A METHOD FOR COMPARING DESIGNERS' BEHAVIOR IN TWO ENVIRONMENTS: PARAMETRIC AND GEOMETRIC MODELING

Rongrong YU, Ning GU and Michael OSTWALD
University of Newcastle, Newcastle, Australia
Rongrong.Yu@newcastle.edu.au, Ning.Gu@newcastle.edu.au, Michael.Ostwald@newcastle.edu.au

Abstract. Previous research into designers’ behaviour in parametric environments suggests that parametric tools support or promote design processes in a wide variety of ways. However, despite the results of such research, there is a lack of empirical evidence directly comparing how design in parametric environments differs from design in traditional environments. Aiming to address this issue, a pilot study is presented in this paper comparing designers’ behaviour in parametric design environments (PDEs) with that in more traditional, geometric modelling environments (GMEs). In the study five designers completed two design tasks respectively in PDEs and GMEs. By employing the method of protocol analysis, different behaviours in the two design environments were identified and compared. This paper focuses on the results of testing the experimental setting and of the coding scheme used in the study. One example set of results from the pilot study is reported – wherein the designer’s behaviour exhibits some differences between the two environments – in order to provide an example of the coding used.

Keywords. Parametric design environments; geometry modelling environments; designers’ behaviour; protocol analysis; pilot study.

1. Introduction

Applications of parametric design in architecture have become increasingly prevalent in recent years. Woodbury (2010) argues that design processes in parametric design environments (PDEs) are different from those in other design environments due to the unique characteristics of PDEs. For example, by changing the parameters of an object, particular instances can be altered or created from a potentially infinite range of possibilities (Kolarevic, 2003). Previous studies suggest that parametric tools can advance design processes in a variety of ways (Iordanova et al., 2009; Lee et al., 2012; Qian et al., 2007). However, despite this past research, there is a lack of empir-
ical evidence for many of the claims about designers’ behaviour in PDEs. The central problem is that there is no basis for comparison. How, for example, can parametric design encourage higher levels of creativity (a common claim) when similarly structured data about creativity in traditional design practice is not available. Thus, the overarching question that drives the present research is, how do PDEs affect or change designers’ behaviour in comparison to that in traditional geometry modelling environments (GMEs)? To address this question, this paper described a process for comparing designers’ behaviour in PDEs and GMEs. In the pilot version of the study presented hereafter, designers were asked to complete two different design tasks with similar levels of complexity in PDEs and GMEs. This paper focuses on the method used, protocol analysis, and some preliminary results are recorded and discussed.

2. Background

2.1. DIGITAL DESIGN ENVIRONMENTS—PDES AND GMES

Past research has suggested that sketching can assist design thinking as an effective design medium (Black, 1990; Schön, 1983). In a similar way, with the increasing application of digital design tools, scholars have started to study the influence of computational design tools on design processes (Bilda and Demirkan, 2003; Fallman, 2003; Gero and Tang, 1999; Kim and Maher, 2008). Parametric design, in its computational form, is a new way of thinking about architectural design although its impact on designers’ processes hasn’t been so well explored. In order to undertake such an exploration, two, different design environments need to be identified.

- **Parametric design environments—PDEs.** Parametric design is a dynamic, rule-based process controlled by variations and parameters, in which multiple design solutions can be developed in parallel. The term “parameters” relates to factors which determine a series of variations. In architecture, parameters are usually defined as building controls or environmental factors. Currently, typical parametric design software includes Generative Component (Bentley Corporation), Digital Project (Gehry Technology) and Grasshopper (McNeel). Scripting tools include Processing based on Java language, Rhino script and Python script, based on VB language from McNeel. In the study described in the present paper Grasshopper was chosen as the parametric design environment. Grasshopper is both an advanced environment for facilitating conceptual design and it is relatively widely used in the architectural profession.

- **Geometry modelling—GMEs.** Since the emergence of digital design tools in the design industry architects have found that working with 3D geometry is preferable to working only with 2D (Bilda and Demirkan, 2003; Kan and Gero, 2009). With the development of BIM, 3D geometry modelling software has gradually replaced 2D drafting software and become the main design tool applied in the building environment. Typical 3D geometry modelling tools applied in architecture include
ArchiCAD (Graphisoft), Revit (Autodesk), Rhino (McNeel), Maya, Sketchup, etc. In the study reported in this present paper, we choose Rhino as our geometry modelling environment for the comparison. The reasons are: first, Rhino is advanced in free-form making so it will not lead to significant differences from the product produced in the Grasshopper environment; second, Grasshopper is an add-on in Rhino, which means that the two design environments are on the same platform. This property serves to minimize the variables in the experiment setting.

2.2. FBS ONTOLOGY

How then to classify the behavior of a designer in an environment? Gero’s FBS ontology (1990) has been applied in many cognitive studies (Kan and Gero, 2005; Kan and Gero, 2009). Its advocates argue that it is potentially able to capture most of the meaningful design processes (Kan and Gero, 2009) and it provides clear transitions between design instances. In the present study it is introduced as the conceptual foundation for developing the coding scheme for protocol analysis. The FBS ontology (shown in figure 1) contains three classes of variables: Function (F), Behaviour (B) and Structure (S). Function represents design intentions or purposes; behaviour represents how the structure of an artefact achieves its functions; and structure represents the components that make up an artefact and their relationships.

![Figure 1. FBS ontology (Gero, 1990).](image)

3. Research Method

3.1. PROTOCOL ANALYSIS

Protocol analysis is a method that is widely used for cognitive studies into designers’ behaviour during design processes (Cross et al., 1996; Ericsson and Simon, 1980).
This method has been applied across a variety of design environments (Kan and Gero, 2009; Kim and Maher, 2008). The general procedure for protocol analysis is that the protocol data (in this case, the video-recorded information of designers’ interaction with the design environments) is collected from the experiment and then transcribed and segmented. A customized coding scheme is then applied to categorize segments. Through statistical analysis of this data, the character of designers’ behaviour can be identified. Protocol analysis deals with a relatively small number of samples, but enables in-depth exploration of designers’ cognitive behaviour.

3.2. CODING SCHEME

This study compares designers’ cognitive behaviour in PDEs and GMEs using protocol analysis, and as categorized in accordance with the FBS ontology. The categorization of elements – the coding scheme – is described in this section. The key issue in developing an appropriate coding scheme is to capture designers’ behaviour as they are working.

The seemingly obvious difference between parametric design and traditional geometry modelling tools is concerned with the application of a rule-based algorithmic process. However, to a certain extent, architectural design has always been a rule-based algorithmic process. Renaissance and Baroque treatises describe logical, rule-based, and parametric methods for creating form and ornament. But as Ostwald (2012) notes, such methods were often peripheral to the design process in the historic examples, while they have become central or pivotal in the most recent developments. For example, in PDEs designers not only design by applying design knowledge, but also by defining and applying rules and their logical relationships using parameters. In GMEs, the rules are present, but less significant or less central.

Thus, in this study, discussion of the rule algorithm refers to the generative engine using rule algorithm. For instance, in the design knowledge level of the model, architects use design knowledge to address how the building adapts to the site, how people use the building and how to satisfy the requirements of clients. In the rule algorithm level, designers apply design knowledge through the operations of parametric design tools, such as defining the scripting rules and their logical relationships or choosing the scripting components suitable for a particular purpose.

Taking this into account, a customised coding scheme based on the FBS ontology was developed, where characteristics of parametric design are reflected (Table 1). Two levels of designers’ behaviour were coded: from the design knowledge level and the rule algorithm level. In PDEs, both levels of behaviour occur; while in GMEs, only the design knowledge level of behaviour occurs.

Designers’ behaviour includes their design intentions and design actions. Intentions inspire design actions, while actions are reflections of design intentions.
Table 1. Coding scheme.

<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Sub-category</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function (F)</td>
<td>Requirement-New</td>
<td>Design knowledge</td>
<td>F-K-Rn</td>
<td>Reading the design brief for the first time</td>
</tr>
<tr>
<td></td>
<td>Requirement-Old</td>
<td>Design knowledge</td>
<td>F-K-Ro</td>
<td>Revisiting the requirement</td>
</tr>
<tr>
<td></td>
<td>Intention-New</td>
<td>Design knowledge</td>
<td>F-K-In</td>
<td>Initial definition or interpretation of function</td>
</tr>
<tr>
<td>Structural</td>
<td>Behaviour (Bs)</td>
<td>Design knowledge</td>
<td>Bs-K-In</td>
<td>Behaviour interpreted from structures</td>
</tr>
<tr>
<td></td>
<td>Expected Behaviour (Be)</td>
<td>Rule algorithm</td>
<td>Bs-R-In</td>
<td>Behaviour from existing scripts</td>
</tr>
<tr>
<td>Structure (S)</td>
<td>Intention-New</td>
<td>Design knowledge</td>
<td>Be-K-In</td>
<td>Expected behaviour from design knowledge</td>
</tr>
<tr>
<td></td>
<td>Rule algorithm</td>
<td>Rule algorithm</td>
<td>Be-R-In</td>
<td>Expected behaviour from rule making</td>
</tr>
<tr>
<td></td>
<td>Intention-Old</td>
<td>Design knowledge</td>
<td>Be-K-Io</td>
<td>Changing the expected behaviour from design knowledge</td>
</tr>
<tr>
<td>Structure (S)</td>
<td>Intention-New</td>
<td>Design knowledge</td>
<td>S-K-In</td>
<td>Expected structure, about size, colour or shape</td>
</tr>
<tr>
<td></td>
<td>Intention-Old</td>
<td>Design knowledge</td>
<td>S-K-Io</td>
<td>Changing the structure</td>
</tr>
<tr>
<td>Geometry element making-New</td>
<td>Design knowledge</td>
<td>S-K-Gn</td>
<td>Initial making of the geometry or talking about the modeling method</td>
<td></td>
</tr>
<tr>
<td>Geometry element change-Old</td>
<td>Rule algorithm</td>
<td>S-R-Gn</td>
<td>Setting the component for the purpose of making the geometry</td>
<td></td>
</tr>
<tr>
<td>Parameter-New</td>
<td>Rule algorithm</td>
<td>S-R-Pn</td>
<td>Setting parameters</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>change-Old</td>
<td>Rule algorithm</td>
<td>S-R-Po</td>
<td>Changing or resetting parameters</td>
</tr>
<tr>
<td>Relationship-New</td>
<td>Rule algorithm</td>
<td>S-R-Rn</td>
<td>Setting the parametric relationship</td>
<td></td>
</tr>
<tr>
<td>Relationship</td>
<td>change-Old</td>
<td>Rule algorithm</td>
<td>S-R-Ro</td>
<td>Changing the parametric relationship</td>
</tr>
</tbody>
</table>
They are further divided into “new” and “old” actions. “New” means that a design instance has happened for the first time, while “Old” means a design instance has happened previously. From analysing the transition between “old” and “new” instances, the “changing” action affecting designers’ behaviour is identified.

3.3. EXPERIMENT DESIGN

In devising the experiment for the pilot study, five designers completed two different design tasks with similar levels of complexity in PDEs and GMEs. Participants were all experienced architects who have had at least two years of parametric design experience and an average seven years of experience in architectural design. The experimental environment was a computer installed with Rhino and Grasshopper. During the experiment designers’ activities and their verbalization (of these activities) were video-recorded by a screen capture programme; the recorded data forms the basis of the protocol analysis. There were two design sessions, one session using Rhino (GMEs) and the other session using Rhino and Grasshopper (PDEs). Designers were asked to finish both sessions in 40 minutes for each design session (80 minutes total). Task 1 was a formal massing concept for a community centre and task 2 was a similar concept for a shopping centre. Both buildings contained some specific functional requirements. The design sessions and tasks were randomly matched prior to the experiment. The two design tasks were created to have similar levels of complexity. Through these decisions, the research method has sought to minimize the impact of variables other than the two different design environments.

4. Results

The aim of the pilot study was to test the effectiveness of the experiment setting and the coding scheme as a precursor to undertaking the main study. Therefore, in this paper, the results focus on testing the experiment setting and the coding scheme. The results of one of the five designers from the pilot study are provided as an example coding for comparison of behaviour in the two environments.

4.1. TESTING THE EXPERIMENT SETTING

The designers each showed a clear ability to understand the design brief and operate the software. The time limit meant that now all design outcomes were resolved to the same level. Some designers stopped at the building mass or a façade design. However, they all considered the site planning as well as the building function in more detail providing a considered response to the conceptual design brief. In both PDEs and GMEs, designers started by reading the brief and inspecting the site model provided. During the design process, they also revisited the design brief. The design
brief provided details concerning functional constraints and site conditions. It was up to designers to decide how many of these conditions to consider. As usual in architectural design, different designers have their own design strategies: some designers preferred to start from functional analysis, thinking through site conditions and road and traffic information before, drawing diagrams of the site to explore these relationships. Other designers focused on the geometry modelling as the priority.

In the experiment, designers rarely sketched so that almost all their actions occurred in the computer. That ensured the design environment was within PDEs and GMEs. During their design processes, designers were asked to "think aloud". In the experiment, some designers verbalized their thinking better than others. This is one of the limitations of the "think aloud" method which has been criticized as overtly influencing participants' perceptions (Ericsson and Simon, 1993; Suwa and Tversky, 1997). In our experiment, there was a post-interview about the design session if the designer didn’t provide a sufficient level of “thinking aloud” to interpret their actions.

4.2. TESTING THE CODING SCHEME

In the first round of testing the coding scheme captured most of the cognitive design activities in both GMEs and PDEs. For example, in the sample testing, 82.2% of the segments were coded in PDEs, while in GMEs, 84.1% were coded. It was also notable that most of designers’ verbalized protocols were accompanied with non-verbalized moves. The two levels (rule algorithm and design knowledge) of design activities have therefore been able to be distinguished clearly.

During the coding process, the categories of “new” and “old” were sometimes ambiguous. In the parametric design process, rules were structured continuously and iteratively meaning it was hard to distinguish whether designers were setting a new rule or making changes to an existing rule. Meanwhile, the categories of geometry making and structure intention became confusing. Most of the time, these appear to happen together, making the coding difficult. Therefore, it was necessary to adjust the details of some subcategorys of the coding scheme.

4.3. A SAMPLE CODING

This is a sample coding from one designer’s protocol. The purpose of this is to provide an example to demonstrate some levels of comparison between the two different design environments.

- **Design issue analysis.** In the FBS ontology, the four classes of concept are called design issues. They are function (F), expected behaviour (Be), behaviour derived from structure (Bs), and structure (S). From the analysis, during the parametric design process, the designer considered the brief from both design knowledge and rule algorithm perspectives. As shown in Figure 2, this designer’s activities shifted
between these two levels. At the beginning of the design session, the designer considered more design knowledge than rule algorithm. The possible reason is that at the early design stage, the designer analysed the brief and started to form the design concept based on design knowledge.

As shown in Figures 3 and 4, in both GMEs and PDEs, the designer had the highest amount of structure related activities and the least function related activities. This matches the results of previous cognitive studies using the FBS ontology.
In the mid design session, the designer was more active in PDEs than in GMEs. In particular, the Be coding is much higher in the PDE than in the GMEs. This is because the designer set algorithm goals more frequently in the mid design session. Additionally, there are more functions in GMEs than in PDEs. That is probably because in PDEs, the designer concentrated on the rule algorithm design; while in GMEs, the designer had more time to think about the function or requirement of the brief.

- **Design process analysis.** As shown in Figures 5 and 6, there were more meaningful design processes than issues. There are eight design processes (see figure 1) presented in the FBS model. In PDEs, they are distributed consistently throughout the whole design session. There are much more reformulation processes in PDEs, especially reformulation 2, which is the transition from S to Be. This shows that this particular designer, in the PDE, tended to introduce more new variables and directions. More analysis processes occurred in PDEs (from S to Bs), possibly due to the way the designer evaluated the geometric model more frequently.

5. Conclusion

This paper presents an analysis of a sample coding of the results of a pilot study exploring the differences between designers’ behaviour in PDEs and in GMEs. The
The purpose of the present paper is not to draw conclusions about PDEs or GMEs, but rather to demonstrate that the method is viable. Thus, the pilot study reveals that: 1) the experiment setting is achievable on the practical level and serves the research aim. 2) The sub-category of the coding scheme needs to be revised in order to better capture designers’ activities. 3) From the sample coding provided, different design issues and processes can be identified to characterise parametric design. These preliminary results of behavioural patterns identified provide us with a better understanding regarding the impact of parametric design. Furthermore, the pilot study provides a sound foundation for adopting protocol analysis for the future main study.

References