

# An empirical basis for the use of design patterns by architects in parametric design

International Journal of  
Architectural Computing  
2016, Vol. 14(3) 289–302  
© The Author(s) 2016  
Reprints and permissions:  
sagepub.co.uk/journalsPermissions.nav  
DOI: 10.1177/1478077116663351  
jac.sagepub.com



Rongrong Yu<sup>1</sup> and John S Gero<sup>2</sup>

## Abstract

This article presents the results from exploring the impact of using a parametric design tool on designers' behavior in terms of using design patterns in the early conceptual development stage of designing. It is based on an empirical cognitive study in which eight architectural designers were asked to complete two architectural design tasks with similar complexity, respectively, in a parametric design environment and a geometric modeling environment. The protocol analysis method was employed to study the designers' behavior. In order to explore the development of design patterns in the empirical data, Markov model analysis is utilized. Through Markov models analysis of the parametric design environment and geometric modeling environment results, it was found that there are some significantly different design patterns being used when designing in a parametric design environment compared to designing in a geometric modeling environment. The article articulates these differences and draws conclusions from these results.

## Keywords

Design patterns, parametric design, protocol analysis

## Introduction

Parametric design has become increasingly prevalent in architectural design. However, there is a lack of empirical evidence supporting an understanding of designers' behavior in parametric design environments (PDEs). In a PDE, designers often adopt existing design patterns based on their experience in using their design knowledge and in using parametric tools. This phenomenon has not been adequately studied and evaluated in architecture design. To improve our understanding of the possible use of design patterns while designing, the results of a cognitive study in which designers were asked to complete two architectural design tasks with similar complexity, respectively, in a PDE and a geometric modeling environment (GME) are presented. Protocol analysis<sup>1,2</sup> was employed to study the designers' behavior. Utilizing a technique that finds patterns in sequential data (Markov model analysis), the results of how designers use

---

<sup>1</sup>Northeast Forestry University, Harbin, China

<sup>2</sup>Computer Science and Architecture, University of North Carolina at Charlotte, Charlotte, NC, USA

### Corresponding author:

Rongrong Yu, Northeast Forestry University, 26 Hexing St, Harbin, 150000, China.

Email: rongrong.yu@uon.edu.au

design patterns in the PDE are presented and discussed. The remainder of this article introduces the background to the study briefly covering parametric design and design ontologies before proceeding to describe the research design and results. This is followed by a section that presents an analysis of the results in terms of empirically grounded design patterns. The article ends with a brief discussion of the implications of the results of this study.

## Background

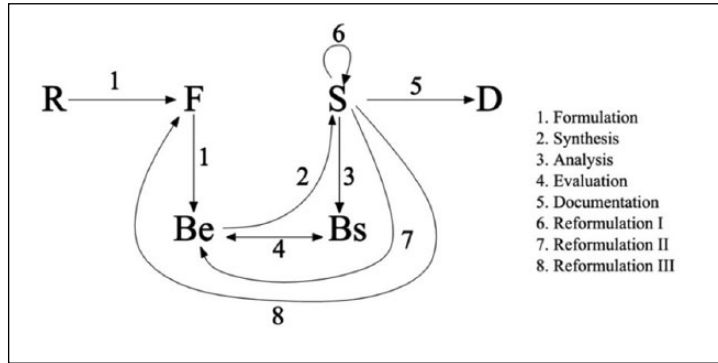
### *Parametric design*

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,<sup>3</sup> it supports the creation, management, and organization of complex digital design models. By changing the parameters of an object, particular instances can be altered or created from a potentially infinite range of possibilities.<sup>4</sup> The term “parameters” means factors which determine a series of variations. In architecture, parameters are usually defined as building variables or environmental factors. In the architectural design profession, parametric design tools are utilized mainly on complex building form generation, multiple design solution optimization, as well as structural and sustainability control. Currently, typical parametric design software includes Generative Component from Bentley Corporation, Digital Project from Gehry Technologies, and Grasshopper from McNeel. Scripting tools include Processing based on the Java language, Rhino script, and Python script, based on Visual Basic (VB) language from McNeel. In this study, Grasshopper was chosen as the PDE. Grasshopper is both an advanced environment for facilitating conceptual design and is in relatively widespread use in the architectural profession.

Previous studies on designers’ behaviors in PDEs suggest that parametric tools advance design processes in a variety of ways. For instance, there is evidence that the generation of ideas is positively influenced in PDEs. In Iordanova et al.’s<sup>5</sup> experiment on generative methods, ideas were shown to be generated rapidly while they also emerge simultaneously as variations. Schnabel<sup>6</sup> showed that PDE is beneficial for generating unpredicted events and can be responsible for accommodating changes. However, researchers have typically studied design behavior in PDE mostly by observing students interactions in PDEs in design studios or workshops. Arguably, this approach cannot provide an in-depth understanding of professional designers’ behaviors. This empirical gap is addressed in this study by adopting the method of protocol analysis to produce empirical results. In 2014, Lee et al.<sup>7</sup> presented a pilot study using protocol analysis to evaluate creativity in PDEs. Results from their study identified some conditions that potentially enhance creativity in the PDE. Using the same method, Chien and Yeh<sup>8</sup> explored “unexpected outcomes” in a PDE. However, without a basis for comparison, it is difficult to suggest how parametric tools enhance or hinder creativity in comparison with traditional design methods. In order to allow results to be compared, they need to be grounded in the same representation. In empirical studies, it is common to develop a representation for the results that is uniquely shaped to the experiment. The disadvantage of that approach is that it does not allow comparisons with other studies. In order to overcome that disadvantage, this study employs a widely used design ontology as the basis of its representation.

### *Function–Behavior–Structure ontology*

As one of the main design ontologies, Gero’s<sup>9</sup> Function–Behavior–Structure (FBS) model has been applied in many cognitive studies.<sup>10–13</sup> Researchers argue that it is potentially capable of capturing most of the meaningful design processes,<sup>13</sup> and the transitions between design issues are clearly classified into eight design processes. The FBS ontology contains three classes of variables, called design issues: Function (F), Behavior (B), and Structure (S). Function (F) represents the design intentions or purposes of the artifact being designed;



**Figure 1.** The FBS ontology.<sup>14</sup>

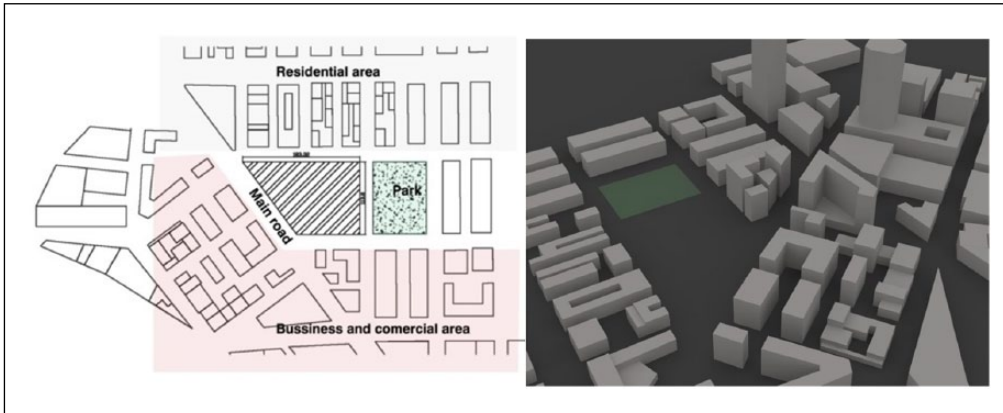
behavior (B) represents the behavior derived (Bs) or expected from the structure (Be) of the artifact; and structure (S) represents the components that make up an artifact and their relationships. The model includes two external design issues: requirements (R) and descriptions (D). The first of these represents design issues that come from outside design, commencing with those articulated by the client, and the second, descriptions, is any documentation of the design. These two additional design issues can be expressed in terms of F, B, or S and do not require an expansion of the ontology. Transitions between design issues are design processes and do not require a separate ontology as they are directly derivable from the FBS ontology. Figure 1 presents the FBS ontology indicating the eight design processes—formulation, analysis, evaluation, synthesis, documentation, and three reformulations I, II, III. Among the eight design processes, the three types of reformulation processes are suggested to be the dominant processes that potentially capture innovative or creative aspects of designing by introducing new variables or new directions.<sup>12</sup> The FBS ontology is claimed to be a universal coding scheme for various design environments.<sup>13</sup> In this study, the FBS ontology was used as the model for developing the coding scheme in the protocol study that was used to capture and produce behavioral data of designers as they are designing. The FBS ontologically based coding scheme consists of the six design issues of requirements (R), function (F), expected behavior (Be), structure (S), behavior from structure (S), and description (D).

## Research design

### *Research method—protocol analysis*

Protocol analysis is a method for turning qualitative verbal and gestural utterances into data.<sup>1,2</sup> It has been used extensively in design research to develop an understanding of design cognition.<sup>12,15,16</sup> According to Akin,<sup>17</sup> a protocol is the record of behaviors of designers using sketches, notes, videos, or audio. After collecting the protocol data, a coding scheme is applied to categorize the collected data, enabling detailed study of the design process in the chosen design environments. As Gero and Tang<sup>18</sup> state, protocol analysis has become the prevailing experimental technique for exploring the understanding of design.

Usually in protocol analysis, concurrent and retrospective protocol collection methods can be applied in design experiments.<sup>1,19</sup> A concurrent protocol involves participants in an experiment verbalizing their thoughts when working on a specific task—also called the “think aloud” method—whereas a retrospective protocol explores what designers were thinking while designing, a process which is applied as soon as they have finished the design task. Some studies have compared these two protocol collection methods. For instance, Kuusela and Pallab<sup>20</sup> argue that concurrent protocols are more suitable for examining the design



**Figure 2.** Site model provided to the designers during the experiments.

process and can generate larger numbers of segments, while retrospective protocols are more suitable for examining design outcomes. Another example of this comparison is Gero and Tang's<sup>18</sup> study exploring design processes. Their results show that concurrent and retrospective protocols lead to very similar outcomes in terms of exploring designers' intentions during design processes. But they also conclude that concurrent protocols are an efficient and applicable method by which to understand design processes. Retrospective protocols are commonly believed to be less intrusive to the design processes.

Importantly, protocol analysis of this type deals with a relatively small number of samples, but it enables an in-depth exploration of the samples. Thus, a study of the cognitive behavior of eight designers is both acceptable and in keeping with past research in this field because of the quality and depth of information that is recorded and analyzed. However, for this reason, we also cannot generalize the results of this research to describe the actions or behaviors of a much larger population of designers. Nevertheless, from such studies, important patterns, which are repeated by designers, can be used to provide an increased level of understanding of the design process.

### **Experiment setting**

Eight designers participated in the experiment, each of whom was a professional architect with an average of 8 years of experience in architectural design, and no less than 2 years of experience using parametric design.

The experiment consisted of two design sessions: one session used Rhino as the GME, and the other session used Rhino and Grasshopper as the PDE. Designers were given 40 min for each design session. Task 1 is a community center design, and Task 2 is a shopping center design, both containing specific function requirements. A pre-modeled site was provided to the designers (Figure 2). Because this study was focused on exploring designers' behavior at the conceptual design stage, designers were only required to consider concept generation, simple site planning, and general function zoning. No detailed plan layout was required. The design sessions and tasks were randomly assigned to the different designers. The order of design sessions was also randomly arranged, as shown in Table 1. During the experiment, designers were not allowed to sketch manually so that almost all their actions happened on the computer to ensure that the design environment is purely within the PDE and GME. This allows us to minimize the impacts of other variables and focus on the two different design environments for comparison during the comparative analyses. During the experiment, both designers' activities and their verbalizations were video-recorded.

**Table 1.** Design session allocation.

Participants	Designer 1	Designer 2	Designer 3	Designer 4	Designer 5	Designer 6	Designer 7	Designer 8
Design session 1	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2
Design session 2	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1	Task 2	Task 1

**Table 2.** General coding information expressed as mean values and standard deviations of the eight participants.

	Design environment	Time (min)	Number of segments	Coded percentage (%)	Speed (segments/min)
Mean	GME	44	224	92.2	5.1
	PDE	48	244	92.2	4.8
SD	GME	11.22	45.32	4.29	1.20
	PDE	7.43	29.71	3.54	0.53

GME: geometric modeling environment; PDE: parametric design environment; SD: standard deviation.

## Analysis results

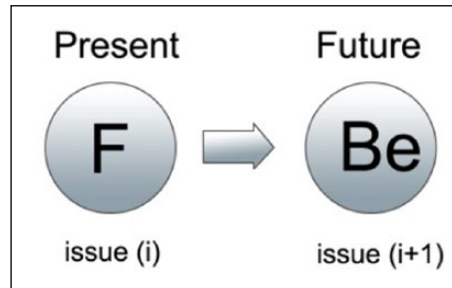
### General results

The protocol analysis takes the video and verbal utterances of the participants and transforms them into a sequence of semantic symbols, called design issues, based on the FBS coding scheme. This sequence of design issues becomes the basis for all quantitative results that follow.

This protocol study employed an integrated segmentation and coding method. Segmentation is the process of breaking utterances into blocks that contain a single design issue. The segmentation and coding process are based on the “one segment one code” principle.<sup>21</sup> This means there is no overlapped code or multiple codes for one segment. If there are multiple codes for one segment, the utterances in that segment will be further divided until there is only one code per segment, that is, one design issue in a segment. Table 2 contains the general information of the coding coverage. The numbers shown in Table 2 are the mean values and standard deviations of the eight protocols. The mean numbers of segments are, respectively, 244 in the PDE and 224 in the GME. Designers also spent more time in the PDE session (48 min) than in the GME (44 min). On average, over 92.2% of segments were coded as FBS codes. Segments not coded included those associated with communication and software management.

### First-order Markov analysis results

The segments in a coded protocol may be thought of more generally as states in the design thinking of the designer. A Markov model describes the probabilities of moving from one state to another.<sup>22,23</sup> When used in design research, it describes the tendency of future design moves using the data gathered from that design session. Kan and Gero<sup>13,24</sup> adopted the Markov model using the FBS ontologically based coding to describe cognitive design processes. Within the context of the FBS ontology, the Markov matrix is a quantitative tool to study design activities based on the transition probabilities between design issues. It can also be used to study transitions between design processes. There are two types of Markov models of interest here: the first-order Markov model and the second-order Markov model. The first-order Markov model expresses the probability of moving to a future design issue depending only on the current design issue, without considering the past design issue (Figure 3). The second-order Markov model expresses the probability of moving to a future design issue depending on the current design issue and the previous design issue that led to the current



**Figure 3.** An example of the first-order Markov model using the FBS ontology, where the arrow contains the probability of the next segment after an F is Be.

**Table 3.** The first-order Markov model analysis.

	R		F		Be		Bs		S	
	GME	PDE	GME	PDE	GME	PDE	GME	PDE	GME	PDE
R	0.20	0.20	0.18	0.30	0.39	0.21	0.11	0.03	0.11	0.26
F	0.03	0.02	0.23	0.11	0.29	0.26	0.23	0.19	0.23	0.43
Be	0.03	0.01	0.10	0.09	0.18	0.17	0.24	0.31	0.46	0.43
Bs	0.02	0.01	0.06	0.07	0.22	0.22	0.25	0.26	0.45	0.45
S	0.01	0.01	0.05	0.06	0.14	0.19	0.40	0.34	0.41	0.41

GME: geometric modeling environment; PDE: parametric design environment; R: requirements; F: function; Be: behavior; Bs: behavior from structure; S: structure.

design issue. This article only presents the results of the first-order Markov model that is calculated using more transitions than the second-order Markov model, producing a rich and reliable dataset.

The values in the Markov model are calculated using LINKODER, a software tool used in protocol analysis.<sup>21</sup> The results of the first-order Markov model in the GME and the PDE are presented in Table 3. The numbers in the table are the average values of the eight designers of the transition probabilities of moving from one design issue to another design issue. Table 3 is read as follows: if a designer is currently at the design issue designated by the code in the left-hand column, each column in that row shows the probability of moving to the design issue designated in the column label in both the GME and PDE.

In order to determine whether the numerical values in Table 3 are different in a statistically significant way, paired sample t-tests are carried out. The results of this are shown in Table 4. This analysis indicates that there is no significant difference between the two design environments except for the transition of function (F) to structure (S) ( $p=0.04 < 0.05$ ), indicating that there are relatively more F-S transitions in the PDE than in the GME. From this, we can infer that designers' tendency of moving from one design issue to another does not vary because of the method used, except for the move from F to S.

### Markov analysis at different design stages

Treating a design session as a single unit of activity may mask behavior that is more localized. Therefore, we divide the protocols into thirds and carried out the same analysis on each third. Tables 5–7 show the paired sample t-test of first-order Markov model analysis at these three different design stages labeled early, mid, and end stages. From the results in these three tables, we can see that at the early design stage, there are no

**Table 4.** Paired sample t-test of first-order Markov model analysis between GME and PDE.

	F					Be					Bs					S				
	Mean	Standard deviation	t	Significance (two-tailed)	Standard deviation	Mean	Standard deviation	t	Significance (two-tailed)	Standard deviation	Mean	Standard deviation	t	Significance (two-tailed)	Standard deviation	Mean	Standard deviation	t	Significance (two-tailed)	
R	-0.001	0.365	-0.010	0.993	0.120	0.480	0.707	0.502	-0.183	0.379	-1.363	0.215	-0.083	0.141	-1.653	0.142	0.148	0.393	1.061	0.324
F	-0.014	0.080	-0.485	0.642	-0.121	0.157	-2.189	0.065	-0.033	0.010	-0.924	0.386	-0.038	0.092	-1.154	0.286	0.208	0.141	4.157	0.004*
Be	-0.020	0.027	-2.117	0.072	-0.011	0.067	-0.471	0.652	-0.013	0.079	-0.447	0.668	0.074	0.147	1.420	0.198	-0.035	0.062	-1.594	0.155
Bs	-0.014	0.024	-1.629	0.147	0.009	0.063	0.392	0.707	-0.008	0.065	-0.328	0.753	0.010	0.080	0.354	0.734	0.005	0.173	0.082	0.937
S	0.000	0.012	0.000	1.000	0.014	0.067	0.583	0.578	0.049	0.090	1.532	0.169	-0.054	0.101	-1.508	0.175	-0.004	0.140	-0.076	0.942

\*Significance (Two-tailed)<0.05.

GME: geometric modeling environment; PDE: parametric design environment; R: requirements; F: function; Be: behavior from structure; Bs: structure.

**Table 5. Paired sample t-test of first-order Markov model analysis at early design stage.**

Early design stage	F				Be				Bs				S							
	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)				
R	-0.030	0.363	-0.234	0.822	0.094	0.487	0.545	0.603	-0.146	0.409	-1.011	0.345	-0.066	0.142	-1.317	0.229	0.148	0.393	1.061	0.324
F	-0.018	0.120	-0.411	0.693	-0.041	0.254	-0.458	0.660	-0.060	0.143	-1.187	0.274	-0.068	0.103	-1.850	0.107	0.185	0.236	2.220	0.062
Be	-0.036	0.050	-2.057	0.079	-0.023	0.141	-0.451	0.665	-0.066	0.110	-1.707	0.132	0.100	0.120	2.359	0.050	0.023	0.144	0.441	0.673
Bs	-0.086	0.210	-1.180	0.277	0.041	0.116	1.005	0.348	0.051	0.109	1.335	0.224	-0.005	0.185	-0.076	0.941	-0.001	0.136	-0.026	0.980
S	0.001	0.043	0.082	0.937	-0.035	0.169	-0.585	0.577	0.041	0.119	0.984	0.358	-0.020	0.131	-0.430	0.680	0.013	0.253	0.139	0.893

\*Significance (Two-tailed)<0.05.

R: requirements; F: function; Be: behavior; Bs: behavior from structure; S: structure.



**Table 6.** Paired sample t-test of first-order Markov model analysis at mid-design stage.

Mid-design stage	F				Be				Bs				S			
	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)
R	-	-	-	-	-0.063	0.177	-1.000	0.351	-0.125	0.354	-1.000	0.351	-0.063	0.177	-1.000	0.351
F	-	-	-	-	-0.109	0.235	-1.309	0.232	0.018	0.348	0.142	0.891	-0.159	0.439	-1.022	0.341
Be	-0.001	0.021	-1.000	0.351	-0.033	0.162	-0.567	0.589	0.073	0.146	1.407	0.202	-0.040	0.194	-0.582	0.579
Bs	-0.004	0.011	-1.000	0.351	0.000	0.094	0.000	1.000	-0.034	0.114	-0.837	0.430	0.131	0.117	3.175	0.016*
S	-	-	-	-	0.015	0.041	1.042	0.332	0.070	0.136	1.451	0.190	-0.061	0.165	-1.051	0.328

\*Significance (Two-tailed)<0.05.

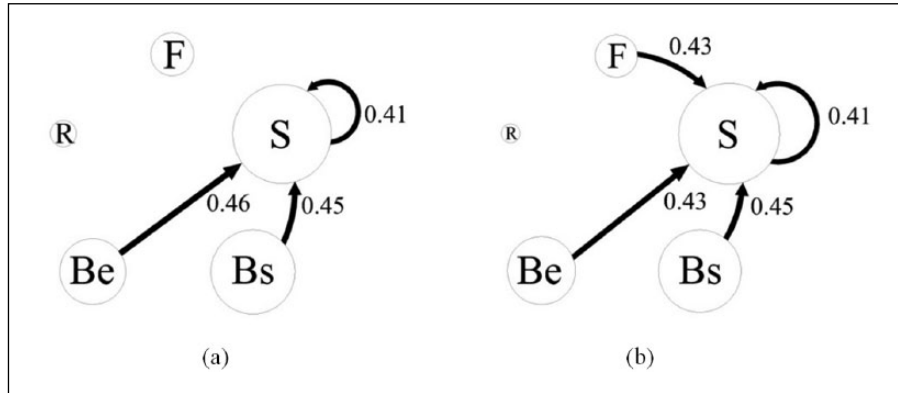
R: requirements; F: function; Be: behavior; Bs: behavior from structure; S: structure.

**Table 7. Paired sample t-test of first-order Markov model analysis at end design stage.**

End design stage	F				Be				Bs				S			
	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)	Mean	Standard deviation	t	Significance (two-tailed)
R	0.063	0.177	1.000	0.351	-0.063	0.417	-0.424	0.685	-	-	-	-	-	-	-	-
F	-	-	-	-	-0.163	0.353	-1.303	0.234	0.223	0.363	1.733	0.127	-0.188	0.382	-1.388	0.208
Be	0.006	0.018	1.000	0.351	0.043	0.091	1.326	0.227	-0.034	0.138	-0.694	0.510	0.204	0.392	1.470	0.185
Bs	-	-	-	-	0.008	0.068	0.312	0.764	-0.021	0.091	-0.658	0.532	-0.065	0.117	-1.567	0.161
S	-0.004	0.011	-1.000	0.351	0.016	0.041	1.111	0.303	0.008	0.089	0.237	0.819	-0.100	0.240	-1.178	0.277

\*Significance (Two-tailed)<0.05.

R: requirements; F: function; Be: behavior; Bs: behavior from structure; S: structure.



**Figure 4.** (a) Transitions above the threshold in the first-order Markov model in the GME. (b) Main transitions of the first-order Markov model in the PDE.

statistically significant differences between GME and PDE in terms of design issue transitions. At the mid-design stage, the transitions from function (F) to structure (S) and from behavior from structure (Bs) back to behavior from structure (Bs) exhibit statistically significant differences between GME and PDE. While at the end design stage, there is a statistically significant difference found in transitioning from F to S transitions. The p value for F-S transition decreases from  $p < 0.062$  in the early stage through  $p < 0.038$  in the mid-stage to  $p < 0.007$  at the end stage. This suggests that the difference of the F to S transition between PDE and GME tends to increase toward the end of design session.

## Design patterns in the PDE

### *F to S design transition*

A descriptive diagram of the first-order Markov model analysis in the GME and the PDE is given in Figure 4. The circles labeled with the FBS codes represent the design issues, and the size of a circle represents the frequency of occurrence of that design issue. Each arrow shows the transition from one state to the other, and the thickness of the line represents a measure of the transition probability between design issues. To demonstrate the main activities of the designers, we select the transitions with the probability value larger than 0.4 and present them in Figure 4. The probability of 0.4 is selected as the threshold to abstract the model; this threshold is twice the random transition probability. In the coding used, each variable has five other states to go to, which means that the random probability is 0.2. The graphs in Figure 4 show that the main patterns of design moves are  $Be \rightarrow S$ ,  $Bs \rightarrow S$ , and  $S \rightarrow S$ ; the transition probabilities are very similar between the GME and the PDE. The primary difference between the GME and the PDE is that the transition probability from F to S is above the threshold in the PDE and below the threshold in the GME.

Within the context of the FBS ontology, this process of transitioning directly from function (F) to structure (S) is excluded from routine ways of design (excluded from the eight design processes expressed in FBS model). Previous research suggests from the study of software designers' behavior that the transition from F to S is a typical design process that occurs frequently. During the F-S process, designers select an existing structure/solution for the particular function/design problem based on their experience or knowledge, which is the process of applying a learned, existing design pattern to the problem rather than a design reasoning process.

Software designers use design patterns when programming and scripting.<sup>25,26</sup> We can infer that when architects apply programming and scripting in their design, such as in a PDE, they exhibit a similar

characteristic of using design patterns. Design patterns are an important concept in both architectural design and software design. In software design, it assists software designers in working more efficiently and makes programming and scripting process traceable. In the PDE, if we can generalize these transitions to design patterns, it would be of assistance to architects in conceptualizing their scripting process. The Markov model provides empirical evidence for the existence of a design pattern when the transition probability is above 0.4.

### *Design patterns in the PDE*

From the Markov model analysis results, we found that design patterns are adopted more frequently in the PDE than in the GME. The idea of design patterns was first introduced by Christopher Alexander:<sup>27</sup>

each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.

That is, a pattern is a documentation of a solution suitable for certain kind of design problems that may occur frequently.

Patterns usually come from a designers' experience,<sup>26</sup> which can be seen as an induction process. Designers generalize examples from their own design experience or from observations of other designers, and abstract the problem-solution pair, and formalize these into "patterns" which can be reused. Those generated patterns can be improved and combined into a network of connections depending on their design purpose.<sup>28</sup> Woodbury<sup>3</sup> defines that "A pattern is a generic solution to a well-described problem. It includes both problem and solution, as well as other contextual information." A design expert has accumulated a large number of examples of problems and solutions in a specific domain.<sup>29</sup> The pattern itself is abstract; when a designer applies a pattern, the designer can revise the pattern to their own preference, or to the specific context of designing.

In the software design domain, educators have found that Alexander's work on design patterns provided a strong foundation for re-usable software design. For instance, Gamma et al.<sup>25</sup> define patterns as a tool to describe compositional ideas in computer programming. This matches our analysis results that in parametric design, more design patterns were used than in geometric design alone. Therefore, we can infer that a feature of programming in a parametric design might affect the utility of design patterns during computational design process. The design patterns identified in the PDE suggest possibilities of using programming-based design patterns. These encode computational thinking and are therefore potentially generalizable.<sup>30</sup>

## **Conclusion**

This article has presented the results of a protocol study that explores the phenomenon of using design patterns in a PDE and GME. From the study, we found that there are more design patterns used in the PDE than in the GME, and this difference increases toward the end of design session implying either that there are more opportunities to apply patterns as the design proceeds or that learning takes place during the design activity. The difference between the two environments is in the additional design pattern in the PDE of  $F \rightarrow S$ , that is, going from function to structure, which is only possible as a consequence of experience as there is no direct design reasoning link between them when designing. Since the main difference between the two design environments is that rule algorithms are used in the PDE and not in the GME, we can assume that the rule algorithm feature affects the development and use of design patterns more during the design process. That is to say in a rule algorithm-related design environment, designers tend to adopt and use existing patterns based on their experience.

In software design, design patterns are used both in education and in practice. The use of design patterns by practicing architects as shown by the empirical evidence presented from these protocol studies implies, if the results are generalizable, that the opportunity exists to involve design patterns in teaching parametric modeling in design.

Further analysis of the empirical data in the protocol is needed to determine whether these patterns re-use existing patterns or are developed during the design session as a consequence of the designer's current activity.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

This research has been supported in part by the National Science Foundation grants CMMI-1161715 and EEC-1463873 to the second author (J.S.G.). 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. Ericsson KA and Simon HA. *Protocol analysis: verbal reports as data*. Cambridge, MA: MIT Press, 1993.
2. Gero JS and Mc Neill T. An approach to the analysis of design protocols. *Design studies* 19(1): 21–61.
3. Woodbury R. *Elements of parametric design*. New York: Routledge, 2010.
4. Kolarevic B. *Architecture in the digital age: design and manufacturing*. New York: Spon Press, 2003.
5. Iordanova I, Tidafi T, Guité M, et al. Parametric methods of exploration and creativity during architectural design: a case study in the design studio. In: *the 13th international conference on computer aided architectural design futures* (ed. Tidafi T and Dorta T), Presses de l'Université de Montréal, Montréal, QC, Canada, 2009, pp.423–439, [http://papers.cumincad.org/data/works/att/cf2009\\_423.content.pdf](http://papers.cumincad.org/data/works/att/cf2009_423.content.pdf)
6. Schnabel MA. Parametric designing in architecture. In: *Proceedings of the 12th international conference on computer aided architectural design futures* (ed. Dong A, Moere V and Gero JS), Sydney, NSW, Australia: Springer, 2007, pp.237–250.
7. Lee JH, Gu N, Jupp J and Sherratt S. Evaluating creativity in parametric design processes and products: a pilot study. In: *Proceedings of international conference of design computing and cognition, DCC'12* (ed. Gero J), Austin, TX: Springer, 2014, pp.165–183.
8. Chien S-F and Yeh Y-T. On creativity and parametric design—A preliminary study of designer's behaviour when employing parametric design tools. In: *Proceedings of eCAADe 2012* (ed. Achten H, Pavlíček J and Hulín J), Czech Republic, 2012, eCAADe in cooperation with Mediatecture, [http://papers.cumincad.org/data/works/att/ecaade2012\\_223.content.pdf](http://papers.cumincad.org/data/works/att/ecaade2012_223.content.pdf)
9. Gero JS. Design prototypes: a knowledge representation schema for design. *AI Magazine* 11(4): 26–36.
10. Gero J and Tang H-H. Concurrent and retrospective protocols and computer-aided architectural design. In: *CAADRIA 1999* (ed. Xie Z and Qian J), Shanghai, 1999, pp.403–410, <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=EEADCA93B37CB7EF2E230A000297E46D?doi=10.1.1.25.7151&rep=rep1&type=pdf>
11. Jiang H. *Understanding senior design students' product conceptual design activities—a comparison between industrial and engineering design students*. Singapore: National University of Singapore, 2012.
12. Kan JWT and Gero JS. Acquiring information from linkography in protocol studies of designing. *Design studies* 29(4): 315–337.
13. Kan JWT and Gero JS. Using the FBS ontology to capture semantic design information in design protocol studies. In: McDonnell J and Lloyd P (eds) *About designing: analysing design meetings*. New York: Taylor & Francis, 2009, pp.213–229.
14. Gero J and Kannengiesser U. The situated function-behaviour-structure framework. *Design studies* 25(4): 373–391.
15. Atman CJ, et al. A comparison of freshman and senior engineering design processes. *Design studies* 20(2): 131–152.

16. Suwa M and Tversky B. What do architects and students perceive in their design sketches? A protocol analysis. *Design studies* 18(4): 385–403.
17. Akin O. *Psychology of architectural design*. London: Pion, 1986.
18. Gero J and Tang H-H. The differences between retrospective and concurrent protocols in revealing the process-oriented aspects of the design process. *Design studies* 22(3): 283–295.
19. Dorst K and Dijkhuis J. Comparing paradigms for describing design activity. *Design studies* 16(2): 261–274.
20. Kuusela H and Pallab P. A comparison of concurrent and retrospective verbal protocol analysis. *American Journal of Psychology* 113(3): 387–404.
21. Pourmohamadi M and Gero JS. LINKOgrapher: an analysis tool to study design protocols based on FBS coding scheme. In: Culley S, et al. (eds) *Design theory and methodology, design society*. Glasgow: Design Society, 2011, pp.2-294–2-303.
22. Ching WK and Ng MK. *Markov chains: models, algorithms and applications*. New York: Springer, 2006.
23. Meyn SP and Tweedie RL. *Markov chains and stochastic stability*. New York and Cambridge: Cambridge University Press, 2009.
24. Kan JWT and Gero JS. Exploring quantitative methods to study design behavior in collaborative virtual workspaces. In: *Proceedings of CAADRIA 2010*, (ed. Dave B, Li AI, Gu N and Park H-J), Hong Kong, 7–10 April 2010, pp.273–282.
25. Gamma E, et al. Design patterns: abstraction and reuse of object-oriented design. In: Manfred B and Ernst D (eds) *Software pioneers*. New York: Springer-Verlag, 2002, pp.701–717.
26. Fowler M. *Patterns of enterprise application architecture*. Boston, MA: Addison-Wesley, 2003.
27. Alexander C, Ishikawa S and Silverstein M. *A pattern language: towns, buildings, construction*. New York: Oxford University Press, 1977.
28. Alexander C. *The timeless way of building*. Oxford: Oxford University Press, 1979.
29. Razzouk R and Shute V. What is design thinking and why is it important? *Review of Educational Research* 82(3): 330–348.
30. Wiggins GA. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge Based System* 19(7): 449–458.