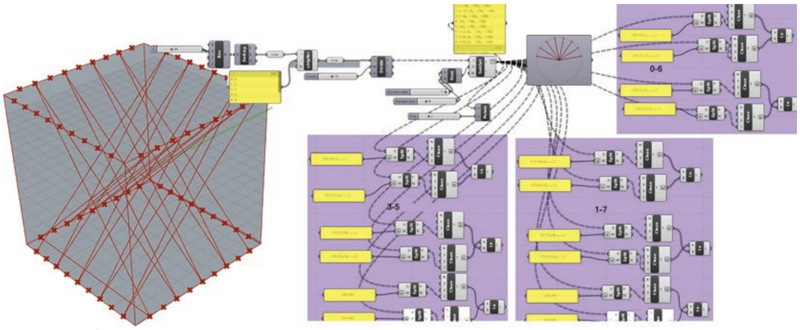
The computational design process of knitting exercise, which is the first exercise of this study, begins with semantic encoding. In this phase students are taught how to create a parametric model of a similar design product. The modelling process is carried out in a similar way to the conventional design process. The model is

02 |



03 | The Grasshopper definition of knitting exercise

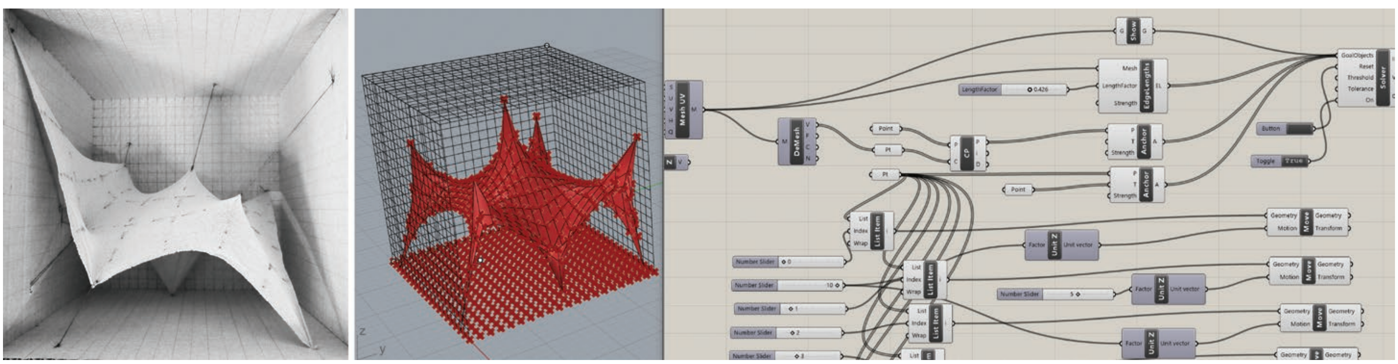
04 | The final products of conventional and computational processes



created by matching the set of points placed on the cube's edges against each other (Fig. 3).
In the responding phase, students try to redo the parametric model by themselves. In this phase the data collected from the conventional design process is used as the parameters of the parametric model. Thus, the bridge between conventional and computational design processes begins to be established.
The responding phase is followed by the creating alternatives phase, which is the last phase of the exercise process. One of the most important goals of computational design is to be able to create a design space, which includes lots of design alternatives. If the parametric model is structured in a proper way, creating alternatives for the design problem will take a very short time. At the end of the creating alternatives phase, the design products' success and its suitability with the design problem show that the parametric model was able to be created in a proper way. The students, who created proper models, also achieved the computational design process.
The second exercise process depends on form finding study. In the semantic encoding phase, students are shown how to model the material behaviour parametrically. In the modelling process, the anchor points on the cube's faces and the tension points on the fabric are matched; these points were computed in the computational design process. This exercise process teaches students a multi-directional approach about computational design, such as how to manage the data, to compute material behaviour, to model the external forces affecting the design system.
In the responding phase, the students try to precisely redo the parametric model phase conducted by the lecturer. In the creating alternatives phase, they try to model the design products, which were created as a result of the conventional design process. Within this process the parameters are redefined as they are computed in the conventional process. In this model redefining the parameters of tension points, anchor points and the data about material behaviour enables alternative models to be created (Fig. 4).
Making the computational data a part of the design process is an output for the performed exercises. As a result of the exercise processes the students were able to experience designing the design process as integrated with hands-on activities. Thus, the initial data about the educational model to be created were collected.

Results

It is widely believed that conventional and computational design processes are in conflict and differ from each other. This study focuses on making computational thinking a part of architectural education. A section of the performed exercises was shared to discuss this. The study attempts to create a unique educational model, which brings the conventional design process, hands-on activities and the computational design process together. These exercises are defined with a spirit recalling the Bauhaus-Industrial Revolution relation. Constructivist approaches have a place in the exercise processes.
The exercise processes start with conventional design methods and are finalised with computational design methods. The hands-on activities and learning by doing are at the core of both methods. This makes the learning process more understandable for the students. Instead of computational design education that only depends on teaching a software, this educational model focuses on turning design process into data and on using the data collected in the computational design process. It was observed that with these exercises, a bridge between conventional and computational design processes can be established.
As a result of the exercise processes, it can be said that the students easily understood the logic of computational thinking because the exercise process was integrated with the conventional methods, which the students are familiar with. Hands-on activities made the students concentrate on the design problem, and their atten-



tion on the course content increased. In addition to this, the flipped classroom model has a positive effect on learning Grasshopper/ Rhinoceros modelling because the students had a chance to watch the tutorials prepared by the lecturer repeatedly and to practice by themselves.

As a part of the pre-test – post-test design evaluation method, a questionnaire, which is drawn up within the frame of the Lawshe method, was conducted with the 14 students taking the course. The results of the questionnaire revealed that the students were already aware of the meaning of computational thinking and of the methods of parametric design even before the course. This shows that students, who are familiar with technology since their childhood, can easily adapt the computational design and fabrication processes. The number of students who thought that computational design ability and digital fabrication process are mutually relevant increased from 28% to 72%. This shows the students could identify the relationship between computational thinking and digital fabrication process. In the conventional design stage of the exercises the problem was deeply analysed, and the computational relations were defined. This process shows that conventional design processes can be performed in a completely explicit way. Carrying out the computational design process after that stage minimises the randomness of the design process because, to define the parameters of the design process, the students start thinking deeply about the design product and try to explain the design steps by themselves.

The computational design process has not emerged suddenly with no reason. It has emerged as a reflection of developing technology and to meet the needs of new fabrication techniques. It is crucial to catch up with developing technology by improving computational thinking skills. The design and fabrication processes are needed to be computed to provide human–machine cooperation. Both the logic of a machine language like G-code and creating an algorithmic definition of a design process are experienced within this exercise series. As a result, making computational thinking a part of the conventional processes of architectural education deserves deeper consideration, and hands-on activities must be at the core of the education system.

NOTES

[1] This paper has been converted from the ongoing PhD thesis.

REFERENCES

Aho, A.V. (2012), "Computation and Computational Thinking", *The Computer Journal*, Vol. 55, n. 7, pp. 832-835.

Dewey, J. (1938), *Experience and Education*, Kappa Delta Pi, Indianapolis.

Gagne, R. (1985), *The Conditions of Learning*, 4th ed., Holt, Rinehart & Winston, New York, USA.

Gershenfeld, N. (2008), *Fab: The coming revolution on your desktop--from personal computers to personal fabrication*, Basic Books, New York, USA.

Lawshe, C.H. (1975), "A quantitative approach to content validity", *Personnel Psychology*, Vol. 28, pp. 563-575.

Oxman, R. (2008), "Digital Architecture as a Challenge for Design Pedagogy: Theory, Knowledge, Models and Medium", *Design Studies*, Vol. 29, pp. 99–120.

Özkar, M. (2007), "Learning by doing in the age of design computation", *Proceedings of the Computer-Aided Architectural Design Futures (CAADFutures)*, 2007, pp. 99-112.

Schoenfeld, A.H. (1987), *Cognitive science and mathematics education*, Routledge, Abingdon, United Kingdom.

Sheil, B. (2012), "The digital generation", in Spiller, N. & Clear, N. (Ed.), *Educating Architects: How Tomorrow's Practitioners will Learn Today, Thames & Hudson*, London, pp. 138-144.

Tzonis, A. (2014), "A framework for architectural education", *Frontiers of Architectural Research*, Vol. 3, pp. 477-479.

Whitney, D.E. (1990), "Designing the design process", *Research in Engineering Design*, Vol. 2, pp. 3-13.