Participant Observation can Discover Design Patterns in Parametric Modeling

Cheryl Z. Qian, Victor Y. Chen, and Robert F. Woodbury
School of Interactive Arts & Technology, Simon Fraser University

Our research aims to understand the mid-level patterns of work that recur across designers and tasks. Our users comprise active architects and civil engineers. The hypothesis is that making such patterns explicit will result in improved expert work practices, in better learning material and suggestions for improvements in parametric design. The literature shows that patterns express design work at a tactical level, above simple editing and below overall conception. We conducted a user experience study based on Bentley's GenerativeComponents, in which geometry can be related, transformed, generated, and manipulated parametrically within a user-defined framework. After interviewing the system's chief, we ran a participant-observer study in the January 2007 SmartGeometry workshop. We engaged designers through the role of tutor and simultaneously observed and discussed their design process. We found clear evidence of designers using patterns in the process and discerned several previously unknown patterns. In February at another 10-day workshop, we found more evidence supporting prior findings. The paper demonstrates that participant observation can be an efficient method of collecting patterns about designers' work and introduces such new patterns. We believe these patterns may help designers work at more creative levels and may suggest new ideas of interest to CAD application developers.

1 INTRODUCTION

The overall goal of our research is to observe, understand, support and augment user intention and action in parametric design systems. In this study we want to analyze and collect the mid-level patterns of work that recur across designers and tasks. In the domain of parametric modeling systems, the target users are active designers such as architects and civil engineers. We want to investigate if making such patterns explicit will result in improved expert work practices, better learning material, and suggestions for improving parametric design interfaces. More detailed questions include the following: Can a designer's learning and working process in parametric modeling applications be well modeled by patterns? What is the proper method to collect and model such patterns? Can patterns be used to express intention? Can design work be well supported by tools that can express intentions?

2 BACKGROUND OF THIS STUDY

We reviewed the literature from parametric modeling, expertise, intentional stance and user modeling to gain the background knowledge of this research.

2.1 PARAMETRIC MODELING

As a family of computing systems intended to facilitate design activities, conventional architectural CAD (Computer Aided Design) software comes in two flavors: drawing tools with symbol libraries and "intelligent" tools that offer component-level design, for instance: walls, doors and windows. The main problem with these modeling tools is that the components are fabricated by CAD software vendors, limiting adaptability—a designer cannot add a structural element if there are no structural elements in the library to add. The majority of professional architects find that these two types of product will cater to most design tasks, although with drawing tools, changes and edits could impact hundreds of drawings, which have to be done manually. At the manual editing stage, the task is pure tedium—error detection and repair. Yet designers must pay full attention during this important contractual and legal process. From a cognitive perspective, attention is the means by which we actively process a limited amount of information from the enormous amount of information available to us through our senses, stored memories and other cognitive processes (Benyon et al. 2005). Many small mistakes on the drawings are hard to detect. Actually, human error is a critical contributor to lapses in system design (Wickens et al. 2004). The "intelligent" solutions aim to overcome this by using object-oriented design but these concentrate on producing documentation and usually fail to model buildings with innovative form.

Parametric modeling software has introduced into practice computational mechanisms and interfaces for representing variation in design. These work best when variation is continuous and distinctly different alternatives are not part of the model. Using parametric modeling, it is possible to develop models that support discrete variation, but it is very difficult to understand the range of possibilities entailed by such models. Parametric modeling interfaces thus provide partial support for expressing variation and, because they are increasingly used in practice, a means by which new variation

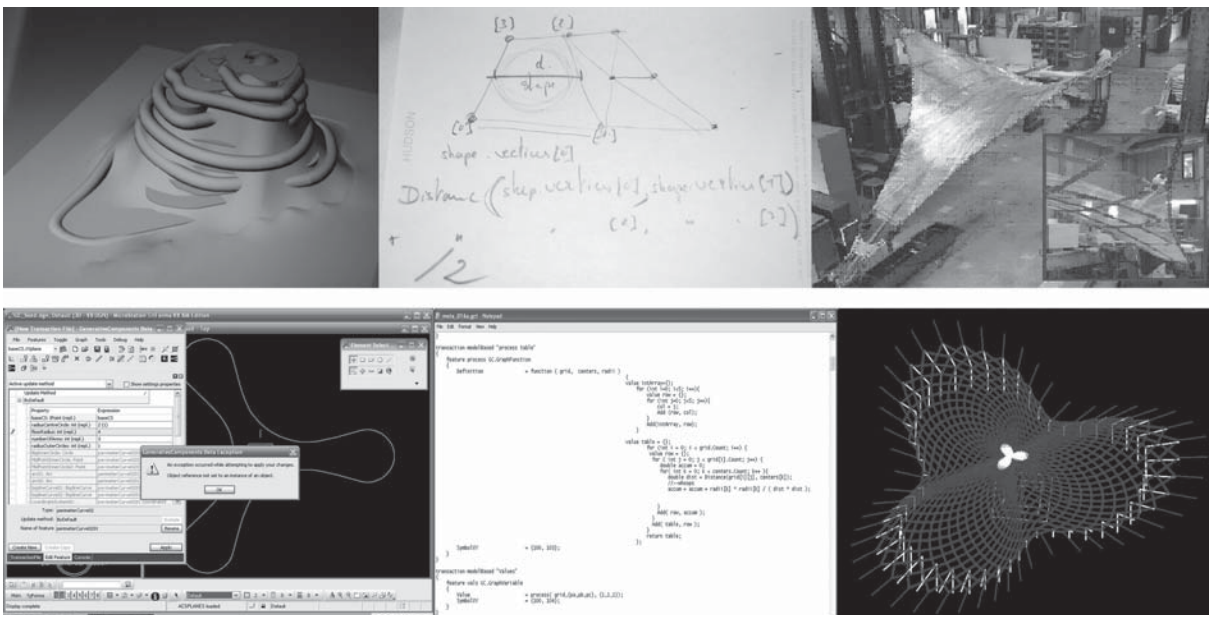

FIGURE 1 Primary material of photos and screenshots collected from the workshop.

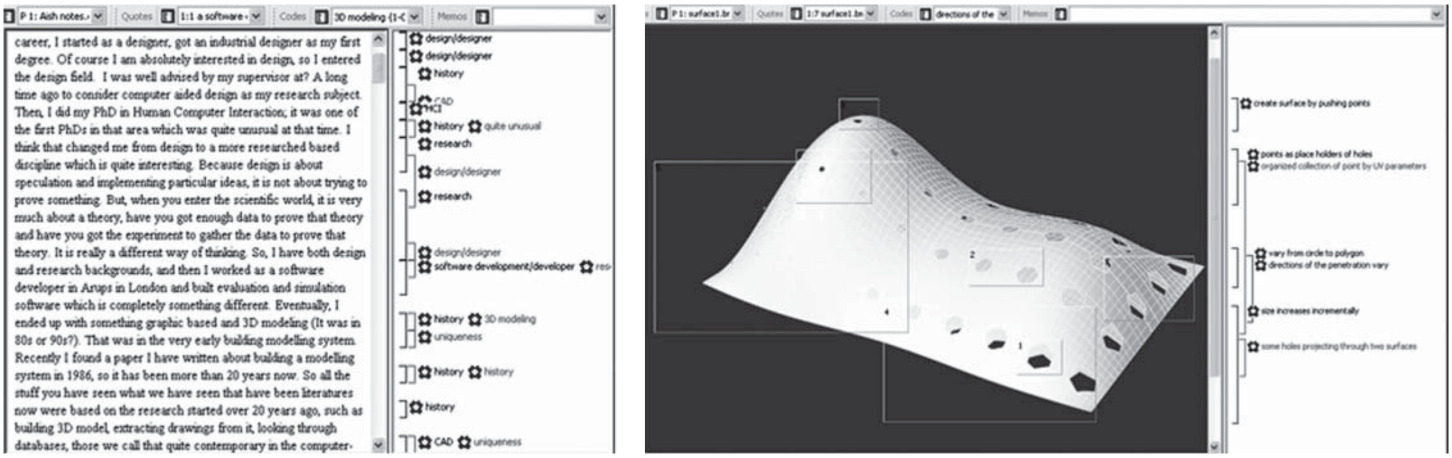

FIGURE 2 Screenshots of the coding process in ATLAS.ti.

techniques can be tested in actual use.

Bentley's GenerativeComponents (Aish 2003), developed by Robert Aish, Chief Scientist at Bentley Systems, is not about walls, doors windows. This tool provides an environment in which geometry (lines, arcs, circles, solids, and surfaces) can be related, transformed, generated and manipulated within a user-defined framework. The end results are complex and sculpted geometry that can be quickly generated and manipulated in real-time, allowing design exploration and variation. Currently, this system has been used to promote and educate parametric modeling in practice, reaching firms such as Foster and Partners (whose recent works include the British and Smithsonian Museum courtyard roofs and the SwissRE headquarters in London), Arup Sports (Beijing Olympic Stadium) and Kohn Peterson Fox (World Bank headquarters). Through hosting SmartGeometry workshops around the world, the community of Generative Components users is evaluating and improving the structure and interface to make it more communicative and supportive for architects, civil engineers and constructors. These events provide an opportunity for us to observe, understand and suggest new interface ideas.

2.2 EXPERTISE IN DESIGN

Understanding designers includes recognizing and supporting their expertise in design. Expertise consists of those characteristics, skills, and knowledge sets that distinguish experts from novices and less experienced people. Many accounts of the development of expertise emphasize that it comes about through long periods of deliberate practice (Ericsson 1999). In recent years, the disciplines of cognitive science and artificial intelligence have devoted a great deal of attention to the nature of expert problem solving and decision making in professional-level tasks. The goal of cognitive science research has been to gain an understanding of the differences between the behavior of experts and novices, and possibly to learn more about how novices can become experts. A key competency of an expert is the ability mentally to stand back from the specifics of the accumulated examples, and form more abstract conceptualizations pertinent to their domain of expertise (Benyon et al. 2005). Experts are believed to be able to store and access information in larger cognitive "chunks" than novices, and to recognize underlying principles, rather than focusing on the surface features of problems.

We are concerned with comprehending expertise in the context of design process. Cross (2004) claimed that although designers change goals and constraints as they design, they appear to retain to their principal solution concept for as long as possible, even when detailed development of the scheme reveals unexpected difficulties and shortcomings in the solution concept. Duncker defined *functional fixedness* as being a "mental block against using an object in a new way that is required to solve a problem" (Duncker 1945). On the other hand, Cross found that creative design solutions arise especially when there is a conflict to be resolved between the designer's own high-level problem goals and the criteria for an acceptable solution established by clients or other requirements (Cross 2004). It is this *break-down* that helps the expert designer to think outside the box and be more creative. Perhaps tools such as patterns can remind designers of functional fixedness in the process and inspire designers during inevitable break-downs in the design.

2.3 INTENTIONAL STANCE

In the process of formulating our research question, D. C. Dennett's *intentional stance* has become a crucial concept. It is the strategy of interpreting the behavior of an entity (person, animal, artifact, or the like) by treating it as if it were a rational agent that governed its "choice of "action" by a "consideration" of its "beliefs" and "desires" (Dennett 1995). Here is how it works: "first you decide to treat the object whose behavior is to be predicted as a rational agent; then you figure out what beliefs that agent ought to have, given its place in the world and its purpose" (Dennett 1995). Then you figure out what desires it ought to have, on the same considerations, and finally you predict that this rational agent will act to further its goals in the light of its beliefs. As Dennett states, "a little practical reasoning from the chosen set of beliefs and desires will in most instances yield a decision about what the agent ought to do; that is what you predict the agent will do" (Dennett 1995). Dennett's argument is that the intentional stance gives us more predictive power than any other method. The purpose of this level of reverse engineering is not just to pry out secrets of history but also to predict events in the present. The strategy of intentional stance not only helped construct our research question, but will also act as the main perspective for us to analyze and infer patterns to support designers' intentions.

2.4 USER MODELING

A fundamental objective of HCI research is to understand how to make systems more usable, more useful, and to provide users with experiences fitting their specific background knowledge and objectives. Designers of systems face the formidable task of writing software for millions of users (at design time) while making it work as if it were designed for each individual user (only known at use time) (Fischer 2001). User modeling research is one of the main approaches attempting to address these issues.

In user modeling use-time and design-time are blurred. If the system is constantly adapting or is being adapted to users, use time becomes a different kind of design-time (Henderson and Kyng 1991). The need to support a broad class of different users leads to high-functionality applications with all their associated possibilities and problems. Fischer argues that a feasible design strategy to support users is that system designers make assumptions about classes of users and sets of tasks in which users want to engage a design methodology leading to domain-oriented systems (Fischer 1994). Most of the user-modeling software appears to use this strategy. However, in Don Norman's recent article "Human-centered design considered harmful," he claimed that activity-centered design could be more successful than human-centered design because of its nature and the communication of intention between builders and designers (Norman 2005). Patterns are a means for expressing activities. If we focus on analyz-

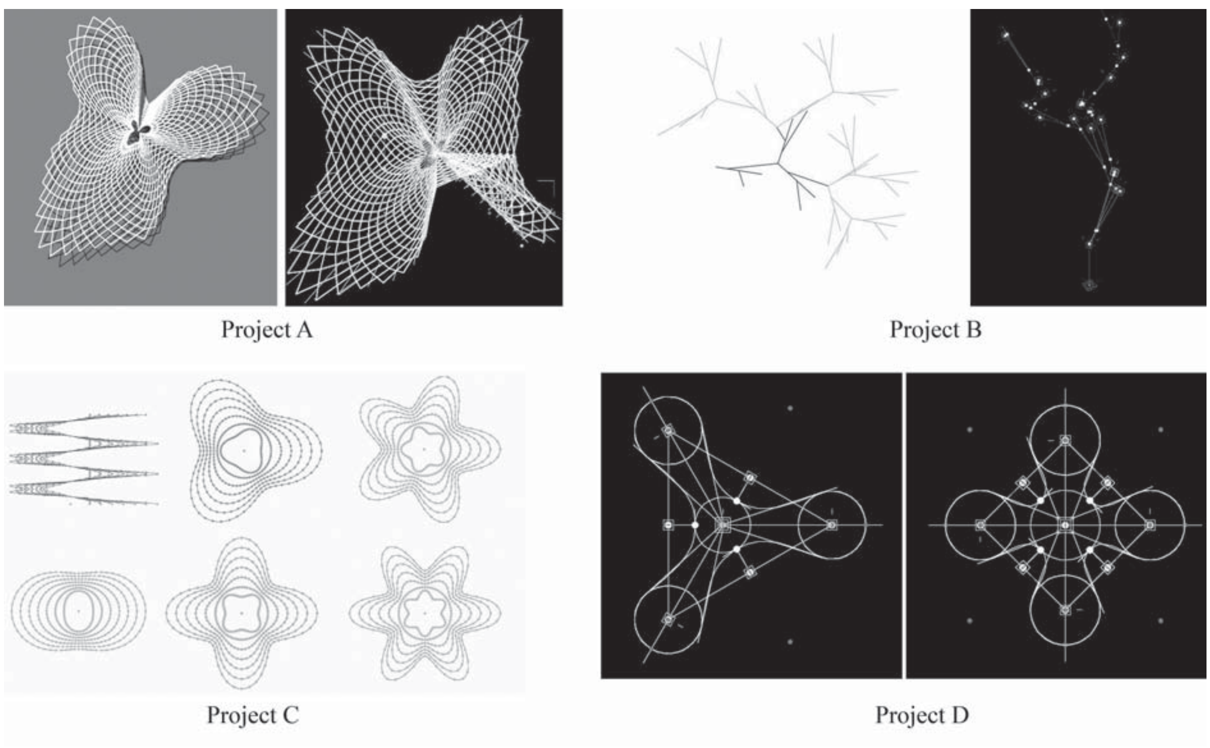

FIGURE 3 Examples of projects which involve the *Branching* pattern.

ing user activities into patterns, it could be another workable approach to provide the "right" information to different users.

3 COLLECT AND DESIGN PATTERNS

One of the most important recent ideas in software development is that of a *design pattern*. The term *design pattern* originated with Christopher Alexander to describe an. Patterns express design work at a tactical level, above simple editing and below overall conception. In the process of parametric design, patterns can be observed as a general repeatable solution to a recurring modeling problem. Architects may use the same pattern in different circumstances and may also derive new patterns as they work. This concept originated in urban architecture but has been adapted quite successfully to software engineering (Gamma et al. 1995) and extended to other disciplines such as interaction design (Tidwell 2006), web usability (Graham 2003), education science (Bergin et al. 2001) and communication (Schuler 2002). Patterns today are very useful because they provide a language for communication among designers. Rather than having to explain a complex idea from scratch, the group of designers can just mention a pattern by name and everyone will know, at least roughly, what is meant. In this sense patterns are an excellent vehicle for the collection and dissemination of the anecdotal and unquantifiable data that Borenstein argues need to be collected before we can see real advances in a developing process (Borenstein 1991). Before observing and analyzing design patterns in the process of parametric modeling, we compare pattern approaches adopted in different domains to understand the nature of patterns and the methods of pattern authoring.

3.1 PATTERN LANGUAGES MAY OVERDETERMINE OUTCOMES

Through the idea of a *pattern language*, architect Christopher Alexander argues that people have an innate ability for design that parallels their ability to speak, and that every community has its own shared design language particular to its place and needs(Alexander et al. 1977). David Week stresses that: "A pattern is a hypothesis or rule-of-thumb, which can be discussed and evaluated relatively independently of other patterns. Patterns are like the individual codons in the DNA sequence for an organization's workplace" (Week 2000). Jenifer Tidwell describes patterns as providing powerful and generic design guidance in a format that is consistent and easy to read and understand – they convey knowledge about good design (Tidwell 2006).

After reviewing the different approaches that patterns take, we decided not to emphasize the *language* aspect of patterns. Although most pattern designers aim to build up a complete pattern language and do not believe independent patterns can survive without links to others, there is still no real "complete" pattern language. The majority of existing patterns are actually organized in collections or catalogs, or a taxonomy (Henninger 2006). On the other hand, Week introduces two ways to express patterns: an informal structure and a formal structure (Week 2000). Although the informal structure only comprises a problem with its context, proposed solution and illustration of the solution, Week's work in culture driven workplace demonstrates that patterns in such simple form are useful for designers. Tidwell's similarly informed patterns have won recognition from users and other experts (Tidwell 2006). Thus we start with several small but useful patterns.

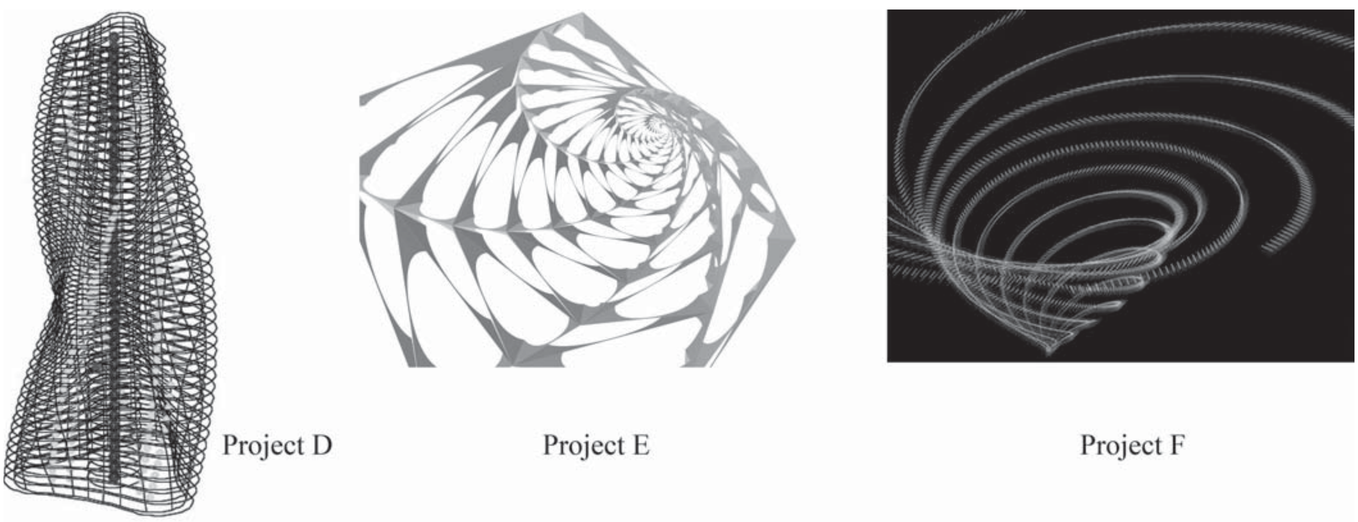

FIGURE 4 Examples of projects which involve pattern *Incremental variation*.

3.2 HOW DO PATTERNS ARISE?

Inspiring as it is, Alexander's notion of pattern opens the door to the otherness that confronts the architect when working with a new context. Patterns and their languages consumed Alexander and his group in years of great effort, but fundamentally conceal serious risks. The solutions for his patterns, mostly found from his or his group's own architectural experience, were abstracted and synthesized into "pattern language" in the 1970s and 1980s. Are these solutions really applicable for other architects in other context (location, time and culture)? The following summarizes how pattern authors in different domains have accounted for the origin of their patterns.

- Workplace Patterns: Week believes that practical knowledge develops through application of general principles to "difficult cases" (Week 2000). In his work, he used "difficult cases" drawn from his own practice. Each case harbors an aporia, where the normal background of practice breaks down. Through gathering data from these projects, he was able to define patterns of workplace design.
- OOP Patterns: In 1995, Gamma and his colleagues published three groups of patterns for object-oriented programming: creational patterns, structural patterns and behavioral patterns (Gamma et al. 1995). In the book, they did not describe how these patterns were generated, but mentioned that these patterns existed in the Model/View/Controller (MVC) triad of classes that is used to build user interfaces in Smalltalk-80. Similar to Week's work, they also used one (maybe several) of their own projects as the pattern sources.
- GUI Patterns: Apart from the book "Designing Interfaces," Tidwell has two websites to publish her UI patterns to support the web application design—a sandbox site and an official site. Tidwell does not introduce how she gathers and confirms the data to design patterns, but we can see new pattern titles emerging firstly in a grey color and then being finalized gradually in the sandbox site. We assume that Tidwell does not have a formal data collection process. Instead, she builds up her patterns by frequent observation of various GUI.
- Web Usability Patterns: Published in 2003, Ian Graham's book introduces the Wu pattern language which aims to improve UI usability (Graham 2003). Patterns introduced in the book are presented from a mixture of software development and business perspectives. In each of the 79 patterns, there is a sentence something like "you meet the problem of ..." There is no hint of whether or not those problems are collected from real life experience or the writer's personal insight.
- Pedagogical Patterns: Bergin et al. introduces some patterns that deal with the diversity of instructional techniques in their recent project report (Bergin et al. 2001). In their website, pedagogical patterns have been grouped into multiples of ten on the basis of course sequence (prior to the course, course as a whole, scale of weeks, scales of days, feedback, dealing with problems etc.). By digging inside their website, we found that some patterns are related to a pattern writing workshop in Vienna and some reference books.
- Communication revolution patterns: From January of 2002, more than eight hundred patterns for communication revolution have been contributed, edited and published in an online collaborative site (Schuler 2002). These patterns' authors are either scholars from universities or members of CPSR (Public Sphere Project). We did not find a complete publication of this ongoing project, but it is clear in the webpages that each pattern has been built upon reference literature or review of real events.

3.1 COLLECT AND DEFINE PATTERN FROM PARTICIPANT OBSERVATION INSTEAD OF CASES

Borchers compares different pattern languages and states that an important goal of any pattern design team is to capture the reasons for design decisions and the experience from past projects in order to create a corporate memory of design knowledge(Borchers 2001). Ideal patterns should be the result of user experience. In our review, most of those patterns may come from authors' own experience or existing cases. However, what are the experiences, how has the experience been captured and what kind of users have been involved? Few authors describe the process. In this study, we use qualitative methods of participant observation to define patterns of parametric modeling. Through the step-by-step implementation of a study, we capture the real experience of how a user is using the parametric modeling application GenerativeComponents.

4 THE PARTICIPANT OBSERVATION STUDY

4.1 CHARACTERISTICS OF THE STUDY

The qualitative study we conducted includes three stages: an interview as the pre-study, two rounds of study in workshops and data triangulation. In the first stage, we interviewed the chief designer of GenerativeComponents Dr. Robert Aish, to obtain his original design ideas of the system, his understanding of user experience, and his assumptions in design patterns. The first study was conducted in the SmartGeometry's January 2007 New

York workshop (Architectural Record 2007). The study has two aims: First, it allows us to gain a preliminary understanding of how experts use and adapt these systems. Second, it will help us to develop the "codable" moments for a complete *thematic analysis* (Boyatzis 1998). The approach we used is *participant observation.* Being the central and defining method of research in cultural anthropology, the method of participant observation includes the user of the information gained form participating and observing through explicit recording and analysis of this information (Anderson 1983). We analyzed the data to discover recurring patterns in a participant's working process. In February 2007, we conducted a similar case study in a workshop in Vancouver (CDRN 2007) and collected data from participants. From January to April 2007, Cheryl Qian was invited to teach GenerativeComponents as a part of one graduate studio course in University of British Columbia. She collected data such as transaction scripts and screenshots from twelve graduate architecture students. The information from these two resources was used as the analyzing source for data triangulation and proof of pervious findings. In the future, we plan to run more such participant observation studies to gather data for design patterns.

4.2 THE RESEARCHER'S ROLE

It is important to recognize that participant observation is a method that combines two somewhat different processes: observation and participation, and this method should be distinguished from both pure observation and pure participation (Bernard 2006). In 1980, a typology to describe a continuum in the "degree of participation" of researchers was developed (Spradley 1980). According to Spradley's continuum, we select the approach of *active participation.* We, as researchers, are engaged in most of the activities that the subjects are doing as a means of trying to learn the rules of their behavior. During the workshop, we engaged as tutors for designers (workshop participants) and simultaneously observed and discussed how they were working. We provided immediate value through tutoring and pose little overhead on any particular designer as the discussion and observation is essentially what already happens in a tutoring session. We became familiar with this software more than three years ago. Having attended GenerativeComponents workshops since 2004 and started to tutor since 2005, we have sufficient knowledge of the application to provide help. We believe this understanding of the context and role enhances our awareness, knowledge and sensitivity to many decisions and issues that would be encountered.

However, due to previous experience, we also bring certain biases to this study. Although we make every effort to ensure objectivity, our biases may shape the way we view and understand the data we collect and the way we interpret our experience. For example, our personal experience and skills might interfere with how we understand a subject's problems and actions. For certain kinds of problems, we might have a set of solutions in mind and thus ignore other alternatives. To avoid such validity issues, while chatting with subjects about their problems, we tried to make our advice suggestive instead of determinative.

4.3 DATA COLLECTION PROCEDURE

- Setting: This study was conducted at the Hudson Hotel in New York City on January 26-29, 2007. The participants (more than 100, mostly architects and civil engineers) were self-selected into five groups. Every group occupied a separate workroom and had three professional tutors. These participants had been competitively selected by the workshop organizers though adjudication of their project proposals. Before this workshop, they all had attended at least a three-day pre-workshop that introduced the basic functions of GenerativeComponents. During main workshop, all the participants worked intensively on their own laptops from morning till late at night.
- Participants (Subjects): In the beginning of the first day, we recruited five subjects through a public process in which they volunteered. Among these five participants, there were three females and two males, all between the ages of 25 and 40. Two of them were graduate students, two were industrial professionals and one was a university educator. Three were architects and two were civil engineers. Three used the workshop to solve design problems in their current commercial projects. Two focused on academic studio designs. All of them started to learn GenerativeComponents the prior year and it was the very first parametric modeling application they had ever learned.
- Events: Apart from three meals and the daily progress report sessions, our participants spent most of their time working at laptops. They sometimes had short chats with their neighbors or negotiated solutions with their tutors. Aiming to have something accomplished by the end of the workshop, they were all quite focused and under pressure. In order not to interrupt their thinking, we interviewed the participants when they were taking a break or during moments of relative calm.
- Process of data collection: With the participants'

permission, we engaged them in short conversations two or three times a day. The conversations were usually 3-10 minutes long. We chatted about the progress in their projects, problems they have met and potential solutions towards those problems. The conversation length depended on the topics emerging during the process. These short conversations were recorded. We also collected various kinds of information such as photos of physical models, digital sketches and hand sketches, screenshots of existing problems, script segments and successful models, and all the feature and transaction files at each stage (Figure 1).

- Data analysis: The data analysis process was aided by a qualitative data analysis application ATLAS. ti 5.0. This software is to help researchers uncover and systematically analyze complex phenomena hidden in text and multimedia data. The program provides tools that let us locate, code, and annotate findings in primary data material, to weigh and evaluate their importance, and to visualize complex relations between them. ATLAS.ti consolidates large volumes of documents and keeps track of all notes, annotations, codes and memos in all fields that require close. We were able to import text data (transcription of important interviews, GenerativeComponents features, and transaction files), images (screenshots or photos taken during the process), and audio data (recordings of short conversations) into ATLAS.ti and create a hermeneutic unit to analyze all the data of one participant's activities as a whole. We adopted the data-driven approach to develop thematic codes based on sampling and design issues (Boyatzis 1998). Figure 2 shows two screenshots of our coding process in ATLAS.ti. The left image is the transcription of one short conversation. In this piece of text, we were able to code information about the subject's understanding of the difference between GC and normal CAD systems. The right image is a rendered image of the model created in GC. Just through this image, we are able to anchor codes such as organized collections of points, penetrating and incremental changes in size and shape, place holders, and radiating directions. Through reading and listening to raw materials again and again, we were able to select subsamples from the project and reduce the raw information. Then we identified some themes as the potential pattern-related ideas within subsamples and compared themes across subsamples. To determine such a code, we focused on comprehending what the problem was, how the participants solve the problem and why she/he selected such a solution. In some cases, the problem was clear but the solutions were uncertain. Sometimes the solution had been presented, but the information about the reason to select such a solution was not explicit. The process of thematic analysis and code development helped us to filter out unnecessary information and get important ideas quickly. The themes and codes created from this unit could be transferred and reused for the analysis of subsequence participants. While analyzing these five participants, some pattern ideas started to emerge.
- Data triangulation: Triangulation is a method used by qualitative researchers to check and establish validity in their studies. There are five types of triangulation: data triangulation, investigator triangulation, theory triangulation, methodological triangulation, and environmental triangulation (Guion 2002). Considering the nature of this study (i.e. the difficulty of changing investigators, theory, or environment), we chose to use the *data triangulation* approach, which uses different sources of data/information. Our data was analyzed triangularly in two levels. Firstly, since we were collecting data from different perspectives (model scripts, interviews and screenshots) and analyzing them as a whole, much of the data was naturally triangulated. For example, in a short conversation we asked one participant how she was able to adjusting the Z direction of points along a Bspline curve based on another reference, both her script and explanation gave us the same answer. Secondly, to triangulate the final findings, we collected data from another six participants in the CDRN February workshop and twelve architecture students in the UBC design studio. In the data analysis, we triangulated by looking for data and patterns that exist across workshop groups.

4.1 ETHICAL CONSIDERATION

We obtained ethical approval before we launched the qualitative study in New York. Participant observation naturally invades the life of the subject and sensitive information such as their roles and progress in the real life project is frequently revealed (Spradley 1980). We ensured that no personal information was collected. Participants were interviewed separately and no information from the interviews was divulged to other participants. Information was collected and saved in a safe place and participants have full access to their own information. In the publications and reports, we only use materials they agreed to share.

5 OUTCOMES AT THIS STAGE

In the interview pre-study, the chief designer Dr. Robert Aish argued that GenerativeComponents aims to engage users with a wide range of skills and to be responsive to support user intentions though breaking down the barriers between the design of the software and the use of the software. Users of GenerativeComponents can not only manipulate the model geometrically but also manipulate the tools of modeling. Existing features can be integrated and new functions can be scripted as tools to support parametric modeling. Recurring phenomena of how users manipulate their tools are the evidence of patterns. Dr. Aish admitted the existence of patterns that users adopted in the application and pointed out that the pattern *Place Holder* has been well established.

We found several rough pattern ideas through analyzing the data of five participants in the New York workshop. After triangulating with the data material collected in the Vancouver workshop, we were able to establish and discuss some patterns with existing examples such as *Branching*, *Incremental variation*, *Filter, Goal seeker, Mapping* and *Sampling.* Here in this paper, we briefly introduce two patterns that could be observed and understood easily from their visual appearance.

- Branching: The intent of *Branching* is to keep a flexible number of branches originating from the same central point. In Figure 3, project A is an aviation museum roof design, project B explores simulated tree branching, project C analyses variations of lacework and project D investigates changes in a tower plan. All of these projects set a variable as the number of branches and keep the flexibility for future alternatives. This variable can be used calculating the rotation-degree of branches, distance between two branches, and the length of branch extension. By adjusting the variable of branch number, the whole model will change significantly and produce more alternatives for designers to explore.
- Incremental variation: The intent of *Incremental variation* is to generate elegant whirling or waving effects by accumulating smooth small variations (such as rotation or resizing) along a path or dimension of the model. In Figure 4, project D is a tower model, in which each floor plan rotates and shrinks while it gets higher. Project E shows an in-progress fabrication photo of a wood fence sculpture made by a landscape artist and project F is a simulation of a whirlwind created by a civil engineer. These three models all chose the same approach of adjusting sub-elements gradually while moving along some dimension(s). Algorithms in D and F link the values of rotation degree, model size and height together. In project E, each beam rotates and changes the length along an arc. To generate the effect of spatial variation, the designer first has to create a basic item with control parameters such as rotation degree, size and length. The next step is to define a path or choose a dimension to grow. We can either divide the path by parametric points (at specific T values) and attach the basic item to each parametric point, or replicate the basic item along the path by steps. With such a parametric structure built up, designers can easily adjust the path and the associated functions (relations between basic item's parameters with the path's T value or between parameters with the number of steps) to explore design variations.

6 FUTURE WORK

A pattern can be presented to its users effectively in both an informal structure and a complete formal structure (Week 2000). We compared different expression forms of patterns examples in landscape architecture, object-oriented programming, web usability, education science and communication domains, and decided to construct our parametric modeling patterns as an adaptation of Jenifer Tidwell's UI pattern expression model. There are mainly six components in a pattern: *diagram, intent* (analogue to Tidwell's *"what"*)*, use when, why, how* and *samples.*

- The *diagram* is a graphic representation of the pattern.
- *Intent* states a one-sentence description of the goal behind the pattern.
- *Use When* describes a scenario consisting of a problem and a context.
- Why states the reasons to use this pattern.
- *How* explains the details of how to adopt the pattern to solve the given problem.
- *Samples* illustrate how the patterns can be used in several different contexts.

Based on this structure, we constructed patterns from the first two rounds of participant observation study. We will attend more GenerativeComponents workshops to discover new patterns. We also want to invite new users and investigators to evaluate the patterns we constructed, perhaps in the form of teaching GenerativeComponents through pattern integration. Their feedback would be essential to polish the final outcomes. Furthermore, we want to apply the patterns on other parametric modeling applications such as SolidWorks and CATIA to test if these ideas are capable of being generalized.

7 CONCLUSION

The literature on design patterns, expertise, intentional stance and user modeling shows us that patterns express design work at a tactical level, above simple editing and below overall conception. Being interested in understanding the mid-level patterns of work that recur across designers and tasks, we aim to provide better learning material and suggestions for improvements in the parametric design applications through discovering those patterns. However, the methods most researchers used to develop their patterns do not satisfy us. We want to capture actual designer experience and understand the reasons for design decisions.

We selected Bentley's GenerativeComponents as the platform and ran participant observation studies in the January 2007 SmartGeometry workshop and February CDRN workshop. In this research, we found that participant observation is not only a useful research method but also an efficient approach to collect and confirm patterns about designers working in a design system. We believe the patterns we discovered from this qualitative study would help in the following fields: In education, the instructor could use patterns to help learners grasp the core ideas of parametric design. In CAD application development, patterns can serve as formative guidelines for developers. More importantly, in the design process, designer intention could be well supported by using patterns and designers could concentrate on their ideas without overt concern about the technical details of using CAD applications and sophisticated mathematical equations.

REFERENCES

Aish, R. 2003. *Bentley's GenerativeComponents: a Design Tool for Exploratory Architecture.* Bentley Systems Inc. http://www.cmu.edu/architecture/graduate/G-CAD/pdf/GenerativeComponents.pdf

Alexander, C., S. Ishikawa, et al. 1977. *A Pattern Language: Towns, Buildings, Construction.* New York: Oxford University Press.

Anderson, J. R. 1983. *The architecture of cognition.* Cambridge, MA: Harvard University Press.

Architectural Record. 2007. SmartGeometry 2007 Winter Conference & Reception. http://www.smartgeometry2007.com/default.asp

Benyon, D., P. Turner, et al. 2005. *Designing interactive systems : people, activities, contexts, technologies.* Harlow, England: Pearson Education.

Bergin, J., J. Ecksten, et al. 2001. Patterns for Gaining Different Perspectives: a part of the pedagogical patterns project pattern language. In *8th Conference on Pattern Languages of Programs, Sept 11-15, Monticello, Illinois.* Illinios: Pace University.

Bernard, H. R. 2006. *Research Methods in Anthropology: Qualitative and Quantitative Approach.* Lanham, MD: AltaMira Press.

Borchers, J. 2001. *A pattern approach to interaction design.* Norwood, MA: Books24x7.com.

Borenstein, N. S. 1991. *Programming as if People Mattered: friendly programs, software engineering, and other noble delusions.* Princeton NJ: Princeton University Press.

Boyatzis, R. E. 1998. *Transforming qualitative information: thematic analysis and code development.* Thousand Oaks, CA: Sage Publications.

CDRN. 2007. Parametric modeling and digital wood fabrication workshop and symposium. from http://www.cdrn.ca/events/wood/general.php

Cross, N. 2004. Expertise in Design: an Overview. *Design Studies* 25(5): 427-441.

Dennett, D. C. 1995. *Darwin's Dangerous Idea: Evolution and the Meanings of Life.* New York: Simon & Schuster.

Duncker, K. 1945. *On Problem Solving.* Psychological Monographs 58(5), No. 270.

Ericsson, K. A. 1999. Expertise. In *The MIT Encyclopedia of the Cognitive Sciences*, ed. R. A. Wilson and F. C. Keil, 298-300. Cambridge, MA: MIT Press.

Fischer, G. 1994. Domain-oriented design environments. *Automated Software Engineering* 1(2): 177-203.

Fischer, G. 2001. User Modeling in Human-Computer Interaction. *User Modeling and User-Adapted Interaction* 11(1): 65-86.

Gamma, E., R. Helm, et al. 1995. *Design patterns : elements of reusable object-oriented software.* Reading, MA: Addison-Wesley.

Graham, I. 2003. *A pattern language for web usability.* Boston: Addison-Wesley.

Guion, L. A. 2002. Triangulation: Establishing the Validity of Qualitative Studies. Gainesville, FL: University of Florida, Cooperative Extension Service. http://edis.ifas.ufl.edu/pdffiles/FY/FY39400.pdf

Henderson, A. and M. Kyng 1991. There's no place like home: Continuing design in use. In *Design at Work: Cooperative Design of Computer Systems*, eds. J. Greenbaum and M. Kyng, 219-240. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Henninger, S. 2006. Disseminating Usability Design Knowledge through Ontology-Based Pattern Languages. In *3rd International Semantic Web User Interaction Workshop (SWUI). Atlanta, GA, Nov. 6th 2006.* http://swui.semanticweb.org/ swui06/papers/Henninger/Henninger.pdf

Norman, D. 2005. Human-Centered Design Considered Harmful. *Interactions* 12(4):14-19.

Schuler, D. 2002. Liberating Voices! A Pattern Language for Communication Revolution. http://diac.cpsr.org/cgi-bin/diac02/pattern.cgi/public

Spradley, J. P. 1980. *Participant Observation.* New York: Holt, Rinehart and Winston.

Tidwell, J. 2006. *Designing interfaces.* Beijing: Sebastpol O'Reilly.

Week, D. 2000. *A hermeneutic approach to the practice of architecture in a foreign culture.* PhD. diss., University of Sydney.

Wickens, C. D., J. Lee, et al. 2004. *An introduction to human factors engineering.* Upper Saddle River, NJ: Pearson Prentice Hall.