

Architects' Cognitive Behaviour in Parametric Design

Rongrong Yu¹, John Gero² and Ning Gu³

^{1,3} School of Architecture and Built Environment, The University of Newcastle
Postal Address: School of Architecture and Built Environment, The University of Newcastle, University Drive, Callaghan NSW 2308, Australia
rongrong.yu@uon.edu.au, ning.gu@newcastle.edu.au

² Krasnow Institute for Advanced Study, George Mason University, and University of North Carolina at Charlotte
Postal Address:
john@johngero.com

Abstract.

This paper presents the results of a protocol study of professional architects' cognitive behaviour in a parametric design environment. A design experiment was conducted in which eight professional architects completed an architectural conceptual design task in a typical parametric design environment –Rhino and Grasshopper. Protocol analysis was then applied to analyse the cognitive behaviour of the architects. In analysing the protocol data, the FBS ontology adopted for developing the coding scheme was sub-divided into design knowledge and rule algorithm classes as the means to capture designers' cognitive behaviour. Applying the method of cumulative analysis, results of the relative cognitive effort expended on design knowledge and rule algorithm classes have been compared and are discussed in the paper.

1. INTRODUCTION

Parametric design has become increasingly prevalent in architectural design in recent years. Previous studies have argued that parametric tools can advance design processes in a variety of ways [1-5].

However, there is a general lack of empirical evidence supporting a comprehensive understanding of designers' cognitive behaviour in parametric design environments (PDEs). As a relatively new design tool, the questions of what impact parametric design has on architects and whether parametric design can adequately support the design process are therefore important to explore.

To address the above questions, a protocol study has been conducted in which eight professional architects were asked to complete an architectural design task in a typical parametric design environment using Rhino and Grasshopper. Protocol analysis [6, 7] was adopted as the research method to analyse the architects'

cognitive behaviour. From the results of cumulative analysis, the cognitive effect on the architects in the PDE are discussed and presented in this paper.

2. BACKGROUND

2.1. Research on parametric design

Parametric design is a dynamic, rule-based process controlled by variations and parameters, in which multiple design solutions can be developed in parallel. According to Woodbury [8], it supports the creation, management and organisation of complex digital design models. Using parametric design tools, designers can make rules according to the performance requirements of a design. A parameter is a value or measurement of a variable that can be altered or changed. In architecture, parameters are usually defined related to building or environmental factors. A parametric design model will have some rules embedded in the system and when one parameter changes, other parameters will adapt automatically [9]. By controlling parameters, particular design instances can be created from a potentially infinite range of possibilities [10]. In the architectural design industry, parametric design tools are utilised mainly for complex building form generation, multiple design solution optimisation, as well as structural and sustainability control.

Parametric design is, in comparison with conventional design, quite different – not only because it offers a new design tool but also a new way of thinking.

Currently, typical parametric design software includes Generative Component, Digital Project, Grasshopper. Scripting tools include Processing based on the Java programming language, Rhino script and Python script based on the VB programming language, and DesignScript. In this study, Grasshopper was chosen as the parametric design environment. Grasshopper is an advanced computer environment for facilitating conceptual design and is a typical parametric environment widely used in the architectural profession.

Previous studies on designers' behaviours in PDEs suggest that parametric tools can advance design in a variety of ways. For instance, there is evidence that the generation of ideas is positively influenced in PDEs. Particularly, in Iordanova et al.'s [1] experiment on generative methods, ideas were shown to be generated rapidly while they also emerge simultaneously as variations. Moreover, Schnabel [2] showed that the PDE is beneficial for generating unpredicted events and can be responsible for accommodating changes. However, researchers have typically studied design behaviour in PDEs mostly by observing

students' interactions in design studios or workshops. Arguably, this approach cannot provide an in-depth understanding of designers' behaviours. This empirical gap will be addressed in the present study by adopting the method of protocol analysis. In 2014, Lee et al. [11] presented a pilot study using protocol analysis to evaluate creativity in the PDE. Results of their study identified some conditions that can potentially enhance creativity in the PDE. Using the same method, Chien and Yeh [5] explored "unexpected outcomes" in the PDE. These preliminary studies varied and were limited in terms of the participant number but provided important precedents for the present study.

2.2. Protocol studies using the FBS ontology

Protocol analysis is a method for turning qualitative verbal and gestural utterances into quantitative data [6, 7]. Protocol studies have been used extensively in design research to develop an understanding of design cognition [12-14]. According to Akin [15], a protocol is the record of behaviours of designers using sketches, notes, videos or audio. After collecting the protocol data, a coding scheme is applied to categorise the collected data, enabling a detailed study of the design process in the chosen design environments. As Gero and Tang [16] stated, protocol analysis has become the prevailing experimental technique for exploring the understanding of design.

As one of the main design ontologies, Gero's FBS model [17] has been applied in many cognitive studies [18-20]. Researchers argue that the model is potentially capable of capturing most of the meaningful design processes [19] and the transitions between design issues are clearly classified into eight design processes. The FBS ontology (Figure 1) contains three types of variables: Function (F), Behaviour (B) and Structure (S). Function (F) represents the design intentions or purposes; behaviour (B) represents the object derived (Bs) or expected from the structure (Be); and structure (S) represents the components that make up an artefact and their relationships. The model is augmented by two external design issues that do not require an extension of the ontology itself as they can be represented as either F, B or S: requirements (R) and descriptions (D). The first of these represents requirements from outside design and the second, descriptions, mean the documentation of the design. Figure 1 shows the FBS ontology indicating the eight design processes—formulation, analysis, evaluation, synthesis, and reformulation I, II, III. Formulation defines the process that produces a function or sets up expected goals from the existing requirement, while synthesis generates a structure as a candidate solution. Analysis produces a behaviour from the existing

structure and evaluation compares Bs and Be to determine the effectiveness of the candidate solution. Reformulation is the process from the structure back to itself, behaviour or function, which is a reconstruction process. Among the eight design processes, the three types of reformulation processes are suggested to be the dominant processes that potentially capture creative aspects of designing by introducing new variables or new directions [13]. By calculating the transitions between design issues, various analyses can be conducted. In this study, the FBS ontology will be introduced as the basis model for developing the coding scheme.

The reason we use the FBS ontology for developing the coding scheme is that: firstly, the FBS ontology has been found useful in a wide range of domains including architecture, civil engineering, mechanical engineering, software engineering, cognitive psychology, manufacturing, management and creativity research. Secondly, it has been applied and tested in a number of cognitive design studies [20, 21] where it has been demonstrated as potentially capturing most of the meaningful design processes and recording clear transitions between design instances. The behaviour of designers, using the FBS ontology as the basis, can be measured from empirically derived data using protocol analysis. Thirdly, Kan and Gero [22] (2012) applied the FBS ontology to a study of software designers' behaviour. Their study suggests that the method is effective for encoding programming or rule-based activities across different design disciplines. Given that PDEs enable scripting and programming activities, similarly the FBS coding scheme will be able to encode both geometric modelling and rule-based algorithmic activities effectively.

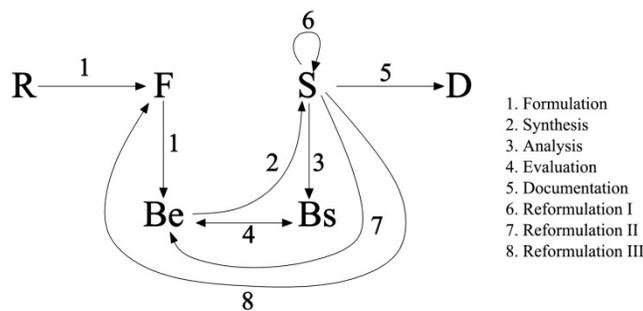


Figure 1: The FBS ontology [After 17]

3. RESEARCH DESIGN

3.1. Experiment setup

The study reported in this paper is focused on eight designers, who are all professional architects recruited with an average of eight years of experience, and with no less than two years of experience in parametric design. Half of the designers are lecturers from universities with previous experience in the practice. Others are architects currently work in architectural design companies. In the experiment, each designer was required to complete the same given architectural design task using the chosen parametric modeller. During the experiment, both designers' activities and their verbalisation were video-recorded by a screen capture program and the recorded data subsequently used for protocol analysis. The design environment is Rhino and Grasshopper, a typical PDE. Designers were given 40 minutes for the design session, although most needed slightly more time to complete the design. The design task was a conceptual design for a commercial building containing specific functions and located on a pre-modelled site that was provided to each designer. Since this study focuses on exploring designers' behaviour at the conceptual design stage, participants were required to only consider concept generation, simple site planning and general functional zoning. No detailed plan layout was required. The tasks were both open and general enough to provide designers with the freedom to enable various possible strategies to be applied during the parametric design process. During the experiment, designers were not allowed to sketch manually so that almost all their actions occurred on the computer to ensure that the design environment was purely within the PDE. All of these controls were put in place to ensure that variables which could potentially bias the study were minimized.

3.2. A coding scheme for two classes of parametric design activities

The ways in which parametric design is used by architects are not well understood which is why Sanguinetti and Kraus argue that parametric design “requires a deeper understanding of how it can support our intentions as architects” [23, p. 47]. Compared to traditional design environments, in a PDE designers not only design by applying design knowledge, but they also define rules and their logical relationships, using parameters. When an architect models a building form using parameters they must assess design variations and data flows, adjust the values of parameters, and create or revise rules accordingly. At this time they are not only thinking about the actual building design, but also the rule design in order to achieve the building design. It is through the control of logical relationships between forms and functions as explicitly defined in the rules that the possibilities for design solutions are heightened [24, 25]. Thus, in a

typical parametric design process, there are two classes of cognitive design activities: the design knowledge and the rule algorithm. In the design knowledge class, architects make use of their design knowledge directly for addressing , for example, how to satisfy various functional requirements, how to adapt a building to the site, and how to shape the way people use the building. In the rule algorithm class, they apply design knowledge through the operations of the parametric design tools, including defining the rules and their logical relationships, choosing the parameters suitable for a particular purpose and relating external data with the defined rules. In order to capture the design processes involved in PDEs, each main variables from the original FBS ontology (R, F, Be, Bs, and S) are further decomposed into the two classes of design activities: the design knowledge class, denoted by the superscript K, and the rule algorithm class, denoted by the superscript R (Figure 2). For instance, when coding a segment in which a designer sets up an algorithm goal that segment will be coded as F^R .

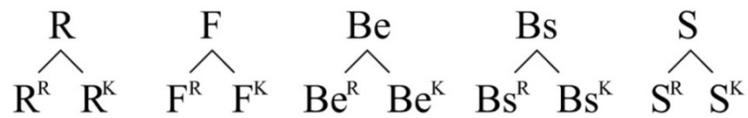


Figure 2: Applying the FBS ontology in the PDE.

3.3. Analysis method – cumulative occurrence of design issues

In order to describe the effort that designers expended on each design issue across the design sessions, cumulative occurrence analysis is introduced to measure the aggregation process of the design issues. The cumulative occurrence of a design issue is shown by the number of design issues of one class that have occurred so far in a session at any particular segment.

We will use measurements based on the cumulative occurrence of the design issues; these measurement methods are adopted from Gero and Kannengiesser’s study on the cumulative analysis of multi-disciplinary designers’ behaviour [26]:

- First occurrence at start: if the design issue occurred at the start of the design session or later. This measures when a particular design issue starts.
- Continuity: if the design issue occurs throughout the design session or stops at a certain point.
- Shape of the graph: if the graph is linear or non-linear. Linearity is measured by the value of R^2 .
- R^2 : if $R^2 \geq 0.95$, then the graph is linear.

- Slope: the slope is a measure of the speed of the design issue generation. The larger the value of the slope, the faster the design issue is generated.

These measurements reveal the details of designers' activities across the whole design session. Using cumulative graphs, we can develop an understanding of the range and scale of the data. The number of segments in each of the eight protocols will be normalised to 100 segments.

4. CUMULATIVE OCCURRENCE RESULTS

4.1. General results

In order to increase the robustness of the protocol coding process, two rounds of segmentation and coding were conducted with an interval of two weeks between them. An arbitration session was subsequently carried out to produce the final protocol from the combination of the two coding rounds. The percentage of agreement between the two rounds was 83.4% [SD=5.7%], and between the individual rounds and the final arbitrated results was 91.5% [SD=3.1%]. These percentages are indicative of the methodological reliability of the coding process and results. From analysing the eight protocols, the average overall numbers of segments was 244 [SD=29.7]. The mean time spent on the design session was 48.4 minutes [SD=7.4] and over 92.2% of all segments were coded using the FBS model. Non-coded segments include those concerned with communication, software management, etc. The quantitative analysis of FBS design issue occurrences is shown in Table 1. The occurrences of design issues were normalized by turning them into percentages of the total number of design issues in each protocol. The results indicate that designers spent most of their cognitive effort on the structure-related design issues which consist of structure S^K (21.56%), S^R (20.49%), and behaviour from structure Bs^K (23.55%). Although designers show individual differences in their design strategies and habits, by exploring the repeated patterns across their behaviours, characteristics of parametric design can be identified.

Table 1. Design issue distribution

	R^K (%)	F^K (%)	Be^K (%)	Be^R (%)	Bs^K (%)	Bs^R (%)	S^K (%)	S^R (%)
Mean	1.52	5.09	11.08	11.59	23.55	5.18	21.56	20.49
SD	0.72	2.81	7.96	6.12	5.09	2.80	9.37	10.18

4.2. Cumulative occurrence of overall design issues

The cumulative occurrence of the two classes of activities in terms of overall design issues in the PDE is shown in Figure 3. The horizontal axis is the normalised segment number, while the vertical axis represents the average cumulative occurrence of eight designers. Table 2 shows the measurements of the data represented in Figure 3. We set the threshold of agreement of 80% of all the protocols to determine the first occurrence at start, continuity, and linearity of the design issue.

The cumulative analysis result from Figure 3 and Table 2 suggest obvious differences between the design knowledge (K) and rule algorithm (R) classes. Design issues for the design knowledge class of all protocols start from the beginning of the design sessions, while those of the rule algorithm class do not. The graph shapes of all protocols of the design knowledge class are linear, while those of the rule algorithm class are not. In both classes, the design issues occur continuously. The two classes of design thinking, K and R, continue throughout the design sessions. The generation speed of rule algorithm occurrence is slower than that of design knowledge. The results suggest that designers consider design knowledge at the beginning of the design session while they think about rule algorithm later. Then they use both design knowledge and rule algorithm throughout the rest of design session.

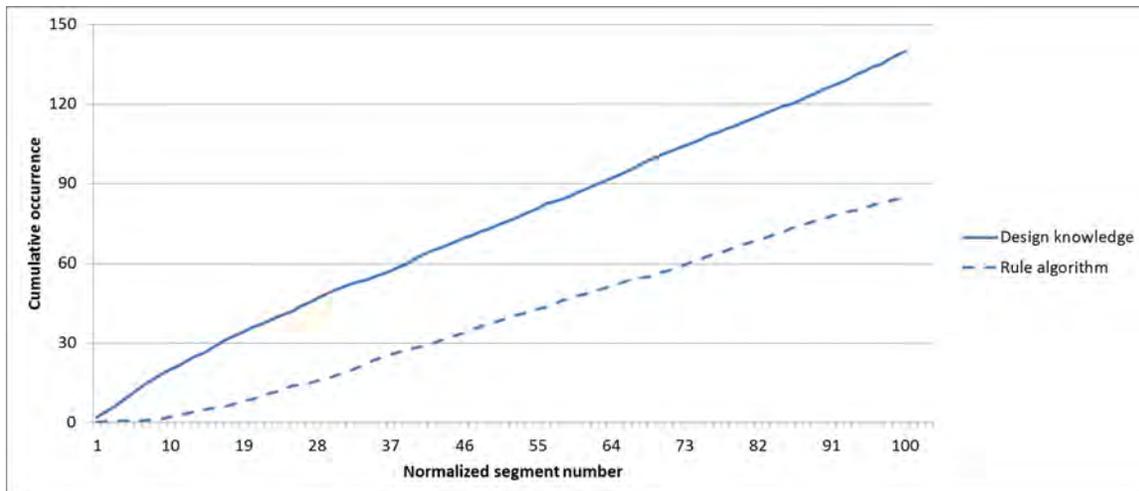


Figure 3: Cumulative occurrence of overall design issues

Table 2. Measurements and observations of cumulative occurrence of overall design issues of the two classes (K and R) expressed as averages of the eight designers

	Slope		R ²		First occurrence at start		Continuity		Shape (linear)	
	K	R	K	R	K	R	K	R	K	R
Mean	0.590	0.408	0.977	0.922	100%	0%	100%	88%	88%	50%
SD	0.152	0.150	0.020	0.078						

In the rest of this section, we look further into cumulative occurrence of design issues Be, Bs and S of the two classes of design activities. The reason we exclude F and R in the analysis is that they only occur at the design knowledge level.

4.3. Cumulative occurrence of design issue Be

The graph of cumulative occurrence of the two classes of activities of the design issue Be (expected behaviour) is shown in Figure 4. Table 3 shows the measurements of the data represented in Figure 4. The cumulative analysis result as shown in Figure 4 and Table 3 suggests: at the design knowledge class, Be starts at the beginning of the design session for all protocols, while at the rule algorithm class it starts later. The Be issues occur discontinuously at both design knowledge and rule algorithm classes, with a nonlinear shape for the cumulative curve. Analysis suggests that designers set up design knowledge related expected behaviours at the beginning of the design session, and the rule algorithm related expected behaviours are set up later.

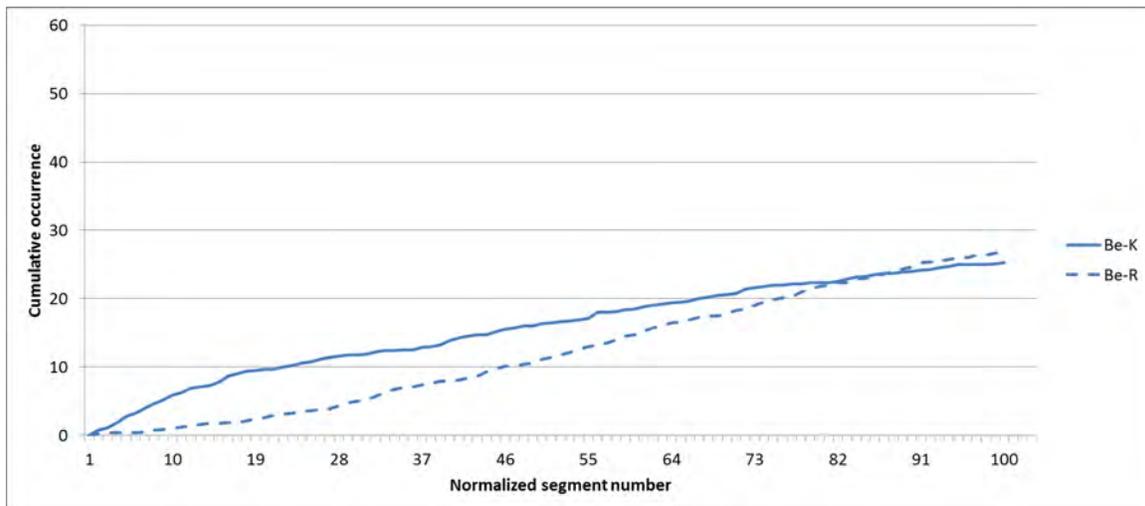


Figure 4: Cumulative occurrence of design issue Be

Table 3. Measurement and observations of cumulative occurrence of Be of the two classes (K and R) expressed as averages of the eight designers

	Slope		R ²		First occurrence at start		Continuity		Shape	
	K	R	K	R	K	R	K	R	K	R
Mean	0.102	0.131	0.908	0.896	100%	75%	25%	63%	0%	38%
SD	0.078	0.073	0.064	0.117						

4.4. Cumulative occurrence of design issue Bs

The cumulative occurrence of the two classes of activities of design issue Bs (behaviour from structure) is shown in Figure 5. Table 4 shows the measurements of the data represented in Figure 5. Figure 5 and Table 4 show that: firstly, the design issue Bs of all protocols do not start from the beginning of the design sessions; secondly, the Bs issue at the design knowledge class occur continuously while at the rule algorithm class it is discontinuous. Furthermore, the shape of the cumulative graph of Bs at the design knowledge level is linear, while at the rule algorithm level is nonlinear. During the design session, the examination of the geometry occurs continuously while the examination of the rules occurs occasionally. The occurrence of examining the rule is slower than examining the geometry.

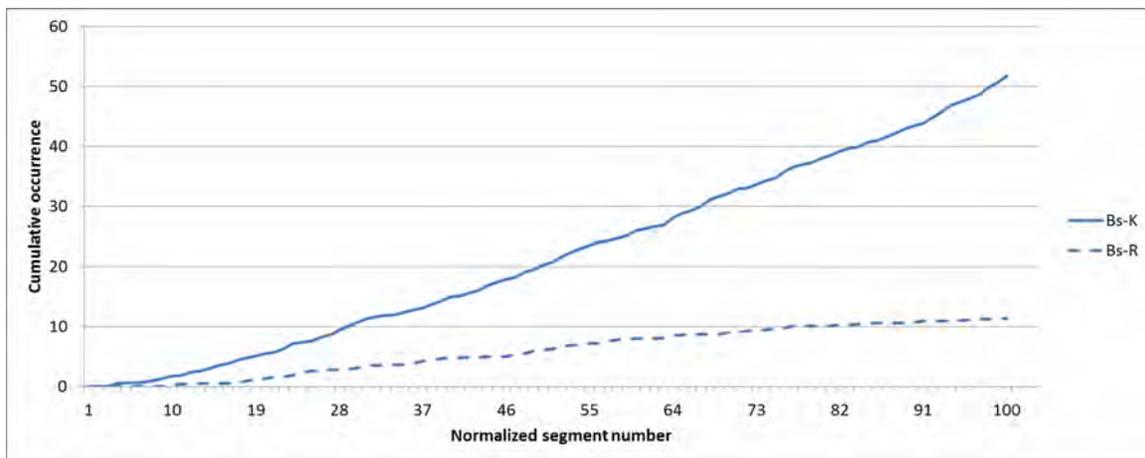


Figure 5: Cumulative occurrence of design issue Bs

Table 4. Measurement and observations of cumulative occurrence of Bs issue of the two classes (K and R) expressed as averages of the eight designers

	Slope		R ²		First occurrence at start		Continuity		Shape	
	K	R	K	R	K	R	K	R	K	R
Mean	0.237	0.061	0.971	0.851	38%	0%	100%	25%	100%	25%
SD	0.043	0.033	0.014	0.110						

4.5. Cumulative occurrence of design issue S

The cumulative occurrence of the two classes of design issue S (structure) in the PDE is shown in Figure 6.

Table 5 shows the measurements of the data represented in Figure 6. Figure 6 and Table 5 reveal some characteristics of design issue S at the two classes of activities based on the cumulative analysis result: firstly, the S issue at the design knowledge class of all protocols starts at the beginning of the design session, while at the rule algorithm class the S issue starts later; secondly, design issue S occurs

continuously at both design knowledge and rule algorithm classes. The cumulative graphs of the S issue in both classes are non-linear

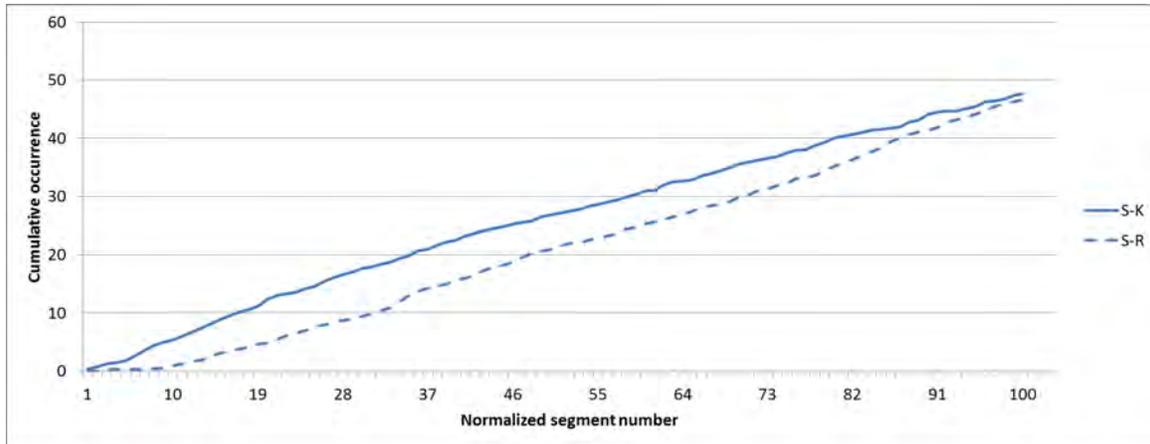


Figure 6: Cumulative occurrence of design issue S

Table 5. Measurement and observations of cumulative occurrence of S issues of the two classes (K and R) expressed as averages of the eight designers

	Slope		R ²		First occurrence at start		Continuity		Shape	
	K	R	K	R	K	R	K	R	K	R
Mean	0.212	0.218	0.944	0.932	88%	13%	100%	88%	63%	63%
SD	0.117	0.108	0.041	0.072						

5. RELATIVE EFFORT OF THE TWO CLASSES

Based on the cumulative occurrence analysis above, we normalise the absolute numbers into percentages. Since the length of each design session varies the number of segments in a session has been normalised to 100 segments, which makes each design session the same length (of 100 normalised segments). In order to be able to compare the relative cognitive effort, the cognitive activations associated with design knowledge and with rule algorithm are separated and aggregated across all eight protocols.

At each normalised segment the relative percentages of cognitive effort expended on the two classes of design activities – design knowledge and rule algorithm – are calculated and plotted. The resulting graphs provide a qualitative overview of the locus of cognitive effort distribution between design knowledge and rule algorithm, in addition to the quantitative values used to produce the graphs.

5.1. Overall relative effort on the two classes

The overall relative effort on the two classes of designers' cognitive activities – design knowledge and rule algorithm is illustrated in Figure 7. The vertical axis represents the average value of relative effort of the eight protocols. The horizontal axis is the normalised segment numbers.

From the overall distribution of cognitive effort shown in Figure7, we can see that initially the cognitive effort on design knowledge dominates that expended on rule algorithm. However, as the design session proceeds, the cognitive effort on design knowledge drops from 100% to approximately 60% of the total in a shape that looks like a decay curve. In parallel, as the design session proceeds, the cognitive effort expended on rule algorithm increases from 0% to approximately 40% of the total in a shape that looks like an excitation curve. Therefore, we can infer that in the parametric design process, designers still expend most effort on design knowledge; parametric scripting is mainly used to support their intention of generating models. Designers started with considering design knowledge related issues, such as building functions; as the design proceeded, they gradually spent more of their cognitive effort on parametric scripting. In the following sections, we articulate these results with further detail about the cognitive behaviour related to Be, Bs and S and draw implications from them.

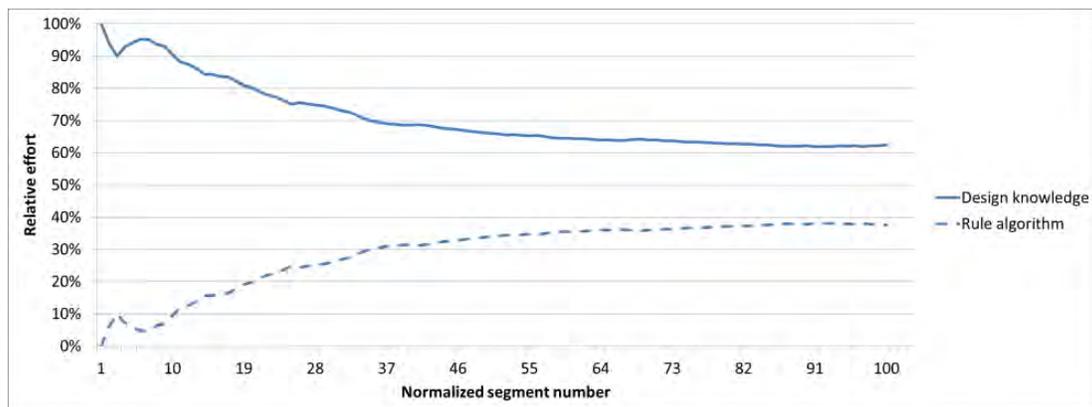


Figure7: Overall relative cognitive effort expended on the two classes

5.2. Relative effort on Be

Expected behaviour (Be) means that “designers use theory or experience to speculate what effect could fulfil a purpose before a specific structure is proposed” [27, p 36-37]. Applying it in rule algorithm related activities, Be^R often can refer to that designers set up algorithm goals or think about the way to achieve those goals.

The relative cognitive effort of Be expended on the two classes during the parametric design process is presented in Figure 8. The relative effort expended on the design knowledge class (Be^K) is higher at the beginning and then decreases across the design session; while that on the rule algorithm class (Be^R) rises toward the end of design session. At the end of the design session, designers' cognitive effort expended on the rule algorithm class exceeds that on the design knowledge class. From this we can infer that later in the design session, designers tend to focus more on setting rule algorithm goals and exploring ways to achieve those goals.

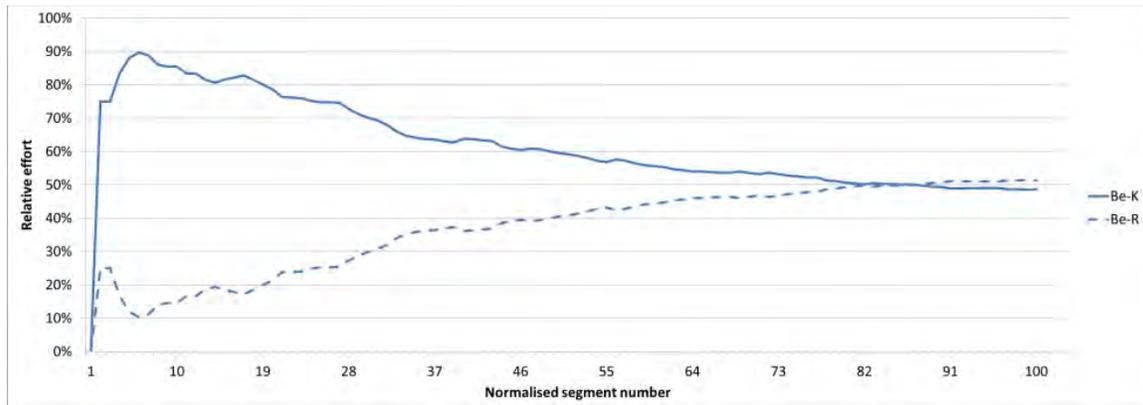


Figure 8: Relative cognitive effort of Be expended on the two classes

5.3. Relative effort on Bs

Structure behaviour (Bs) refers to the behaviour derived from the structure: for the design knowledge class, Bs^K represents the evaluation of the existing geometry/structure; while for the rule algorithm class, Bs^R means evaluating the structure of the rule algorithm.

The relative cognitive effort of Bs expended on the two classes during the parametric design process is presented in Figure 9. Designers expended noticeably more cognitive effort on the design knowledge class than on the rule algorithm class during the whole design session. Design knowledge related activities decrease from 100% to approximately 70% during the first third of the design session and then remain unchanged at between 70%-80%; the rule algorithm related activities increase from 0% to 30% during the first third of the design session and then remain unchanged at between 20%-30%. From the results in Figure 9, we can also infer that in terms of Bs, rule algorithm related activities (Bs^R) do not commence at the beginning of the design session. One of the possible reasons is that designers only examine rule settings

after their rule concepts achieve a certain degree of maturity. Also, designers examine the geometry far more than checking the rule setting.

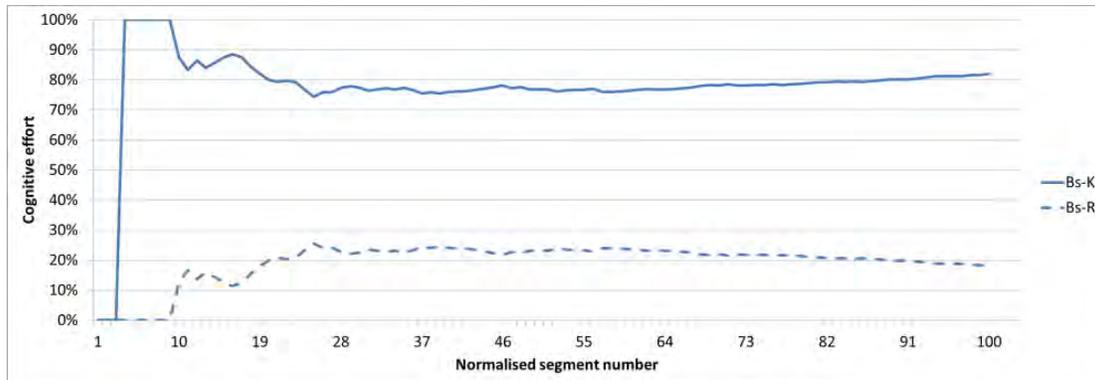


Figure 9: Relative cognitive effort of Bs expended on the two classes

5.4. Relative effort on S

Structure (S) variables describe ‘the components of the object and their relationships, i.e., “what it is”.’ [28, p 374]. In the design knowledge class, structure (S^K) refers to the elements or relationships of the geometries; while in the rule algorithm class, structure (S^R) is defined as the structure of the rule algorithm – the components of rules and their relationships for parameterisation.

The relative cognitive effort of S expended on the two classes during parametric design process is shown in Figure 10. From the results in Figure 10, we can infer that design knowledge related activities dominate the design process during the early and mid-stages of the design session and then decrease and converge to a level similar to that of rule algorithm-related activities. The cognitive effort expended on S^K decreases from 100% at the beginning of the sessions to approximately 50%; while those on S^R increase from 0% to approximately 50%.

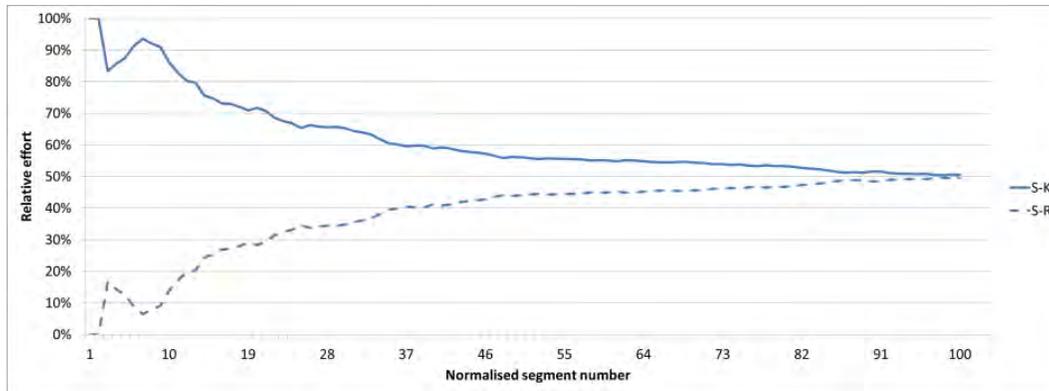


Figure 10: Relative cognitive effort of S expended on the two classes

6. CONCLUSION

This paper reveals architects' cognitive behaviour in parametric design with a protocol study on eight professional architects. Through protocol analysis and based on the results of cumulative analysis of the design issues, the paper suggests that: firstly, the division of the design knowledge and rule algorithm classes is useful in understanding the designers' behaviour in the PDE; secondly, some characteristics of the two design classes can be explored in terms of the first occurrence at start, continuity, and linearity; and thirdly, the relative cognitive effort analysis indicates that the design knowledge related activities dominate the parametric design process for all cognitive design issues. From the analysis of the relative effort expended on the two classes of design activities, we can identify the impact of rule algorithm in the parametric design process. Initially the cognitive effort on design knowledge dominates that on rule algorithm. However, as the design proceeds, the cognitive effort on design knowledge drops from 100% to approximately 60% of the total. In parallel, as the design proceeds, the cognitive effort on rule algorithm increases from 0% to approximately 40% of the total. Therefore, we can infer that in the parametric design process, designers still expend most effort on design knowledge; parametric scripting is mainly used to support their intention of generating models. Designers start with considering design knowledge related issues, such as the client's requirement and building functions; when the design proceeds, they gradually spent more cognitive effort on parametric scripting.

These results have been produced using the method of protocol analysis. The common limitation of protocol analysis is that only a relatively small sample size can be accommodated in the method.

Nonetheless, it can still provide a very detailed data set and enable an in-depth analysis of the collected protocol data. Another limitation is that the study is based on simulated experiments, which would not be exactly the same as those undertaken in an actual design practice. However, in order to explore specific issues in the study and to produce reliable data, we have to constrain the experiments. By studying designers' cognitive behaviour in the artificially simulated design experiment, it is possible to identify and isolate certain detailed design activities and design processes by only focussing on those specific issues.

The implications of the analysis results, if they are found to be generalizable, is that practicing architects with experience in using parametric design tools make use of those tools very early in a design session and make increasing use of them as the design session proceeds. This implies that the designer is substituting rule algorithm for design knowledge. This opens up ways of encoding design patterns that can form the basis of reusable rules that allow a designer to develop a style of designing and through the rule parameterisation a style of designs. Each design generated through the use of these patterns but parameterised individually is unique and responds to each unique program but forms part of an overall style associated with an individual designer or design team. The findings of the research also have implications for architectural education in the digital design age. Architectural education should include and address how design knowledge is captured in a rule algorithm and in parametric design in general rather than focusing on the technical skills of rule algorithm use.

From the designer's perspective, parametric design tools can provide architects opportunities for designing through both design knowledge and rule algorithm, which opens up many possibilities: complex forms are able to be generated and managed more efficiently; parameters and external data can be embedded and linked to a design to enable more rational and better solutions; design variations can be developed in parallel and changes at different stages of the design can be easily made and traced. At the same time, new challenges emerge: first and foremost the role of architects is changing, such they need to be both architects and programmers/scripters. Using parametric tools, a designer's programming/scripting skill has an impact on design. Through qualitative observation of our experiment, designers appear to use the programming/scripting method or existing scripts they are familiar with. This can be both efficient and constraining. The ways designers use parametric tools is therefore critical for the success of parametric

design. Some characteristics of architects' cognitive behaviours in parametric design have been identified in this study, which can be directed to develop guidelines for parametric designers.

Acknowledgements

This research has been supported in part by the National Science Foundation grant CMMI-1161715. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of National Science Foundation.

References

1. Iordanova, I., et al. Parametric methods of exploration and creativity during architectural design: A Case study in the design studio. eds, 13th International Conference on Computer Aided Architectural Design Futures. Presses de l'Univ. de MontréalMontréal.2009, 423-439.
2. Schnabel, M.A. Parametric Designing in Architecture. in A. Dong, V. Moere, and J.S. Gero, A. Dong, V. Moere, and J.S. Gero eds, Proceedings of the Twelfth International Conference on Computer Aided Architectural Design Futures. SpringerSydney.2007, 237-250.
3. Qian, C.Z., V.Y. Chen, and R.F. Woodbury. Participant Observation Can Discover Design Patterns in Parametric Modeling. eds, ACADIA2007. Halifax.2007, 230-241.
4. Abdelmohsen, S. and E.Y.-L. Do. Analyzing the significance of problem solving expertise and computational tool proficiency in design ideation. in T. Tidaf and T. Dorta, T. Tidaf and T. Dorta eds, CaadFutures 2009. Presses de l'Univ. de MontréalMontréal.2009, 273-287.
5. Chien, S.-F. and Y.-T. Yeh. On Creativity and Parametric Design---A preliminary study of designer's behaviour when employing parametric design tools. eds, of eCAADe 2012. eCAADe in cooperation with MediatectureCzech Republic.2012.

6. Ericsson, K.A. and H.A. Simon, Protocol analysis : verbal reports as data. 1993, Mass.: MIT Press. viii, 426 p.
7. Gero, J.S. and T. Mc Neill, An approach to the analysis of design protocols. Design Studies. 19(1). 21-61.
8. Woodbury, R., Elements of Parametric Design. 2010, New York: Routledge.
9. Ostwald, M., Systems and Enablers: Modeling the impact of contemporary computational methods and technologies on the design process, in N. Gu and X. Wang, Editor^Editors eds., Book Title], Publisher]: Place Published, 2012. 1-17.
10. Kolarevic, B., Architecture in the digital age : design and manufacturing. 2003, New York, NY: Spon Press.
11. Lee, J.H., et al. Evaluating Creativity in Parametric Design Processes and Products: A Pilot Study. in J. Gero, J. GeroJ. Geros eds, Proceedings of international conference of Design Computing and Cognition, DCC'12. SpringerTexas.2014.
12. Atman, C.J., et al., A comparison of freshman and senior engineering design processes. Design Studies. 20(2). 131-152.
13. Kan, J.W.T. and J.S. Gero, Acquiring information from linkography in protocol studies of designing. Design Studies. 29(4). 315-337.
14. Suwa, M. and B. Tversky, What do architects and students perceive in their design sketches? A protocol analysis. Design Studies. 18(4). 385-403.
15. Akin, O., Psychology of Architectural Design 1986: Pion, London.
16. Gero, J. and H.-H. Tang, The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process. Design Studies. 22(3). 283-295.

17. Gero, J.S., Design prototypes: a knowledge representation schema for design. *AI Magazine*. 11(4). 26–36.
18. Gero, J. and H.-H. Tang. Concurrent and Retrospective Protocols and Computer-Aided Architectural Design. eds, CAADRIA1999. Shanghai.1999, 403-410.
19. Kan, J.W.T. and J.S. Gero, Using the FBS ontology to capture semantic design information in design protocol studies, in J. McDonnell and P. Lloyd, Editor^Editors eds., Book Title], Publisher]: Place Published, 2009. 213-229.
20. Kan, J.W.T. and J.S. Gero. Can Entropy Indicate the Richness of Idea Generation in Team Designing? eds, CAADRIA 2005. New Delhi, India.2005, 451-457.
21. Kan, J.W.T. and J.S. Gero. The Effect of Computer Mediation on Collaborative Designing. eds, Proceedings of the 14th International Conference on Computer Aided Architectural Design Research in Asia (CAADRIA) Yunlin, Taiwan.2009, 411-419.
22. Kan, J.W.T. and J.S. Gero, Studing software design cognition, in M. Petre and A.V.d. Hoek, Editor^Editors eds., Book Title], Publisher]: Place Published, 2012.
23. Sanguinetti, P. and C. Kraus. Thinking in Parametric Phenomenology eds, Parametricism (SPC) ACADIA Regional 2011 Conference. University of Nebraska-LincolnLincoln.2011, 39-48.
24. Hernandez, C.R.B., Thinking parametric design: introducing parametric Gaudi. *Design Studies*. 27(3). 309-324.
25. Karle, D. and B. Kelly. Parametric Thinking. eds, Proceedings of ACADIA Regional 2011 Conference.2011, 109-113.
26. Gero, J.S. and U. Kannengiesser. Commonalities across Designing: Empirical Results. eds, Proceeding of International conference of Design Computing and Cognition (DCC'12). Berlin: SpringerCollege Station.2014, 285-302.

27. Jiang, H., Understanding senior design students' product conceptual design activities--a comparison between industrial and engineering design students. 2012, National University of Singapore.
28. Gero, J. and U. Kannengiesser, The situated function-behaviour-structure framework. *Design Studies*. 25(4). 373-391.