ITU A|Z • Vol 14 No 1 • March 2017 • 1-11

An investigation on the attitudes towards adopting the design by coding paradigm

Bülent Onur TURAN[1], Kemal ŞAHİN[2], Ümit IŞIKDAĞ[3]
[1] boturan@gmail.com • Department of Informatics, Mimar Sinan Fine Arts University, Istanbul, Turkey
[2] kemalsahin@gmail.com • Department of Informatics, Mimar Sinan Fine Arts University, Istanbul, Turkey
[3] uisikdag@gmail.com • Department of Informatics, Mimar Sinan Fine Arts University, Istanbul, Turkey

Received: March 2016 • Final Acceptance: November 2016

Abstract

The recent developments in the digital technologies have led to an increase in the parametric modelling efforts and the use of parametric patterns in design. In order to manage this new paradigm where the design is becoming more digital, the designer needs to utilize his/her analytical abilities in an effective manner as this is the only way for precise representation of concepts in the actual design itself.

The parametric design is a holistic process. A key element of this is the development of design through coding. In this context, the aim of the study was understanding the attitudes of architects and architecture students towards adoption of analytical/algorithmic methods and specifically the object oriented paradigm and design-by-coding. In the first stage a randomly selected group of volunteers were provided training in object oriented programming concepts and design-by-coding over 15 weeks (and a total of 45 hours). Later in this study two data collection tools were utilised to understand the attitudes of the participants towards design-by-coding. The first tool used was an attitude scale, while the second tool used was a web based questionnaire survey. The results indicate that the participants did not show a significant positive attitude towards design by coding. This result might have been caused by the difficulties faced during the study, as the participants were experiencing a design-by-coding exercise as the first time. In fact, it is also observed that algorithmic thinking and parametric design abilities of the designers can be enhanced by training on design-by-coding concepts.

Keywords
Design, Coding, Parametric modelling, Architecture, Attitude.

doi: 10.5505/itujfa.2017.18209

1. Introduction

Design can be regarded as a process oriented problem solving activity. In this activity alternative solutions are proposed, evaluated and re-formulated through the process in order to arrive at a final design proposal. Computer Aided Design (CAD) can be defined as the use of computational tools to facilitate the design activities and to support the capabilities of the designer. CAD systems are used in every stage of the design, for instance to produce conceptual design in early stages, for the generation of design information and documentation, for the visualisation of information in order to facilitate the discussions on design with 3rd parties (such as other architects, engineers, contractors and the owner), and in later stages to produce final design, blueprints and presentations (Schmidt and Wagner, 2005).

Parametric design approaches that emerge together with the advancements in digital technologies along with the developments in rapid prototyping and 3D printing technologies, have triggered major changes in design and development of architectural shapes. These changes in design mediums articulate the changes in the design process itself. Today many designers concentrate more on parametric design, and the focus today is on the design of the process itself rather than the design of the final form.

Traditional architectural education focuses on enhancing the understanding of students through visual perception. On the other hand, parametric design which is viewed as the key paradigm shift requires that the designers have the ability of algorithmic and analytical thinking to build and interpret the relations between the objects.

This study focuses on understanding the attitudes of architects and architecture students towards adoption of analytical/algorithmic methods (and specifically the object oriented paradigm and design-by-coding). The adoption of these methods (as mentioned in Section 4) would in turn contribute to their -parametric design thinking- and enhance their design capabilities.

In traditional architecture education, design through the use of coding/programming or visual coding has not played a central role, and coding is usually learned by personal exertions.

In this study, the authors provided object oriented programming training to a randomly and voluntarily formed group of designers with the focus on enhancing their parametric thinking capabilities. The attitudes of the trainees towards the adoption of analytical/algorithmic methods (and specifically object oriented paradigm and design-by-coding) were then analysed at the end of the training effort.

The results of the study provide key findings on whether training in analytical/algorithmic methods (and object oriented paradigm) would be beneficial to design education in an era where parametric design thinking is becoming a key design capability.

2. Coding the design

The act of design has been defined by different researchers as a “Problem solving process” (Newel and Simon, 1972), “Cognitive task” (Akın, 1986), “A reflection in action” (Schon, 1987), and as “Knowledge based activity” (Coyne et al., 1990). In light of these expert views and definitions this research interprets the act of design as “a decision making process and a method for organising information towards providing a solution”. From the architectural point of view, the solution provided by the design is related to spaces, forms and the order of these.

Many researchers agree that the design (as a process and as an action) has three core stages: analysis, synthesis, and evaluation. These three stages can be defined as decomposition of the problem into several parts, and unification of these parts with a new interpretation and testing of the final product.

The interpretation of design as a cognitive task reveals other definitions of design. For example, Jones (1992) defined design process as both a “glass” and a “black box” process based on the cognitive method followed. In the “black box” process, most of the design is completed in an intuitive way which cannot be expressed with clearly defined tasks in the mind of the designer. In this approach the input and output of the process is explicit but what is happening inside the mind of the designer remains as a mystery. In the "glass box" process, design is accomplished through steps which can be clearly defined by the researcher. In this method the process is well defined and the details are explicit along with the input and the output.

It is already known that the design process has been defined as a series of steps (i.e. in an algorithmic way) even before the digital technologies came into practice. The algorithmic thinking, which is advocated by designers following the glass box method, later formed the basis for many computational design concepts such as Computer Aided Design (CAD), Computer Aided Architectural Design (CAAD), Digital Design, Computational Design, Parametric Design, Generative Design and Relational Design.

3. Parametric design & modelling

Parameter is a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation (oxforddictionaries.com, 2016). In mathematics and computer science a parameter is interpreted as a symbol that denotes a quantity. In mathematics a parameter is used to define a (usually unknown) variable and its variability range, and in computer science it points to an address of a memory block where the value of a variable can be stored. Parametric design is focused on making the design by utilising parameters. As design process can be regarded as a problem solving process, it would be better to elaborate on "the interpretation of design as a problem" before moving on to the parametric design and modelling in detail.

Rittel and Webber (1973) classified the design problem in two dimensions. The first one is a tame problem where the scope which have been focused on is clear. Examples include a mathematical problem such as solving an equation; or the task of analysing the structure of an unknown compound. The second one is a wicked problem which has no definitive formulation. Wicked problems are observed in public decision making such as deciding on the location of a freeway, or a modification to the school curricula. The design problem can also be defined as wicked problem as it is impossible to define the problem by its all variables and parameters.

In parallel with many researchers in the field Lawson (2006) views the architectural design as a process of understanding and problem solving. In this context the solution of the design problem (i.e. the generation of the design) can be viewed as the process of choosing the best alternative from a set of options based on designer's priorities. In architecture these priorities can be objective or subjective as the characteristic of architectural design is situated in the middle of science, art and technology. The priorities can diverge between scientifically rational or irrational alternatives for each designer. A holistic approach is necessary for a design problem solving. In parametric design, the designer should step back from his/her "traditional designer" identity, and, rather than proposing a direct solution to the design problem, the designer needs to focus on generation of a relationship network that would in turn morph into the actual design itself. This is also mentioned by Woodbury (2010) as; "*…rather that the designer creating the design solution (by direct manipulation) as in conventional design tools, the idea is that the designer establishes the relationships by which parts connect, builds up a design using these relationships and edits the relationships by observing and selecting from the result produced*".

In this paper, the design is considered as a process which includes interpretation of the problem, determination of alternatives, and an exploration based on (evaluation of and decisions on) the alternatives. The "parameters" in this context are assumed as tangible factors that will help in defining the system which generate alternative solutions to the design problem and its limits. From this perspective the parametric design can be defined as "the process of defining and solving the design problem using a set of tangible variables."

Parametric design enables the evaluation of numerous alternatives which would be impossible to evaluate without the use of this approach, and also facilitates the proposal of the satisfactory solution to the design problem. The aim of the parametric design is proposing many alternative solutions to the design problem as a result of systematic thinking. In addition, small variations in the values of the variables (stored within the parameters) such as metrics, angles, colours, would have an impact on the overall project due to the algorithmic structure of the parametric design and this structure also makes the design/designer less error prone (Woodbury, 2010).

A parametric model can be regarded as a virtual representation of a solution to a design problem. This representation is composed of attributes having constant and variable values and the representation appears as a system of dependent and independent variables. From this perspective parametric modelling can be defined as a process of generation of the geometric representation of the design (i.e. the virtual representation) based on the attributes and components which are defined with parameters.

The traditional design has advantages in the beginning as it is easy to generate an initial model in this approach. In traditional design it is not necessary to build relations that would have an impact on the overall model. In fact, traditional approach can become problematic in later stages when a component would need to be added or removed manually to the design. In contrast to the traditional design approach, in parametric design the designer would not focus on changes that would affect the final product, but would focus more on the processes and building relationships between components. Later the designer can; redefine these relationships, change the values of the variables and generate many alternatives. These alternatives are then evaluated in order to come-up to a final product (i.e. the final design). This approach requires a new way of systematic thinking which was not thought to be the part of the design process previously. In this approach the designer would need to take a step back and think about all relations and reflect them in the model. The initiation phase of the parametric design can be thought as being of more cost when compared with traditional design, as the initiation of the parametric design requires all relations to be correctly defined in order for the parametric model to be successful. In fact, as mentioned by Woodbury (2010), as the system is built upon the relations, in every change that would occur, the design would transform into new alternatives without disrupting the coherence of the design, and this would increase the ability of the designer in exploring different alternatives.

4. Coding

Coding can be defined as a process of declaring instructions to process raw inputs and transform them into to meaningful outputs (i.e. information). Coding also helps in building relations between the newly generated information and information that is residing within the system (Kernighan, 1999). In this context coding can be defined as a process for providing a meaning to data. Coding languages are developed to help in making the coding process more clear and easy to understand. The code needs to be rationalized and systematically constructed before being physically implemented in the digital world. Algorithms are used to express and implement the systematic construction of coding. As indicated by Chabert (1999) an algorithm is the art of articulating a process as a series of steps based on a data model. Once an algorithm is constructed, it is then coded in a programming language. Algorithms are used in defining the stages and conditions, in order to solve a problem or for completing a set of tasks in a process. Kolarevic (2015) indicated that conditions and statements defined in algorithms can be used to guide the design and generation of geometric forms. In this way, unique geometric forms can be discovered. Large datasets can be used to generate and derive different design alternatives in algorithmic approach. Then, the decision making process would also be facilitated through the use of these alternatives. The advancements in visual programing languages allowed the rapid implementation of algorithms, which are developed by the designers, in program codes. Today, the rich user interface of and advanced interaction opportunities in visual programing languages, has made coding more flexible (Jonhston, 2004). Thus, it is now possible to generate design by coding through utilising user friendly coding languages and environments.

5. Attitude scales and coding in design

According to Kagitcibasi (1998), social psychologists define attitude as "a tendency imputed to a person who systematically forms the thoughts, feelings and behaviour of that person with respect to a physiological entity". Several methods are developed to measure the attitude. One of the commonly used methods is utilising Likert scales. Likert type scales provide ordinal data which would provide structured information, and facilitate the evaluation of attitudes. Thus, as Cohen et al. (2007) indicated they are frequently used in determining the attitude.

As mentioned by Heersink and Moskal (2010), there is no standard attitude scale for measuring the attitude of users in learning computer programming and coding. In fact, researchers in the field developed their own attitude scales to measure attitudes towards learning coding practices. For example, Wiebe et al (2003) developed a ten-item five-level Likert scale for measuring attitude of programming students in terms of self-confidence and motivation. Later, Hutchinson et al. (2008) developed their own scale to measure the attitude of students towards programming. Another scale developed by Hongwarittorrn and Krairit (2010) measured the attitude of learners in terms of 6 main topics and by using 50 items. Some developers of these scales were influenced by items in the scale developed by Fennema-Sherman (1976). An example of this can be found in Wiebe et al (2003), where authors developed their "Computer Science Attitude Scale" based on Fennema-Sherman scale. The scale of Wiebe et al (2003) consisted of 5 main topics and 57 items. The scale measures attitudes towards adopting computer science and programming concepts, mainly focusing on the programming side. Baser (2013) has utilised the scale of Wiebe et al (2003) and proved the reliability and validity of the scale. In this paper, in order to measure the attitudes, the authors utilised the scale of Wiebe et al (2013) as in the form that is implemented in Baser (2013). The act of design in this research is interpreted as a method for organising information towards providing a solution, and also a process towards finding a solution to a problem. In this context, Algorithmic Thinking and Coding can be viewed as value adding concepts for design. As algorithmic thinking is viewed as a pool of abilities that are connected to constructing and understanding algorithms such as:

- the ability to analyse given problems
- the ability to specify a problem precisely
- the ability to find the basic actions that are adequate to the given problem
- the ability to think about all possible special and normal cases of a problem

Algorithmic thinking has a strong creative aspect: the construction of new algorithms that solve given problems (Futschek G., 2006). Understanding Coding as a method of abstraction towards developing a reasoning about a problem and its solution, would have a positive impact on enhancing the design capability of an individual. Therefore, it would have positive impact on abilities related to analysis of problems, understanding multiple dimensions of a problem and providing a solution.

6. Research context

The research was focused on how training provided on object oriented programming concepts would have an impact on attitudes of learners towards design-by-coding thinking. In this study a novel research methodology is proposed and implemented. The research was completed in 2 phases. In the first phase an experimental study has been conducted together with the learners. The second phase of the research covered data collection, analysis and evaluation processes which are completed by the researchers.

In the first phase of the research a randomly selected group of volunteers have been trained in object oriented programming concepts and in design by coding practices over 15 weeks (and a total of 45 hours). In the training sessions, firstly the theoretical aspects of parametric design have been introduced to the participants. In this phase subjects related with Fundamentals of Parametric Design, Parametric Modelling, Algorithmic Thinking, Designers' Role, and Patterns for Parametric Design were introduced to the students and group discussions on these subjects have been conducted. Furthermore, algorithmic analyses were conducted on parametric patterns found in examples from architectural design.

In the second phase of the research both qualitative and quantitative methods are utilised. The quantitative method included an experimental study; the qualitative method included a content analysis. In the experimental study participants have been provided hands-on training in a Java based programming language called "Processing". The language was then used by the participants for generating 2D and 3D parametric designs through design-by-coding approach. Mentor support was provided to the participants at this stage. The participants were required to generate designs using parametric patterns with the limitation of using a maximum of 3 parameters. This limitation is used as a balancing factor for preventing the variances of the attitudes from one participant to another. Three of the design outputs produced as a result of design-by-coding experiment were as follows. Figure 1 provides a result of a parametric pattern generation exercise. In this case, the generated pattern is of a recursion type. In this example, the design is generated by iterations on X and Y axis.

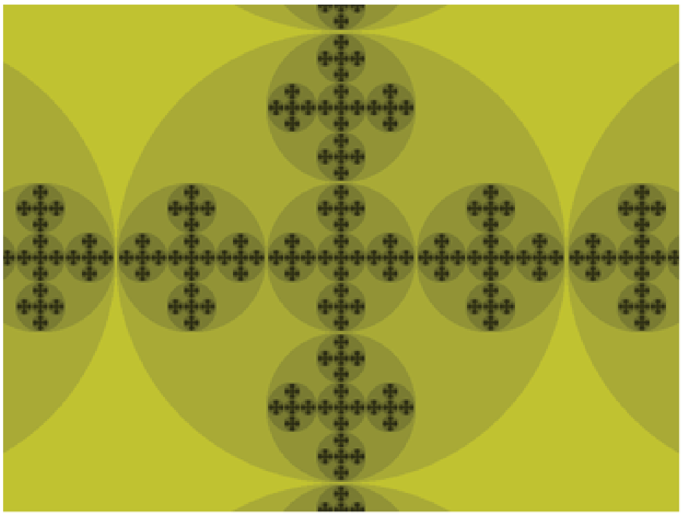

Figure 1. *Recursion pattern implementation.*

Figure 2 illustrates a pattern generated by using force field / attractor point approach. In this exercise the design is formed around an attractor point.

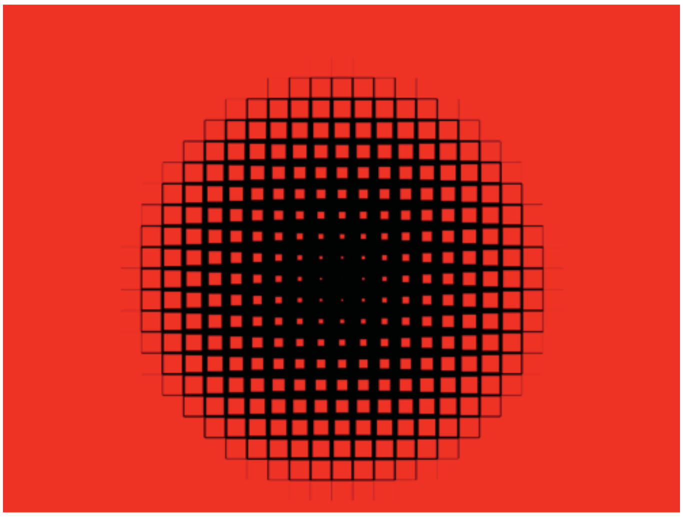

Figure 2. *Attractor point pattern implementation.*

Figure 3 depicts a pattern generated by the tiling approach. In this specific exercise the pattern is formed by tiling up a generated pattern within a pre-determined geometry (in this case a square).

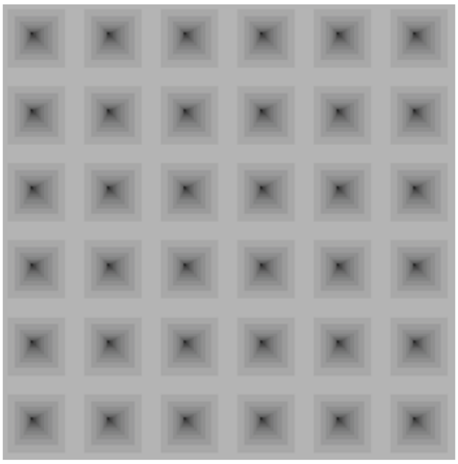

Figure 3. *Tiling pattern implementation.*

6.1. Research questions

The research aimed understanding the attitudes of learners towards adopting design-by-coding thinking by adoption of the object oriented programming paradigm. The research questions developed within this context are as follows:

- What is the attitude of learners (who are currently studying or have studied the architecture curricula) against computer programming?
- Do the learners develop self-confidence related to their parametric modelling skills while modelling through programming? Does learning programming support self-confidence?
- Do learners find design by coding useful?
- Do learners think that they will be successful in design through programming?
- Do learners have an intention to use coding in design, and parametric design through coding- in their business practices?
- Does gender play a role in the attitude of learners?
- What would the social impact of success be if learners would adopt design-by-coding paradigm?

In addition, although some examples from the generated design outputs are provided in the previous section, it needs to be noted (as a limitation) that, this research did neither concentrate on evaluating the design outputs, nor focused on assessing the role of design knowledge on enabling better design outputs. In essence, this research is conducted with a narrower focus, i.e. to understand if the "design-by-coding thinking" can be promoted through provision of training on related concepts and coding.

6.2. Sampling method

The sample used is a randomly selected voluntary group of designers who are currently studying or have studied the architecture curricula. The key selection criterion for the sample was their lack of knowledge and experience regarding programming and parametric design. The participants were chosen considering that they did not have any knowledge on computer programming and parametric design prior to the experiment. As the participants did not have knowledge on programming and parametric design, support and mentoring had to be provided during the experiment. Thus, due to i.) the nature of the experiment and ii.) the face to face mentoring provided during the study, the number of the participants has remained low (as 10), based on the limitation (and low number) of mentors (2), who participated in the experiment.

The "Processing" language was chosen as the main instrument to teach object oriented programing concepts and parametric modelling. The reason behind this choice was, its object oriented nature, the popularity of the language in generation of geometries and its ease of use. In addition, the use of "Processing" language is advised by digital design and digital arts professionals.

6.3. Data collection tools

In this study two data collection tools were utilised. The first tool was an attitude scale implemented in Baser (2013). The reason behind the selection of this tool was that the reliability and validity of this scale was tested and approved previously (Baser,2013). Second tool used was a web based questionnaire survey.

7. The attitude scale

The attitude scale utilised in the study is originally adopted by Baser (2013) from the attitude scale of Weibe which is mentioned in the previous section. The reason behind using the scale of Baser (2013) in this study was that the validity and reliability of that scale was tested and proved by the adopting author. In addition, this scale mainly focuses on measuring the attitudes of participants related to programming instead of measuring their attitudes towards general IT understanding which is the focus of other related scales. The scale of Baser (2013) consists of 40 items. These items were categorized in 4 different factors in parallel with research focus.

These factors were:

- F1: Self Confidence and Motivation

- F2: Value of Programming
- F3: Attitude towards achieving success in parametric design through programming
- F4: Social perception of -success in programming-

Following the training on the Processing language and exercises on adopting coding-by-design concepts, the attitude scale was applied with face to face interviews. All of participants filled a form containing the attitude scale at a research meeting. The attitude scale used was similar to a 5-level Likert scale, (1) indicating the complete disagreement with a positive attitude and (5) indicating the complete agreement with a positive attitude.

8. Questionnaire survey

A web based questionnaire survey is implemented following the attitude scale exercise. The questionnaire consisted of 2 open-ended questions which were focused on understanding the views of the participants related to training they were provided. A considerable amount of qualitative data (i.e. in the form of texts of 500 words per participant) were gathered. In the following stage, a content analysis, focused on understanding the views of the learners on the training content and material provided, were conducted.

9. Findings of the research

The survey results were analysed using statistical methods. In the first stage tests were conducted on the survey results to check the Normality of the distribution of the result variables. Following this, in order to analyse the attitudes of participants against programming a one sample T-test is utilised in the research. Table 1 summarises the results of the T-test.

Based on the evaluation of the overall scale it is difficult to state that the participants have developed significantly positive attitudes towards programming and coding-by-design, as the results of the T-test did not reveal any significant difference. The lack of significant positive attitude might be due to discomfort and lack of self-confidence of the participants who are new to programming.

In terms of the first factor (F1: Self Confidence and Motivation), a significant difference in the ratio of Self Confidence and Motivation has not been observed. The time limitations in training hours and low number of exercises for practical implementation of coding, might have prevented the development of self-confidence (related to programming) in the participants. More hands-on coding exercises would contribute more to the development of self confidence in the trainees. In addition, demonstration of successful cases of parametric design and -design by coding- can be key factors for increasing the confidence related to coding. The results of the questionnaire survey support these views. For example, participant P6 mentioned that they needed more practical examples, while participant P4 indicated the ratio of practical examples provided in the training as very low against the theoretical details that were provided during the training. P4 also mentioned that the time of training that they were provided was very limited.

The second factor F2: Value of Programming, measured if the participants viewed programing/coding as a key/valuable skill for making parametric design. The results of the T-test did not reveal any significant difference

Table 1. *One-sample T-test results.*

	Participant Background	N	Mean	Standard Deviation	Average Standard Error	T- Statistic Value	Degree of Freedom	P-Value
Overall Scale	Architecture	10	3.634	0.605	0.043	4.833	218	0.000
F1	Architecture	10	3.151	0.774	0.058	5.265	218	0.000
F2	Architecture	10	3.798	0.768	0.057	4.932	218	0.000
F3	Architecture	10	4.171	0.735	0.055	0.474	218	0.636
F4	Architecture	10	4.395	0.509	0.038	0.759	218	0.448

N: Number of Participant

related to this factor. Participants did not seem to have realized importance of coding for parametric design.

In factors F3 and F4, the attitude of participants towards achieving success in parametric design through programming and the social perception of success in programming were measured. The results of the T-test revealed significant difference in these factors. In addition, the views of the participants gathered by the questionnaire survey were indicating significant positive attitudes. The survey findings indicate that if the training time for design by coding can be increased and this training is supported with more practical examples, productive and creative parametric designs can be accomplished. For example, P4 indicated that in their architectural firm they also implement parametric design successfully, but found his current knowledge and experience as "not sufficient enough" related to the parametric design. On the other hand, this reveals that he is confident that he would be successful in making parametric designs once he has spent required level of time and effort on the subject. The design studio environment of the training sessions and coding exercise, where sharing of knowledge in a collaborative way is encouraged, had a positive influence on the social perception of success in programming. For example, P7 indicated that the trainers were very friendly and supportive and this had contributed to their learning experience and success in design by coding.

An independent sample t-test is used to evaluate the role of gender in attitude towards parametric design through coding. The Table-2 below summarises the T-test results.

The results presented in the table reveal that the attitude of female trainees towards design-by-coding was more significantly positive than the male trainees. This result might be influenced by high attendance rates of female trainees and their more active involvement in collaborative programming tasks.

The views of the trainees acquired from the questionnaire survey had a positive tendency both towards the training exercise and implementing parametric design in their projects. For example, P2 and P4 mentioned that they have learned key facts related to their career in design during the training. In addition, P4 mentioned that they have developed parametric designs in their office prior to this training, and this training helped in developing core knowledge in the field of parametric design.

The negative views of the trainees were gathered around the limitation of time allocated for the training sessions and low number of practical/hands-on examples provided during the training. P1 indicated that they needed more time for training and also need more practical examples. P5 mentioned that practical examples provided would need to have been explained in more detail.

10. Conclusions

The research investigated how a training provided on object oriented programming concepts would have an impact on attitudes towards parametric design thinking. The sample used in the experiment is a randomly selected voluntary group of designers who are currently studying or have studied the architecture curricula. They were provided training in object oriented programming concepts and design by coding. The attitude of the participants was analysed based on an attitude scale and a questionnaire survey. The findings can be summarised as follows.

The results of the attitude scale demonstrated a neutral attitude as output. Thus, it is difficult to conclude that the participants have shown a significant positive attitude towards design by coding. In fact, the views mentioned

Table 2. *Independent sample T-test results.*

Gender	N	Mean	Standard Deviation	Average Standard Error	T- Statistic Value	Degree of Freedom	P-Value
Male	5	3.552	0.636	0.062	3.897	218	0.000
Female	5	3.862	0.541	0.049			

N: Number of Participant

in the questionnaire survey were more positive. In the study it is observed that the learners are not familiar with algorithmic thinking, but it is also observed that algorithmic thinking and parametric design abilities can be enhanced by providing training on coding. The results indicate a strong positive correlation between the rate of attendance to training and positive attitude against coding. The design-by-coding exercises conducted through following object oriented programming paradigm, and visualisation of design had a positive impact on enhancing the (design) decision making abilities of students and motivating students to implement design by coding approach.

The research has shown that training will contribute to the rapid adoption of parametric design concepts by designers and thus will contribute to the implementation rate of parametric design approach. If designers and design students are encouraged to adopt object oriented concepts in all fields of design this will have a positive impact on motivation towards moving to parametric design.

The low number of participants of the research can be seen as the major barrier and limitation that has be overcome to generalize these findings back to the target population of the study (designers who are currently studying or have studied the architecture curricula). In order to improve the external validity of the research, a further study will be conducted with more mentors and higher number of participants in the near future.

The future research in the field will focus on the following subjects.

- Measuring the attribute of learners against design-by-coding following a two phase training which covers both graphical and parametric design concepts.
- Enhancing the training to cover more practical implementation and coding exercises.
- Enhancing the training using more interactive tools and environments.
- Providing training based on real life cases that are being designed and implemented in projects.
- Finding methods towards increasing the motivation of learners towards implementing parametric design through coding.
- Investigating the relationship between the attitude of learners and their success rates.
- Implementing the survey by increasing the sampling size and number of mentors.
- Investigating the design outputs to verify their compliance with parametric design principles.
- Research on the factors behind the lack of positive attitude towards design-by-coding.

The adoption of new design methods such as parametric design will benefit from training on design-by-coding concepts and object oriented coding. It is also important to know the impact of how these new design concepts on creativity and productivity, and thus future research is required in these fields as well.

References

Baser, M. (2013). Developing Attitude Scale Toward Computer Programming, International Journal of Social Science, Volume 6 Issue 6, p. 199-215, June 2013.

Chabert, J. L. (1999). A History of Algorithms: From the Pebble to the Microchip. Springer Verlag. ISBN 978-3-540-63369-3.

Cohen, L., Manion, L., & Morrison, K. (2007). Research methods in education (6. ed). Routledge.

Coyne, R.D., Rosenman, M.A., Radford, A.D., Balachandran, M. & Gero, J.S. (1990). Knowledge–Based Design Systems, Addison Wesley Publishing Company, Massachusetts.

Fennema, E., & Sherman, J. A. (1976). Fennema-Sherman mathematics attitudesscales: Instruments designed to measure attitudes toward the learning of mathematics by females and males. Journal for Research in Mathematics Education 7(5), 324–326.

Futschek G., (2006), Algorithmic Thinking: The Key for Understanding Computer Science, *Lecture Notes in Computer Science, Vol. 4226*, 159–168.

Heersink, D., & Moskal, B. M. (2010). Measuring high school students' attitudes toward computing. In Proceedings of the 41st ACM technical symposium on Computer science education (pp 446–450). ACM.

Hongwarittorrn, N., & Krairit, D. (2010). Effects of program visualization (jeliot3) on students' performance and attitudes towards java programming. The spring 8th Int. Conf. on Computing, Com. Tech.

Hutchinson, A., Moskal, B., Dann, W., & Cooper, S. (2008). Impact of the Alice curriculum on community college students' attitudes and learning with respect to computer science. American Society for Engineering Education, Pittsburgh, PA.

Johnston, W.M., Hanna, J.R.P., & Millar, R.J. (2004). "Advances in dataflow programming languages". ACM Computing Surveys 36 (1): 1–34. Retrieved 2011-02-16.

Kernighan, B. W. (1999). The Practice of Programming, Pearson

Kolarevic, B. (2005). "Designing and Manufacturing Architecture in the Digital Age", Architectural Information Management 05, Design Process No: 3, pp.117-123.

Lawson, B. (2006). How designers think: The design process demystified. Burlington, MA., Architectural Press.

OxfordDictionaries.Com (2016). Oxford Dictionaries. Retrieved June 15, 2015 URL: http://www.oxforddictionaries.com

Rittel, H. & Webber, M. (1973). "Dilemmas in a General Theory of Planning," Policy Sciences 4, Elsevier Scientific Publishing, Amsterdam, 155-159.

Wiebe, E., Williams, L., Yang, K., & Miller, C. (2003). Computer science attitude survey. computer science, 14(25), 0.86.

Woodbury, R. (2010). Elements of Parametric Design, Routledge, NY.