DESIGN PATTERNS: AUGMENTING DESIGN PRACTICE IN PARAMETRIC CAD SYSTEMS

by

Zhenyu Cheryl Qian

B.Arch., Southeast University, 1999

M.A.Sc., Simon Fraser University, 2004

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the School
of
Interactive Arts and Technology

SIMON FRASER UNIVERSITY
Summer 2009

APPROVAL

Name: Zhenyu Cheryl Qian

Degree: Doctor of Philosophy

Title of dissertation: Design Patterns: Augmenting Design Practice in Parametric CAD Systems

Examining Committee:

Chair: ______________________

Dr. Marek Hatala, Graduate Chair

____

Dr. Robert F. Woodbury, Senior Supervisor
School of Interactive Arts and Technology

____

Dr. Lyn R. Bartram, Supervisor
School of Interactive Arts and Technology

____

Dr. Alissa N. Antle, Supervisor
School of Interactive Arts and Technology

____

Dr. Halil I. Erhan, Internal Examiner
School of Interactive Arts and Technology

____

Dr. David Week, External Examiner
Chair and Consultant, Assaí, Sydney, Australia

Date Defended: JUL 29 2009

SFU SIMON FRASER UNIVERSITY
LIBRARY

Declaration of Partial Copyright Licence

SIMON FRASER UNIVERSITY
THINKING OF THE WORLD

STATEMENT OF ETHICS APPROVAL

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

(a) Human research ethics approval from the Simon Fraser University Office of Research Ethics,

or

(b) Advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University;

or has conducted the research

(c) as a co-investigator, in a research project approved in advance,

or

(d) as a member of a course approved in advance for minimal risk human research, by the Office of Research Ethics.

A copy of the approval letter has been filed at the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Bennett Library
Simon Fraser University
Burnaby, BC, Canada

Abstract

Parametric Computer Aided Design systems facilitate complex design activities through generating, representing, and storing parametric relations and variations in design. The systems' increasing complexity impedes the ability of designers to explore and develop their work. Pattern, as a rhetorical structure, is a generic solution to a well-described problem. I hypothesize that a designer's learning and working process in parametric modeling applications can be well modeled by patterns, and making such patterns explicit can result in improved expert work practices, better learning material and support for collaborative design in parametric design environments. To achieve these goals, I designed and conducted a series of qualitative studies to search, gather, elicit, author, and communicate patterns in the context of a parametric modeling system - Bentley's GenerativeComponents$^{\mathrm{TM}}$. Applying a cognitive framework of design activities, the evaluation studies show evidence of how parametric design patterns can augment and support design practice.

Keywords

Design Patterns, Parametric CAD System, Cognitive Design Framework, Participant Observation, Augment Design Practice

For my husband, Yingjie Victor Chen, who offered me unconditional love and support throughout the course of this dissertation.

"Art is the imposing of a pattern on experience, and our aesthetic enjoyment is recognition of the pattern."

— *Alfred North Whitehead,* DIALOGUES, *1954*

Acknowledgments

I feel very fortunate to have had the guidance and support of my committee throughout this engaging PhD journey. My deepest gratitude goes to my senior supervisor, Dr. Robert Woodbury, for his excellent direction and invaluable support in the production of this work. I owe much of my success to him. I would also like to thank my other supervisors, Dr. Lyn Bartram and Dr. Alissa Antle, who provided invaluable feedback and thoughtful commentary throughout the research process. Their critical insights helped to strengthen my work as well as positively affirm my approach to the research.

I would like to extend my gratitude to Dr. Halil Erhan and Dr. David Week, my thesis examiners, for a challenging and thought-provoking oral examination.

I thank my collaborators in the research and peers in the SIAT graduate program who offered their support, advice, and friendship during these past five years.

I am grateful to the fifty-one participants in my study who shared their opinions and practices despite the pressures and time constraints of the intensive design workshops and studios. They generously allowed me to use their multimedia data in the work anonymously. Without their voluntary and enthusiastic participation in the research, the project would not have been possible. A sincere thank you goes to the people from Bentley's Generative-Components development team for their kindly support during the studies.

My final appreciation goes to my family. Their love and support made the difficult times bearable and the good times better.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Premise

From a rational perspective, design activity can be essentially understood as a process intended to solve problems. Or, in John S. Gero's [86] words, "a goal-oriented, constrained, decision-making exploration, and learning activity." To design is to understand a context and propose (often novel) solutions that satisfice constraints and balance relationships across domains. Design work is increasing in complexity in contemporary industry. Contemporary design usually involves the interaction of many designers working on different elements. The context for my work is thus the classical task of design, conditioned by current trends of increasing complexity and collaboration.

Computer aided design (CAD) software is a family of computing systems intended to facilitate complex design activities. Parametric modeling is one of the current industrial forces in CAD systems, particularly in architecture and building engineering. Although such tools have been used in other areas for many years, their recent uptake in the building domain is yielding many new features for CAD systems and new insights for architecture designs.

One of the most significant dimensions of parametric modeling software is in representing variation in design. This works best when variation is continuous and distinctly different alternatives are not part of the model. Paradoxically, designers are not taking full advantage of these tools. Really bright people design and develop parametric design systems. Some even brighter people use them. But people can use them in really dumb ways. One part of the reason is contingency - we often use these tools to build things quickly in design and throw away code when finished the immediate task. We sacrifice clarity for speed. Another part of the reason is that we do not know how to use these tools effectively and elegantly and the tool developers do not know how to convey such use. Furthermore, tool developers may not be sure how to write systems that are easy to learn and use. Designers do not know how to explain their design behaviors to the developers. There is a midground of rhetoric, explanation, structure, and thinking that we are not adequately covering as a parametric design community. My dissertation aims to augment this midground.

1.2 The Motivation

At the outset of this doctoral study, my initial research questions were: "How do designers' intentions play out in action?"; "Is it possible for a designer to discern intention from the record of his or her design actions?"; and, "Can we create and improve tools to support or activate their intentions?" Refining and generalizing these questions led me to my PhD research and its studies.

Holding a Bachelor degree in Architecture and a Master degree in Interactive Arts, I am specifically interested in the cognitive and human-computer interaction (HCI) issues of design in terms of supporting designers' learning, collaboration, and ability to solve increasingly complex problems. Since 2003, we (my research group at SFU) had the opportunity to work closely with professional designers using one parametric modeling system – Bentley's GenerativeComponents$^{\mathrm{TM}}$. We see that most designers using parametric modeling are amateur programmers. Their preferred style of programming is to find, copy, and modify. Through observation of their design practice and related conversations, we became aware that similar design and coding strategies were used repeatedly. People keep on reinventing the wheel and doing the same thing again and again.

My research proposes using patterns to support designers (amateur programmers) in using the complex parametric design tool. A pattern is a generic solution to a well-described problem. Its description includes both problem and solution, as well as other contextual information. Being a clear expression of a practical idea, it is a great device for discussion and sharing. I use patterns as constructs to collect, organize, and communicate regularities in designer use of parametric design tools. My strategy is to use patterns to help designers focus away from the toolbox and onto their design work. Thus they can record and reuse effective design work and build a community of practice around parametric design.

1.3 The Questions

This research tries to observe, understand, support, and augment users' design practice in parametric design systems. The users I focus on comprise active designers - more specifically, architects and civil engineers. My objective is to understand the mid-level patterns of work that recur across designers and tasks. The research questions are:

- Can a designer's learning and working process in parametric modeling applications be well modeled by patterns?
- Can making such patterns explicit result in improved expert work practices, in better learning material and support collaborative design in parametric design interfaces?

The first question seeks to examine if pattern is the correct form to capture and encapsulate parametric design practice, while the second seeks to explore usability issues and implications of parametric design patterns for use in practice. To answer these questions, I designed and conducted a series of qualitative studies on designers' using one parametric CAD system - Bentley's GenerativeComponents^{TM} (GC). GC is one of the most advanced parametric modeling tools in the current commercial Architecture, Engineering and Construction (AEC) market. This system promotes and educates parametric modeling both in industry and academia, reaching design firms for commercial projects as well as architecture schools for design studios. Through hosting workshops around the world, the community of GC users evaluates and improves GC to make it more supportive for design and engineering. People come to attend these workshops to solve real design problems and share experience. Thus GC provides a great platform for me to conduct this parametric pattern research.

1.4 The Structure

Catherine Marshall outlines four key questions underlying the research process [135]:

1. What are we trying to do?
2. How are we trying to do it?
3. How well are we doing?
4. How can we do it differently/better?

The answers to the first question establish the purpose and goals of the project. After introducing the research questions in Chapter One, Chapter Two introduces the social justification of this research with a particular emphasis on cognitive design perspectives. Chapters Three and Four provide a conceptual justification for the work through review of existing literature on parametric design systems and design patterns.

The second key question turns goals into specific plans. I use five chapters to answer this question. First, I introduce my research methodology from framework to strategies in Chapter Five. In my research plan, there are three stages: (1) a pilot study to search for first evidence of pattern-like phenomena (Chapter Six answers "why it is important?"), (2) a series of participant observation studies to gather and elicit evidence for pattern-like thinking (Chapter Seven investigates "is it a real problem?"), and (3) the development of the repository of parametric design patterns. The repository's story has two parallel plots. Chapter Eight describes the authoring process of thirteen patterns and Chapter Nine illustrates the repository system development to support pattern communication.

The third question demands an evaluation of how the research is working to achieve its purpose and goals, through reflection and self-awareness, and through researching those whom you are trying to impact. Chapter Ten employs two participant observation studies to investigate whether or not parametric patterns are constructed and interpreted cognitively and whether or not they are supportive for the design practice. We wish the patterns discovered from design practice would help in following aspects. First, in education, where the instructor may use patterns to help learners to grasp the core ideas of parametric design. Secondly, in CAD application development, where patterns can serve as formative guidelines for developers. Last, but most important to the design process, designers may use patterns to concentrate on their ideas without overt concern about the technical details of CAD applications and sophisticated mathematical equations.

Informed by the research findings, the fourth question requires a reevaluation of research purpose and goals. Instead of introducing renewed planning and actions, Chapter Eleven presents the knowns and unknowns of my research domain and proposes future directions.

In this PhD research, I designed and conducted seven qualitative studies. To make the reading easier, I use Table 1.1 to summarize the purpose, event details, and research methods of all these studies.

Research Stage	**Event**	**Time/Location**	**Duration**	**Population /Size**	**Number of Subjects**	**Research Methods**
Search for pattern-like phenomena	SmartGeometry 2007	Jan/New York		19 expert users	6	formal interview
Collect and elicit pattern ideas	SmartGeometry 2007	Jan/New York	2+4 days	100+	5	participant observation, contextual inquiry
	CDRN Workshop 2007	Feb/Vancouver	10 days	25	6	
	UBC Design Studio 2007	Jan-Apr/Vancouver	12 afternoons	12	12	
	ACADIA 2007	Oct/Halifax	2 days	15	6	
Evaluate patterns	SmartGeometry 2008	Feb-Mar/Munich	2+4 days	154	10	participant observation, contextual inquiry & sketch journaling study
	ACADIA 2008	Nov/Minneapolis	3 days	20	6	

Table 1.1: List of Research Studies

Chapter 2

Understand Design Process and Design Activity

Design is important. It underlines many aspects of modern work. From spoons to aircrafts, design affects each human being's daily life and drives cultures. This PhD research proposes to support designers through searching, collecting, modeling, and communicating regularities in design practice of parametric design tools. Before diving inside specific concepts, I want to first outline the social justification for the research.

This chapter aims to build up the literature foundation of this research to understand design process and design activity. Starting from investigations of the conceptual meaning of design, I reviewed three groups of cognitive design frameworks from the perspectives of context, representation and work analysis. Such a review produced a list of important cognitive issues I should observe during my empirical studies.

Design is cognitively a complicated process. In the latter part of chapter, I choose to pay close attention to three aspects of the design activity - learning, expertise, and collaboration - which I use to observe and evaluate my study. I reviewed the current literatures and projects around these concepts. In Chapter Ten, I will discuss whether or not my work of parametric design patterns can support designer and design in these three directions.

2.1 Understand Design Process

2.1.1 Definitions to Illustrate Complexity of Design

Design is complex. Archer wrote that [17]:

> "Design is that area of human experience, skill and knowledge which is concerned with man's ability to mould his environment to suit his material and spiritual needs."

In this PhD work, I choose to understand design as a rational, logical, and sequential process intended to solve problems. In this conception, the notion of "design process" is synonymous with "problem-solving process", which in all but its most abstract forms works by consultation and consensus. The process of design begins with the identification and analysis of a problem or need, and proceeds through a structured sequence in which the problem is examined, ideas explored and evaluated until an optimized solution is devised.

Yet, design has also been conceived in other ways:

- "a science of understanding people's *needs* and their unique *relationship* with art, literature, history, music, work, philosophy, community, technology and psychology. The act of design is *structuring* and *creating* that balance." – Clement Mok (digital visual design)
- "a Renaissance attitude that combines *technology, cognitive science, human need* and *beauty* to produce something." – Paola Antonelli (Museum of Modern Art)
- "a fundamental soul of a man-made *creation* that ends up expressing itself in successive outer layers of the product or service." – Steve Jobs (Apple Inc.)
- "a *conscious* effort to impose a meaningful *order*." – Victor Papanek (design philosopher)
- "depends largely on *constraints*." – Charles Eames (furniture design)
- "Everything is design. *Everything*!" – Paul Rand (graphic design)

Defining design is challenging. To define the meaning of design, there are broad definitions of design and specific ones - both have drawbacks. Either they are too general to be meaningful or they exclude too much. However, different designers do provide various perspectives. Through highlighting and filtering the keywords in those quotations, the meaning of design becomes richer than simply the process of problem solving and production. It also involves understanding the context, fulfill needs and constraints, and balancing the relationships across domains.

Kim Vicente [236] listed eleven interrelated characteristics to subsume the different types of *complexity* we can find in designing socio-technical systems: large problem space (composed of many different elements and forces), social (composed of many people who must work together to make the system function properly), heterogeneous perspectives (composed of expertise from different background and represent the potentially conflicting values of a diverse set of disciplines), distributed (located in different places), dynamic (changes over long time constants), hazard (potential hazard in systems), coupling (many subsystems that are highly interacting), automation, uncertainty, mediated interaction (the goal-relevant properties cannot be directly observed), and disturbances (dealing with unanticipate events) [236, p.14-17]. Although this list was specified for designing socio-technical systems, I think it also implies the high degree of complexity found in many other design domains.

Good design begins with the needs of the user. No design, no matter how beautiful and ingenious, will be successful if it doesn't fulfill the user's needs. While this may sound obvious, many products and services have failed because of a failure to understand this. Finding out what the user wants is the first thing a designer should do. The designer then builds on the results of that inquiry with a mixture of creativity and production insight. Ideas seem strange may worth exploring and the "common-sense" solution may not be the right one. Designers often hit on counter-intuitive concepts through sketching, prototyping, brainstorming, or user testing. Watching users in real-world situations can provide insights into their behavior, leading to novel ideas that wouldn't otherwise have been encountered, had the designer simply thought about the situation.

2.1.2 Cognitive Frameworks to Understand Design Process

To understand design, we should first understand humans (no matter from the prospective of designers, users, tools, or the outcome). Cognitive science provides such an entrance. Different cognitive frameworks have been built up to interpret the cognitive process of understanding users and generating creative ideas. Cognitive science tries to understand the nature of human kind by drawing on multiple empirical disciplines, including psychology, philosophy, neuroscience, linguistics, anthropology, computer science, sociology, and biology to investigate the nature of intelligence. I construct a cognitive framework to understand the cognitive aspects of design process from three perspectives: context, representation, and work process analysis.

2.1.2.1 Study Context

As Brooks [36] suggests, taking *context* seriously means "finding oneself in the thick of the complexities of particular situations, at particular times and with particular individuals." To maximally benefit designers, HCI literature should provide (1) a broad background of comparative understanding over cross domains, (2) high-level analysis useful for evaluating the impact of major design decision, and (3) information that suggests actual designs rather than simply general design guidelines or metrics for evaluation. From these literature, I choose three approaches, *activity theory*, *situated action models*, and *distributed cognition*, to understand the context aspect of design process.

Activity Theory proposes that the "activity itself is the context." Leont'ev describes an activity as being composed of subject, object, actions, and operations [125]. Actions are conscious processes because they are goal-directed (one holds a goal in mind), and different actions may be undertaken to meet the same goal. Actions are similar to *tasks*, as defined in the HCI literature [164]. The dynamic aspect of activity theory is that the constituents of activity are not fixed but can change as conditions change. Thus, we should not consider the context as an outer container or shell inside of which people behave. Instead, it is people who consciously and deliberately generate contexts (activities) in part through their own objects.

Situated Action Models emphasize "the emergent, contingent nature of human activity, and the way activity grows directly out of the particularities of a given situation [218, 219]". In this theory, the unit of analysis is the relation between the individual and the environment. Situated action emphasizes responsiveness to the environment and the improvisatory nature of human activity [122]. Suchman emphasized that, "a central tenet of the situated action approach is that the structuring of activity is not something that precedes it but can only grow directly out of the immediacy of the situation [219]". As a result, instead of studying of more durable, stable phenomena that persist across situations, this theory provides a useful corrective to restrictive notions that put research into something of a 'cognitive straitjacket'. In this theory, rigid mental representations such as formulaic plans or 'rational problem solving' cannot account for real human activity.

Distributed Cognition focuses on the "propagation of knowledge between different individuals and artifacts, and the transformations that external structures undergo when operated on by individuals and artifacts [105]". Although the cognitive system is akin to an Activity Theorists's activity, distributed cognition "moves the unit of analysis to the system and find its center of gravity in the functioning of the system" [160]. This theory is concerned with structure-representations inside and outside the mind - and the transformation these structures undergo. Compare with traditional cognitive science [162] that focuses on individual's cognition, main interest of distributed cognition is cooperating people and artifacts. According to Andy Clark, human cognition is characterized by the dynamic feedback loops that span brain, body, and world [54].

All above three frameworks for analyzing context are valuable in that they underscore the need to look at real activity in real situations, and that they squarely face the conflux of multifaceted, shifting, intertwined processes that comprise human thought and behavior. However, they also differ from each other in terms of the *structure of activity* and the *persistence of structure* [160]. Situated action perspective lets us take careful notice of what people are actually doing in the flux of real activity, while distributed cognition leads us to understand useful of design by combining the formal and cognitive properties of artifacts with observations on how artifacts are used.

Considering the analysis unit, a distributed cognition analysis begins with the positing of a 'system goal'. System goal is an object, but also is "an abstract systemic concept that does not involve individual consciousness [160]". In activity theory and distributed cognition, an object is (partially) determinative of activity. But in the situated action model, every activity is by definition uniquely constituted by the confluence of "situation" formed by particular factors. One activity cannot be distinguished from another by reference to an object (motive) [122].

In the study of context, there is an important facet, *persistent structures* (such as artifacts, institutions, and culture), playing in shaping activity. Activity theory focuses on the historical development of activity and mediating role of artifacts. In the literature of distributed cognition, there exists a similar notion. Hutchins introduces *collaborative manipulation* as “the process by which we take advantage of artifacts designed by others, sharing ideas across time and space [105]”. Nardi states clearly that “persistent structures are a central focus for both activity theory and distributed cognition [160]”. However, situated action models can not well accommodate durable structures that across different activities and persist over time. In some recent articles the discussion of “routine practices” and “routine competencies” started to emerge [219] to account for the observed regularities in the work settings studied. The studies continue to report minute particulars for detailed episodic events, and weave in descriptions of higher level behaviors as well.

The empirical research methods adopted by these three approaches also differ. Situated action analysis rely on “logs“ recorded through videotypes or other recordings of recordable, observable behavior[220] . Accounts from study participants describe in their own words what they think are doing. Activity theory, on the other hand, focuses on the value of interview data. HCI community has widely accepted that people cannot articulate what they are doing. Primarily at the level of operations, it is very difficult to describe how you type or how you know. But this generalization does not apply to the higher conscious levels of actions and objects [159]. In a skillful interview or teaching process, operations are often brought to the subject’s conscious awareness. At least to some degree, operations can be talked about and described. I find a major difference between activity theory and situated action. In the former, the structuring of activity is determined in part by human intentionality before the unfolding in a particular situation. In situated action, activity can be known only as it plays out in situ.

Although all these three frameworks has their own merit, Nardi [160] argues that the activity theory seems to be the richest framework for studies of context in “its comprehensiveness and engagement with difficult issues of consciousness, intentionality, and history.” Distributed cognition, as a supplement, also helps to understand design ideas sharing across time and space.

2.1.2.2 Study Representation

As the primary currency of design activity, representations can be categorized as either verbal-conceptual and visual [6], or internal and external [163]. Within cognitive science, there has been a move towards analyzing the interaction between internal and external representations in design. Vera and Simon stress that [233, p.12], “A fundamental problem for cognitive modelers is to interleave internal and external states in order to achieve naturalistic behavior." For two decades, Norman [163, 165, 166] has been describing cognition in terms of "knowledge in the head" and "knowledge in the world". In 1987, Larkin and Simon explored the difference of textual and diagrammatic information in term of computational requirements and affordance. Comparison of Diagrams and informationally equivalent text passages are framed in three perspectives: search (accessing information), recognition (matching information to knowledge in long-term memory), and inference (creating new knowledge). They pointed out that diagrams typically allow dramatically more efficient recognition than do equivalent text stimuli [121]. This computational model of diagram use is primarily focused on internal representations. Later on, external representations started to draw research attentions. Larkin shifted her thinking and started to consider the role played by external displays in cognitive problem solving [120]. Hutchins [104] discusses how various cultural artifacts that are integral to work practices have evolved such that they reduce the cognitive processing required by the operator to translate one form of external representation into another.

Emerging from this trend, *external representations* have received a more central functional role in relation to internal cognitive mechanisms. It potentially allows researchers to account more adequately for how graphical representations affect the design process. Scaife and Rogers proposed a new agenda for research on how different kinds of graphical representation are cognitively processed [197].

External cognition is a branch of cognitive science that focuses predominately on the role played by external representation in human cognition, and on the relationship between internal and external representations. Scaife pointed out that there are distinctions between *language-based external representations*, such as books, and *graphic forms of external representation*, such as diagrams and maps [198].

Research has been conducted on interactive multimedia (IMM) technologies, virtual worlds with autonomous agents, and mixed reality environments according to some key conceptual design issues such as explicitness and visibility (perceptual interferences), cognitive tracing and interactivity (feedback providing), ease of production (possibility for reconstruction), combining external representations (graphical and textual), and distributed graphical representations (collaborative potential).

Researchers focus on the cognitive process of external representation interactions, cognitive benefits of different representations, and the properties of internal and external structures. To extend the practicability of external cognition research, Scaife and Rogers aim at "a conceptual framework that can lead to a better understanding of the basis of design and use new technologies to support learning [198]". What they describe as an analysis of "cognitive interactivity" is that, "on the way technologies allow new forms of representation not possible with existing media, and how these new forms may be exploited for learning. They propose an emphasis on identification of the different kinds of cognitive benefits that particular representational formats and technologies provide [197]".

2.1.2.3 Study Process Analysis

Cognitive Work Analysis (CWA) is a formative, constraint-based framework for analyzing complex socio-technical systems such as the design process. This research first adopted the framework for Cognitive Systems Engineering [185], and continues to contribute to the development of analyzing framework [236]. CWA is fundamentally different from conventional task-based analysis methods. Instead of describing what workers should do (normative) or what they presently do (descriptive), CWA is formative in that it describes what workers could do to achieve work purposes. Unlike normative approaches, CWA recognizes the essential adaptive role that humans play in safely controlling the unanticipated in complex socio-technical systems. Unlike descriptive approaches, CWA aids in distinguishing between workers' essential *goal-directed actions* and *work-around activities* required to cope with existing work support systems. This branch of literatures focus on describing "worker" behaviors and the detailed "working" process. However, since design is also a form of active working, I adopt this framework to understand and analyze the design process.

To systematically explore design possibilities, CWA focuses on identifying the constraints that shape designers' goal-directed behavior rather than cataloguing specific designer tasks and procedures. This approach is inspired by ecological psychology's tenet that complex human behavior in operational settings can be usefully explained by analyzing the complexity of their surroundings. These characteristics make CWA useful for complex socio-technical systems where traditional task-based analysis produce voluminous and uninterpretable data, or do not systematically address risks posted by unanticipated events.

Rasmussen's framework for CWA provides a taxonomy of nested constraints that guide designer behavior in complex socio-technical systems [185].

Work Domain Physical or intentional constraints determine what can be purposefully accomplished within the socio-technical system. Designers cannot perform operations that violate laws of nature or exceed the capabilities of the process and equipment.

Control Tasks Within the limits of the work domain, execution of generic control tasks are constrained by the order of operations, and the system states in which they are necessary or effective.

Strategies The same control task can typically be performed in many ways using different cognitive strategies. Experts shift between strategies in response to changing demands, for example time, information accessibility, and cognitive resources.

Social Organization and Cooperation Depending on the cognitive strategy, control tasks can be assigned. An organization's social structure determines how information and decision-making capabilities relating to the work environment are distributed.

Worker/Designer Competencies The above constraints determine the physical, perceptual, and cognitive demands placed on humans in the system. Designers must be capable of meeting these demands through skill, rule, or knowledge based behavior. Analysis of worker competencies can guide system design, training programs, or development of support tools.

Applications of CWA can be diverse. The Australian Department of Defense, for example, has applied CWA to a wide range of projects including specifying, designing, and using training systems; evaluating alternative equipment design proposals; and developing team designs [158]. CWA has been used extensively by Cognitive Engineering Laboratory (CEL) to design interfaces, experiments, and training, to analyze data, and to categorize performance measures and knowledge elicitation techniques [235]. It has proven to be an invaluable research and design tool in these instances.

2.1.3 Conclusion of Cognitive Design Frameworks

My review of cognitive design frameworks seeks to anchor sound design mechanics. I do so by focusing on design decisions related to key design elements: context, representation, and process analysis.

- *Context*: Research identifies how system design benefits from the explicit study of the context in which users work [36]. The model user has to be connected with a social group and supporting artifacts. But with this realization about the importance of context comes many difficult questions [160]. For example, what exactly is context? What is the correct unit of analysis if the individual is no longer central? What are the relations between artifacts, individuals, and the belonging social groups? I compare three approaches to the study of context: activity theory, situated action models, and distributed cognition. Through consideration and comparison of the key concepts each approach promulgates, I study their research methods and search for the usefulness of each for the design of technology. That analysis suggests that the activity theory is suitable for understanding comprehensive context issues around the design process, and distributed cognition is useful to investigate the social context it focuses on the transformation between individuals and artifacts.

- *Representation*: Research on external cognition is initiated by a move of analyzing the interaction between internal and external representations in design, specially focusing on instructional design [197]. Their framework of key conceptual design issues and examples of case studies are useful references as evaluation criteria for my research.

- *Process Analysis*: Jens Rasmussen distinguished three models that can be adopted for the specific purpose of grouping work analysis techniques: *normative models* that prescribe how a system should behave, *descriptive models* that describe how a system actually behaves in practice, and *formative models* that specify the requirements that must be satisfied so that a system can behave in a new desired way [184]. Research in CWA outlines the constraints that guide the researcher adopt the analysis framework to investigate details of a complex work/design process. As can be seen through the sample case study [236], conducting a full work analysis is exhausting. However, I believe a close look at the work domain, task type, strategies used, social organization, and designer competence is a quite sufficient analysis on a design process study.

My interpretation on these three cognitive design frameworks will serve as the criteria to evaluate my research outcomes and discuss their cognitive aspects.

2.2 Understand Design Activity

Design as a process can take many forms depending on the object being designed and the individual or individuals participating. Apart from the cognitive design frameworks discussed above, there are several conditions that complicate design activity: tool learning, expertise building, and collaboration.

2.2.1 Cognitive Learning Tools

One of the primary goals of this research is to support professional designers to learn how to use design tools. Cognitive learning tools, defined here as instruments that support or perform cognitive processes for learners in order to support learning, both in the traditional supportive instructional environments and modern open searching environments. I also try to make this PhD research as a kind of cognitive learning tool to support tool learning.

There is a Chinese saying that goes, "sharpen your tool before tackling a task". Learning the tool becomes necessary before design work begins. In order to achieve the objective of scale production and professionalism, gaining experience with the tools is essential. There are many forms of learning process, ranging from formal classes to informal self-instructional. Different learners are suitable for different learning styles. However, necessary support is always needed to help the designer to learn their tool in an efficient way. One example could be using multimedia learning. Learning is no longer only through textbooks and classrooms. Multimedia has the potential to use several perceptual channels, it could take better advantage of human capacities for processing information [197]. Richard Mayer [140] describes multimedia learning as "based on the idea that instructional messages should be designed in light of how the human mind works." Thus, multimedia learning is potentially a more effective and efficient learning form. Beyond multimedia learning, learning could be supported through several dimensions as following.

2.2.1.1 Support Reflective Learning

Donald Norman [165] distinguishes between two forms of thinking - *experiential* and *reflective*. Experiential thinking evolves from one's experiences with the world. Reflective thinking, on the other hand, requires deliberation and external support such as books, computers or other people. The process is like, encounter a situation, think about it, reflect on stored knowledge, make inferences about it, and determine implications and reasons for it.

Reflective thought is a careful, deliberate mode of thinking that helps us make sense of what we experience and what we know. Norman [165] contends that computers may be able to support reflective thinking, "when they enable users to compose new knowledge by adding new representations, modifying old ones and comparing the two." Along a similar vein of thought are other conceptions such as Donald Schon's "reflective practitioner" [201] and the more general notion of meta-cognitive thinking. Rather than simply taking teacher's word, learners must reflect on their own knowledge, and decide what they need to know. In brief, as a cognitive tool, it should be a reflective use of technology because using it aims to engage learners in reflective thinking, which purportedly leads to knowledge construction.

2.2.1.2 Notion of Scaffolding Tool

Recent design research on interactive learning environments has incorporated the notion of *scaffolding* [188]. Traditionally, the term scaffolding was used to describe the process by which a teacher or more knowledgeable peer assists a learner, altering the learning task so that the learner can solve problems or accomplish tasks that would otherwise be out of reach [247]. Scaffolding entails a delicate negotiation between providing support and continuing to engage learners actively in the learning process. Lepper et al. [126] describe it as maintaining an "optimum" level of challenge for learners. In his vision, scaffolding refers to ways the software tool itself can support learners rather than only teachers or peers. The tool itself changes the task in some way so that learners can accomplish tasks that would otherwise be out of their reach. Moreover, the software should provide "additional assistance beyond what a simpler tool would have provided to allow learners to accomplish more ambitious tasks [126]". There have been a number of research projects on designing software scaffolding tools and have draw several design principles. Key ideas underlying these principles are some common assumptions about how to make a more productive knowledge construction experience for learners, such as enabling learners to inspect multiple views of the same object and providing visual conceptual organizers to give access to functionality [183].

Software scaffolding provides support to help make the learning more tractable for learners. For example, KIE (Knowledge Integration Environment) provides prompts to encourage or remind students what steps to take [66]. In some cases, graphs also help to represent the learning experience and make the information easily tractable. For instance, graphical organizers help students plan and organize their problem solving strategies [182]. Applications of the concept of scaffolding are not been limited to learning settings. They can be transplanted to other environments requiring tools to advise and support knowledge construction and idea creation. My research domain, parametric design systems, is such an environment.

2.2.2 Expertise in Scaling Up to Complexity

Supporting design cognitively also includes recognizing and supporting designers' expertise in design.

2.2.2.1 What is Design Expertise?

Expertise consists of those characteristics, skills and knowledge of a system, that distinguish experts from novices and less experienced people. Broadly, there are two approaches to the understanding and study of expertise [108]. One approach is to understand expertise as an emergent property of communities of practice. expertise is jointly constructed by tools for thinking and scripts for action within social groups. In the second approach, expertise is a characteristic of individuals. It is a consequence of the human capacity for extensive adaptation to physical and social environments. Many accounts emphasize that expertise is developed through long periods of deliberate practice [78]. Not only have larger amount and more complex knowledge, experts are believed to be able to store and access information in larger cognitive "chunks" than novices can, and to recognize underlying principles, rather than focusing on the surface features of problems.

An early experimental investigation was done to understand cognitive differences between experts and novices [50]. The experiment showed that experts represented a given problem differently from novices. Experts represented the problem according to its underlying principle, whereas novices represented on the problem's literal features. This finding of deep versus shallow knowledge representation has paved the way for many later studies of expertise in particular domains.

Where other studies of design expertise that examine expert versus novice performance, expert designer behavior and outstanding designers, Cross focuses specifically on expert performance [62]. He claims that expertise in design is distinct from other fields. Although designers change goals and constraints as they design, they appear to retain their principal design solution for as long as possible, even when detailed development of the scheme reveals unexpected difficulties and shortcomings. Creative design solutions arise when "there is a conflict to be resolved between the designer's own high-level problem goals and the criteria for an acceptable solution, as established by the client or other requirements" [62].

From a variety of software systems, Nardi and Miller [161] identified a continuum of three kinds of users: end users, local developers, and professional programmers. End users focus more on their own domain, have little or no programming education and tend to have less interest in computers. Local developers know the local domain and have advanced knowledge of computing. In the collaboration, they serve as a resource for end users, training them and developing code for them. Programmers go through professional training, have much broader and deeper knowledge of computing than local developers. Programmers contribute to end users and local developers, help them learn new tools and functions.

Knowing how to use a design tool (even at guru level) does not necessarily make a professional designer. On top of support tool learning, I also hope my design outcomes can encourage the communication among end user, local developers, and programmers, and support professional designers' expertise to archive their design target more efficiently.

2.2.2.2 Issue of Scaling Up

This PhD research is concerned with understanding expertise in design, and *scaling up* designer problem solving abilities to greater complexity in practice.

The challenge of scaling up is in how to expand applicability of the solution (or strategy) of a small problem to larger problems without sacrificing the goals of logical expression and understanding. From a size standpoint in computer programming, scaling up refers to the programmer's ability to apply strategies learned from small programs in larger programs [38].

It is hard for a designer to program like a computer scientist. Being able to scale up solutions conceptually is also a challenge. Computer scientists invest years to obtain skills such as procedural decomposition and recombination, abstraction and encapsulation, iterative code refinement, and code performance verification across expected inputs [57]. Designers also have the skills of divide-and-conquer and abstraction. But these skills translate poorly to design because expertise is domain-dependent. Procedural abstraction, structured programming and proper variable organization are hard for designers. Even as accomplished amateur programmers, designers have limited knowledge of established data structures and algorithms and prefer a copy-and-modify strategy of work. Being able to scale up is an essential expertise we expect the designers to build up during the design process. Modern days designers make their design tasks much more complicated than before. My research outcomes, parametric design patterns, are capsules of strategies with samples of small problems. It is possible to copy and modify the strategies solve more complex problems. I want to understand patterns' utility in assisting designers to increase the complexity at which they can effectively work.

2.2.3 Collaborative Design

Design is an inherently creative collaborative process. Almost all complex artifacts (buildings, airplanes, computer and software etc.) are designed through the interaction of many participants working on different elements of the design. Collaborative design is typically for large, expensive and time-consuming works. Design decisions are strongly connected and depends upon each other. It is difficult to converge on a single design that satisfies these dependencies and is acceptable to all participants. Recent research from the cognitive science and complex system negotiation literatures has much to offer for understanding of principles of motivation [80] and dynamics of this collaborative design process [113].

Architectural design and urban planning involves highly cooperative tasks. Individual phases within a project often change between close cooperative situations, for example, during design and review meetings, and individual work carried out by the participants or third parties. During the design and review meetings problems are discussed and solutions or alternatives are proposed. There is a commonly held view in Computer Supported Collaborative Work (CSCW) that design collaboration occurs largely at the project outset: "the actual preparation of particular solutions is once more performed by the individuals (leaving the final decision to one of the following meetings). Real collaboration is often limited to the creation of early paper-based sketches [35]". Design professionals would argue that this view is far from true. Collaboration happens in face-to-face discussion, phone calls, emails, exchange of media and code. Meetings are a mere surface. There is a urgent need to foster clarity and communication in out-of-meeting exchanges. From an architect's point of view, it would be desirable to have additional support to introduce the design ideas, explore alternatives, improve the cooperation, and accelerate design and review cycles.

There is another collaborative design culture that designers with different levels of expertise and interests can work together to improve their work desigaided design (CAD) system users to find out how they use the sophisticated customization and extension facilities offered by the software. They found that users of varying levels of expertise collaborate to customize and improve their CAD environments. Within a group of users, there is at least one local expert, called *local developer*, who provides support for other users. According to in-depth interviews around the collaborative activities, Nardi and Miller found that other users sincerely replied on such local developer's contributions.

Many research projects (e.g.[35]) propose complex and expensive tools or systems, such as augmented environments, to support collaborative design practice. I want to argue that larger effects are to be found in the asynchronous exchange of well-considered design representations. Parametric modeling is transforming practice, both within and between firms. But it is evident that this transformative technology imposes new needs for a higher level of support for its communities of practice. I conjecture that design patterns are a good device around which to structure such support.

Chapter 3

Parametric Design Systems

Parametric modeling software has introduced computational mechanisms and interfaces for representing variation into design. Unlike the CAD domain, parametric systems have been used in other areas for many years. By using parametric modeling in CAD systems, it is possible to develop models that generate, record, and represent design alternatives. However, it is very difficult to understand the range of possibilities entailed by such models. I examine the impacts of parametric modeling in *physically based modeling* and *spreadsheet modeling* in order to understand their effects on the domains in which they are applied. The literature review on other parametric design systems provides insight into the domain and aids in identifying areas in need of additional research and development.

3.1 What is Parametric Design?

Parametric design is an approach to product modeling that "associates engineering knowledge with geometry and topology in the product model by means of constraints [15]". A parametric system is based on a hierarchy of variable-controlled dependencies. Each active variable causes the overall system to change its behavior and thereby generate variations without losing the overall coherence and integrity of the system. Sussman identifies parametric design systems by two fundamental concepts [222]: a *parameter* that acts as a variable to which other variables are related, and these other variables can be obtained by means of parametric equations; and, a *constraint* that specifies a relation that must be maintained.

Every parametric design system is built upon a constraint solver. There are several methods to implement such systems; these methods are used to distinguish types of systems.

When the flow of computation is fixed by the procedural representation of the constraints, the model is unidirectionally parametric or called "procedural" [148]. Procedural systems are relatively straightforward to implement, but can only express a modest range of variations. It is quite difficult to add new types of dependencies between the fixed sequence of variable assignments.

More advanced methods are parametric and variational methods. Parametric and variational models have been developed to encourage reuse of existing models to support innovative design [206]. The constraint satisfaction algorithm employed distinguishes this two types of system. Theoretically, "parametric systems solve constraints by applying sequential assignments to model variables, where each assigned value is computed as a function of the previously assigned values. Variational systems solve constraints by constructing a system of equations representing the constraints and solving all constraints of the systems simultaneously based on a numerical equation-solving procedure or equivalent method [206, p.82-83]". Some literature distinguish such parametric modeling systems from the user's perspective: *propagation system* (compute from knowns to unknowns with a data flow model) and *constraint system* (solve sets of continuous and discrete constraints) [250].

Evaluation methods are also different for parametric and variational methods. parametric models are based on explicit sequential constraint satisfaction, the evaluation is quick, but they cannot deal with general mutually coupled constraints. Variational models using implicit constraint satisfaction techniques. They can deal with coupled constraints, but they are slower and limited in the ability to handle incompletely specified models, and have difficulty in detecting inconsistent models.

The line between parametric and variational models is blurred. Currently many system employ a hybrid of both types. Researchers have looked at hybrid techniques to integrate both methods. One approach is to extend constraint propagation methods by a relaxation technique for handing coupled cyclic constraints. This method was pioneered by the Sketchpad system [223] and ThingLab laboratory [31]. Another method tries to decouple constraint equations in subsets that can be processed sequentially. Two frequently cited examples are the algebraic solution [168] and the rule-oriented method [234]. Shah and Mantyla argue that the terms parametric and variational, "have been used almost interchangeably in technical and particular commercial context" [206, p.80].

During the past two decades, significant progress has been made on parametric design systems. In 1995, Shapiro and Vossler [207] complained that, "the modern parametric system may not robustly support parametric and variational modeling because the meaning of a 'parametric family' is not well-defined". During recent years, several sophisticated parametric modeling applications representing variations in design, including Generative-Components [4], Revit [189], and CATIA [45], have been introduced to the commercial CAD market.

3.2 Families of Parametric Design Systems

Parametric models can be applied in different design systems. By varying the inputs, the model can produce different kinds of outcomes. Based on the broad sense definition, we see parametric design systems developed in different domains: computer graphics (physically based modeling), business management (spreadsheet modeling), architecture design, industrial design, electronic engineering VLSI system, manufacture engineering, etc. A scan of the research literature reveals the vastness of the parametric modeling domain and the fragmentary situation of its literature. In many respects, there are few relations and connections built up across domains.

Before entering a more detailed discussion and analysis, I outline recent publications on parametric design systems in related domains.

- **Industrial Design**. The parametric modeling theory and systems exist in industrial design domain. Chu et al. presents a parametric design system for 3D tire model production [53]. This work has been integrated into a CAD/CAM system for actual model production. The surface model of tire grooves is parameterized designed, allowing for rapid creation of other grooves with simple design tables. Bezirtzis et al. presents the Interactive evolutionary design (IED) system. It has the potential to change development processes from generating individual solutions to designing parametric models that are employed to create a vast space of possible solutions [28].

- **Civil Engineering**. The civil engineering discipline has long used parametric modeling software such as Constructive Solid Geometry (CSG) [15, 157], and SolidWorks [212].

- **Electronic Engineering**. Parametric modeling has an irreplaceable position in technologies of constructing and simulating VLSI system circuit models [48]. Parametric models are closed-form equations that can be implemented in circuit simulation environments either via interfaces to external code or as traditional sub circuits.

- **Manufacturing Engineering**. A recent publication presents a parametric modeling software for the design of turbo-machinery blades [115]. It provides the ability to interactively construct 3D turbo-machinery parametric blades of various types.

3.3 Media Effects of Parametric Design Systems

Apart from the general features of parametric design systems, I am also interested in investigating their media effects. Considering the limited population of current parametric CAD systems, I reviewed the other two parametric design families, *physically based modeling* and *spreadsheet modeling*, using McLuhan's tetrad of media effects.

3.3.1 Analysis Framework: Tetrad of Media Effects

Published posthumously by his son Eric McLuhan, "Laws of Media" [143] brought together Marshall McLuhan's ideas as a concise tetrad of media effects. These four laws demonstrate McLuhan's abiding concern with pedagogy in that they are meant to be used as tools to analyze the patterns of effects that different technologies produce. McLuhan phrased them as four questions with which to consider any artefact [143, p.98-99]:

- "What does it extend, enhance, accelerate, intensify or enable?
- When pushed beyond the limit of its potential, it will reverse. What were its original characteristics? Into what does it reverse?
- What does it displace or obsolesce, that is, render relatively without dominant power or influence?
- What does it retrieve from the past that had been formerly obsolesced?"

McLuhan arranged these four probes in oppositional form as a tetrad of quadrants. The specific arrangement – extension, reversal, obsolescence, and retrieval, beginning at the upper left corner and moving clockwise – illustrates relationships between pairs of aspects: the upper pair relative to the lower pair, the left pair relative to the right (Figure 3.1).

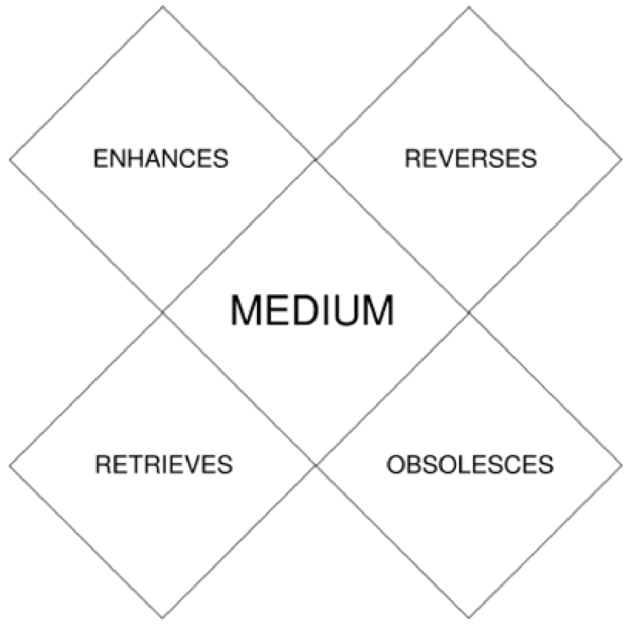

Figure 3.1: Traditional Diagram of McLuhan's Tetrad of Media Effects.

Variations on this diagram characterize the tetrad structure as four laws clustering or circulating around the center of media. It is important to remember that these four laws exist simultaneously. There is no chronological order among them although sometimes they are talked as succeeding each other.

The tetrad is not meant to be a scientific tool, but one for exploration of the "grammar and syntax" of each artefact, a dynamic tool to describe "situations that are in process" [143, p.116]. In general, the tetrad helps us understand both the *thing itself*, be it a particular technology, business process, model or concept; and society's collective response to it. Some researchers have adopted McLuhan's tetrad framework to analyze the effects of media topics, such as linear axiomatic text [141] and geographic information systems (GIS) [221]. In this research, I employ the framework to analyze parametric design systems for physically based modeling and spreadsheet modeling.

My interpretation of this tetrad is that McLuhan's laws of media construct a two-dimensional coordinate system where each law occupies one quarter of the space (Figure 3.2) (rotate 45 degree). This system evaluates media effects according to the coordinates of

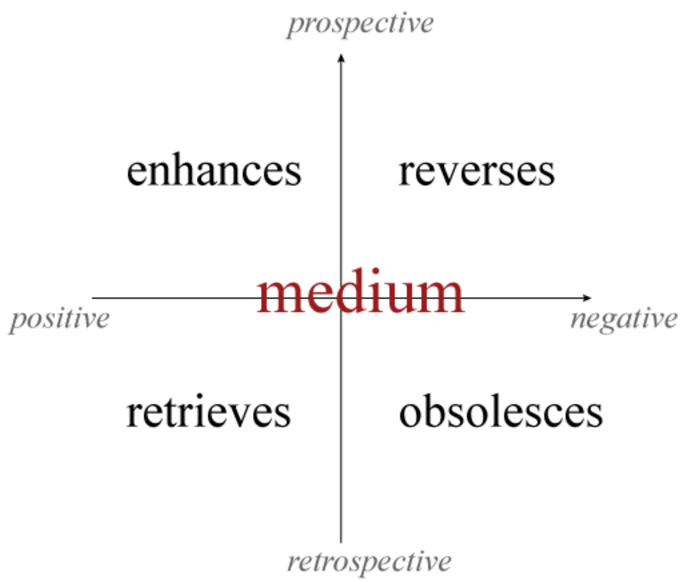

Figure 3.2: A Coordinate System to Represent McLuhan's Tetrad.

negative-positive and retrospective-prospective. As an analysis tool, this structure significantly decreases the number of differentiating criteria from four to two, limiting analysis to issues of influence type and time.

3.3.2 Physically Based Modeling

Computing is often applied to the modeling or simulation of physical systems. A branch of computer graphics called physically-based modeling aims to 'incorporate physical properties into computer models in order to allow numerical simulation of their behavior [23]". Common themes in physically-based modeling include the simulation of rigid or flexible bodies, motion, inter-body interactions, and constraint-based control. For example, how a cloth drapes over an object is determined by surface friction, fabric weave, and the internal stresses and strains generated by forces in or on the objects. In addition, sophisticated rendering techniques can be tied in with physically based models, enabling precise simulations of the interaction of light and matter. These models are founded on mathematical equations and often involve numerically intensive computation to simulate their behavior.

Many constraint-based physically based models were developed in the 1980s, including energy constraint on parameterized face models [173], body models [242] and constraint-based models for human skeletons [20], in which connectivity of bones and limits of angular motion on joints were specified. A number of other prominent research directions exist in the domain, including cloth simulation [240], deformable solids [175], and stream-erosion of terrain models [111]. Free-form deformable models, first introduced to computer graphics in 1987 by Demetri Terzopoulos, were particularly relevant in the context of modeling non-uniform rational B-Splines (NURBS)[227]. Since the behavior of a deformable model is governed by physical laws and characterized by imposed constraints, this new approach provided a means to overcome NURBS's drawbacks of tedious indirect shape manipulation and manual adjustment. Concluding works before 1990, Foley et al. [82] stated that physically based modeling was able to determine the shape of objects, but simulating the motions still had quite a few challenges.

Current domains utilizing physical based modeling include the following:

Human Simulation Derived from the studies of parameterized human body models in 1980s, more and more specifically focused topics emerged in the past two decades, such as musculoskeletal simulation, realism motion animation, facial simulation, and surgical simulation.

- **Musculoskeletal Simulation -** Biomechanics researchers developed mathematical models of many different musculoskeletal structures. These models exam muscle forces and explain how the geometric relationships among the muscles and bones transform these forces into moments about the joints. A software package called Software for Interactive Musculoskeletal Modeling (SIMM) was developed in 1995 to enable users to develop, alter, and evaluate models of many different musculoskeletal structures without programming [70]. In 2003, solid mechanics models were created to fully account for the 3D deformation of muscle by viewing it as visco-elastic solid [226]. The associated contact can be solved by using a Finite Element Method (FEM). However, these models are difficult to create and expensive to simulate. Pai et al. [170] indicated that "the fibre -like properties of muscles are not directly represented in the FEM discretization, but must be bolted on using auxiliary function." Between these two extremes, a new model that provides both fast simulation and physically realistic 3D solid behavior was introduced in 2005 [170]. This new approach designs a computational primitive (Musculoskeletal Strand) that fits the inherent properties and constraints of muscles and tendons in a natural way. Adopting line-based muscle models and the realism of physically based deformable solids, this model can simulate the close contact between muscles and bones in forearm, as well as complex tendon paths in the fingers.

- **Realism Motion Animation -** An ambitious goal in the area of physically-based computer animation is the creation of virtual actors that autonomously synthesize realistic human. There was a considerable history of progress in animating virtual humans during early 90s [21, 100]. Until the end of 1990s, although researchers offered the enormous variety of controlled motions that humans may perform, little effort had been directed at how the resulting control solutions may be integrated to yield composite controllers with significantly broader functionalities. In 2001, Faloutsos and his colleagues proposed a simple but effective framework for composing specialist controllers into more general and capable control systems for dynamic characters. An important technical contribution within their controller composition framework is an explicit model of pre-conditions, which characterize those regions of the dynamic figure's state space within which an individual controller is able to successfully carry out its mission [79].

  Physically-based methods rely on "intensive computation that scales poorly to a large optimization problem [79]" . In contrast, data-driven approaches take advantage of large motion databases by composing segments of example motions. This approach can create long sequences that achieve coarse-grained goals, but only if appropriate motion segments exist in the database. Clearly, data-driven methods cannot record all possible interactions among multiple characters in advance. Liu et al. describe techniques for adapting a small collection of motions to create complex new motions with multiple characters in 2006 [132]. For the benefits of users, Chai and Hodgins conducted their research in another direction. They present a technique for generating animation from a variety of user-defined constraints [46]. This system in 2007 demonstrates the effectiveness of this approach by generating whole-body and facial motion from a variety of spatial-temporal constraints.

- **Facial Simulation -** Computer facial animation can be categorized into 2D based approaches and 3D based approaches. While 2D approaches preserve the original illumination conditions and color information of human skin, most of them suffer from the problem of limited facial expression and being difficult to be used when the expression from a different viewpoint is required. The main problem of 3D approach in the old time was the low quality of texture data and inappropriate illumination. Terzopoulos and Waters incorporated a physically based approximation to facial tissue and a set of anatomically motivated facial muscle actuators to develop a new generation of 3D hierarchical human face in 1990 [229]. The model was efficient enough to produce facial animation at interactive rates on a high-end graphics workstation. Then in 1994, Terzopoulos and Qin developed Dynamic NURBS (D-NURBS), a modeler that can interactively sculpt curves and surfaces and design complex shapes to "required specifications" [228]. Some recent works [30, 109] explored photo-realistic 3D human skin illumination model through considering the oily surface, skin bump and the subsurface scattering.

  A specific topic of facial animation termed performance driven animation, started from the work of Williams [243]. In 3D case, this method uses blending or free form techniques to perform the shape reconstruction. Physically based simulation is more complex to estimate because of the dynamic behavior. Sifankis et al. in 2005 gave a good example about how the physical control parameters can be computed from the motion capture data [209]. Chen and Prakash's 2006 paper [47] presents a detailed performance analysis for a facial expression synthesizer.

- **Real-time Haptic Interaction and Surgical Simulation -** In order to to provide users with a sense of navigating three-dimensional worlds in a compelling manner, stereoscopic display technologies and motion tracking devices are now widely integrated in a variety of simulators. The next set of challenges in increasing the realism of the user experience involves more general haptic interactions with dynamically modifiable 3D models. Physically based Finite Element Models (FEM, also used in musculoskeletal simulation) are an essential ingredient for building such systems. In 2004, Botsch and Kobbelt introduce an editing system allowing the user to "specify intuitively a region of influence and corresponding boundary conditions, and deforms the boundary surface by minimizing an approximate shell strain energy [32]". On the other hand, a large number of surgical simulators have been presented for a whole host of medical procedures such as endoscopy, suturing, and microsurgery. One recent work by Lindblad et al. [130] demonstrates a physically-based framework for real-time interaction with 3D solid models discretized by FEM. The model formulation allows fast progressive updates, can be used in modeling the addition of new elements as well as dynamic inter- and intraelement changes in model connectivity.

Garment and Cloth Simulation In 3D garment simulation, there is always a contradiction between efficient (e.g. real-time animation) and high fidelity (e.g. deformations of cloth). In 2003, Chittaro and Corvaglia concluded that the high computational complexity of the simulation makes it very difficult to achieve both goals: existing systems are tailored to favor one of the two [51]. A few seconds of high fidelity garment animation might require hours of computation. The results maybe satisfaction for the movie industry, it still presents issues (e.g. integration with existing textile design tools) from the aspect of the fashion designer.

Another bottleneck in cloth simulation is collisions. Since all points are on the surface, all points may potentially collide with each other in any given time step. Moore and Wilhelms proposed that repulsion forces are useful for contact whereas exact impulse-based treatment is useful for high velocity impact. As a result, "authors have toyed with using both" [151]. For example, Sims's project switched between instantaneous impulses for high velocities and penalty spring forces for low velocities to treat his evolving articulated rigid body creatures [210]. Bridson et al.'s 2005 work presents an algorithm to process collision, contact, and friction in cloth simulation efficiently and robustly. They use both techniques of repulsion forces and exact impulse-based treatment in a fully hybridized and efficient manner [34]. They also show how their simulation data can be post-processed with a collision-aware subdivision scheme to produce smooth and interference free data for rendering.

Hair Modeling Unlike solids or fluids, which have been studied for over a century and well modeled by now classical equations, hair remains a largely unsolved problem described by no well-accepted model. The form and characters of human hair were started to be considered and modeled since 1988 [131]. The past couple of years has seen a renaissance in hair modeling, rendering and animation. Rubin's 2000 book presents a unified hierarchical formulation of theories for three-dimensional continua, two-dimensional shells, one-dimensional rods, and zero-dimensional points [192]. The discrete approximation of the strand model comes strikingly close to the strand-as-serial-multi-body-chain model, first proposed by Hadap and Magnenat-Thakmann [98]. Since then the paradigm is successfully used for hair simulation in 2005 [52]. Many researchers have accepted this approach of modeling strand as serial chain of rigid segments connected by spherical joints [97].

"Absorbed" Topic: Solid Deformation Solid deformation was one of the key topics in physically based modeling during the 1980s. After two decades, this topic have been rapidly developed and significantly enriched. However, it also somehow disappeared. This domain had become the fundamental part of nearly all the other topics. When we talk about the muscles and tendons in musculoskeletal simulation, interactions between characters in realism motion animation, cutting skin and organs in surgical simulation, fibre in cloth simulation, and hair strand in hair animation, these are all deformable solid models. This exists everywhere in the domain of physically based modeling. Here is one of the few independent works I could found on this topic. Melek and Keyser [144] in 2007 present a free form deformation (FFD) based method for approximating large-scale deformations due to smaller-scale physical simulations.

Earth System Sciences Continuing the 1988's stream-erosion terrain models [111], physically based modeling also have been widely adopted in the domain of earth system sciences. Recent works include Jurgen Schmidt's book which helps to implement state-of-the-art soil erosion prediction technologies within soil and water conservation planning and assessment through soil erosion physically based models [200] and Giertz et al.'s work of using physical liquid model to analyze hydrological processes in a tropical headwater catchment [87].

Synthetic Music Physical modeling has become a big buzzword in the music industry these years. Julius O. Smith III from Stanford University organized and published the theory around this topic in his new online book [211]. In the physical modeling synthesis technology, the user does not create sound directly. Instead, s/he creates and controls a process that produces the sound. The process models the actual instrument. By specifying some parameters, the user plays the synthetic instruments to make beautify music. With physical modeling technique, the parameters involved are much easy to understand because they have a real counterpart in physical instruments.

Physically Based Nature Sound Other sounds created in the nature can also be reproduced through physical modeling. In 2005, Van den Doel [232] develops a physically based liquid sound synthesis methodology. The fundamental mechanism is the acoustic emission of bubbles. A stochastic model for the real-time interactive synthesis of complex liquid sounds such as produced by streams, pouring water, rivers, rain, and breaking waves is based on the synthesis of single bubble sounds.

3.3.2.1 Tetrad Effects of Physically Based Modeling

Above I briefly studied the evaluations of several physically based modeling design systems, as well how such technology has changed or improved on people's work. Grouping and summarizing them under different topics is the first level of abstraction. Seeking for ideas that are more explicit, Table 3.1 lists and compares the commonalities of main contributes across different literature and different topics. Significant common features have been marked with color boxes in the table. After these two levels of abstraction, we can see some obvious similarities beginning to emerge. In the following, we examine physically based modeling using McLuhan's tetrad:

Prospective - Positive What does it enhance?

- **Realism in computer graphics**: Realism is one of the most common goal in computer graphics [82]. Most of the recent achievements, such as close contact between muscles and bones [170], complex motions with multiple characters [132], various subtle facial expressions [47] and vivid collision and friction in cloth simulation [34], are all clear evidences of this progress.
- **Collaboration across different simulation methods and domains**: Researches have been conducted to integrate achievements in different areas to produce complicated. For example, Faloutsos and his colleagues [79] developed an effective framework for composing specialist controllers into more general and capable control systems for dynamic characters. On the other hand, surgical simulation is a new branch that built up at least four existing domains: free-from deformation, geometry-based cutting, musculoskeletal simulation, and simulation-based animation [130]. More ambitious research topics will emerge when their required knowledge is available.

Musculoskeletal Simulation	Realism Motion Animation	Facial Simulation	Surgical Simulation	Garment & Cloth Simulation	Hair Modelling	Other Topics
1990 hip, limb model moment-generating capacities of muscles 1995 SIMM develop models without programming, but 5 no 3d shape, curved muscle paths & contact 2003 3D elastic solid difficult & expensive 2 Method of FEM 6 2005 solve a balance between 2 extremes fast simulation & 3D realistic behavior close contact of bone 2 line-based model --> complex tendon path 4	1993 inverse kinematics. coordinate transformation to calculate parameters automatically 1 1995 synthesize complex dynamics of running, diving. 3 2001 a framework to compose specialist controllers into a system. A conflict between quality and sequence length 2 2006 adapt a small collection of motions to create complex interaction motion 3 between characters 2007 automatically learn from motion capture data and then enforce as a motion 4	1972 start to model 2D,3D human faces 1990 use motivated facial muscle actuators to develop 3D hierarchical face 1990 performance driven animation 1 1994 Modeller Dynamic NURBS to interactively sculpt curves & surfaces 2003 photo-realistic 3D human skill illumination model 2005 good example about how physical control parameters be computed from motion capture data 4 2006 parametric volumetric muscle structure with interveave function, synthesize various facial expressions in real-time animations 1	Ready stereoscopic display, motion tracking device, hardware & FEM 6 1999 fast updates to deformable solids 1 2002 schemes for handling nonlinearities 2004 deform boundary surface by minimizing a shell strain energy 2005 use continuum models to support cutting, remeshing continuously as the interacting process 2007 integrate previous experience, 3 a model formulation allows for fast 1 progressive updates & dynamic changes in model connectivity	A balance between efficiency & high fidelity 2 1988 use both repulsion forces & impulse-based treatment 1994 work closely with industry switch 5 between instantaneous impulse for high velocities & penalty spring forces low velocities 1999 FEM model 6 to produce impressive animation of fracture 2005 an algorithm to process collision, contact & friction in cloth efficiently and robustly. use repul sion forces & impulse based treatment , provide interference free data	1988 start to consider model human hair 2000 a unified hierarchical formulation of theories 3 2001 the discrete approximation of the strand model 2002 an elaborate continuum theory 1 behind the dynamics of thin deformable objects 2003 paradigm is successfully used for hair simulation typically modelled using mass-spring-hinge system, as individual hair stands. Not effecive in consistent coordinates and twist dynamics 2007 modeling strand as serial chain of rigid segments connected by spherical joints.	Solid Deformation no longer an independent topic. become foundations of other topics Earth System Sciences 1988 stream erosion terrain models 2000 soil srosion models 2002 headwater catchment models Sound Made by Liquid 2005 a physically based liquid sound synthesis methodology Music Industry 2007 physical modelling intruments

1	continuous & fast update	2	balance the conflict	3	unify & integrate previous progress
4	from capture data to simulation	5	serve for real users	6	use the FEM method

Table 3.1: Strategy Analysis of Physically Based Modeling Literature.

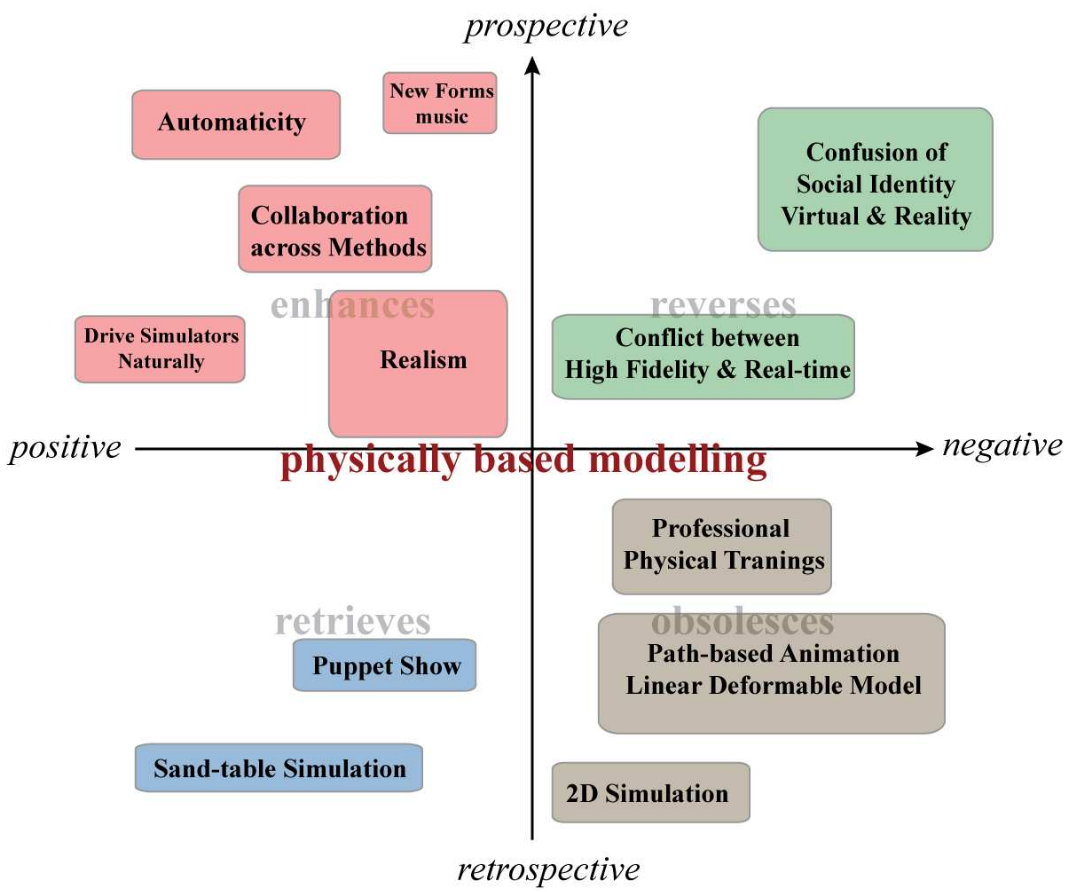

Figure 3.3: Tetrad Effects of Physically Based Modelling.

- **Automaticity of making physical rules**: Trying to simulate the reality, researchers input physical rules of the physical world they analyzed as constraints to the simulation systems. Now, the recent works are able to automatically learn from motion capture data and recreate the motion through the digital model [209, 46]. Reality can affect the digital model more directly.
- **New forms of music composition**: real time synthesis of music instrument can create much real and vivid music effects [211]. The user is able to create his/her own orchestra, focusing more on the music composition for better music and sound.
- **Operate simulators more naturally**: In the early 1990s, researchers noticed users might have difficulty with programming to operate simulators (car, flight etc.), and discussed about how to work with real users and decrease the programming component. Such discussions started to disappear in recent publications. Instead, Chai and Hodgins [46] argues that the approach of letting the system automatically learn a statistically dynamic model is naturally for the benefits of users.

Prospective - Negative What does it flip into when pushed to extremes?

- **Conflict between high fidelity and continuous real-time updates**: Both high fidelity and continuous real-time updates are key criteria to achieve realism. Balancing both is one of the hottest topics in recent publications [47, 130, 169]. However, the cost of satisfying the both needs extremes the computational power. Due to the limitation of human's perception system, we can have a reasonable limit on real-time update, e.g. visually a refresh rate of 60 per second can generate smooth animation. There is a saying that "the flapping of a butterfly wings can create a storm and change the direction of a storm." It is almost impossible to re-duplicate all physical laws (and calculate the effect) on an object to create the extreme high fidelity. It is also unnecessary to do so. At a certain stage, a limit on how many physical properties and rules should be applied onto the model should be set up.
- **Confusion of social identity, virtual and reality**: Online games and social cyberspaces in general, are often touted as liberating environments where one is

free to explore facets of their personality repressed by the constraints of "real life". Researchers have pointed out that the immersed experience with multi-player games reveals that "the confusion between online selves and player can be problematic for the player's real life"[75].

Retrospective - Negative What does it make obsolete?

- **2D Simulation**: Parker in 1972 categorized computer facial animation into 2D based and 3D based approaches [173]. In 1990, Terzopoulos and Waters [229] still describes that 2D approaches preserve the original illumination conditions and color information of human skin. Currently 3D simulation is the dominant method in simulation and animation industry.
- **Path-based animation and linear deformable model**: Software like 3Ds-MAX and Macromedia Flash provides path based animation tools. Hodgins's 1995 work in physically based modeling demonstrates the success in implementing a slew of specialist controllers capable of realistically synthesizing the complex dynamics of running, diving, and various gymnastic manoeuvres [100]. Muller et al.'s 2002 paper [155] illustrates schemes for handling nonlinearities in solid deformation. For example, to simulate a free falling object, if the system calculates the gravity, initial speed, initial direction, and air resistance, the path of the object can be easily calculated. Simple forms of animation tools have been replaced from the gaming industry.
- **Professional training experiments**: Recent achievements on surgical simulation, such as remeshing continuously as the cutting proceeds during interaction [251], make the distance surgery and visual autopsy training possible [237]. In the near future, some traditional physical training experiments such as autopsy would be supported or partly replaced by new physically based Virtual Reality tools.

Retrospective - Positive What does it retrieve?

- **Puppet show**: In the traditional puppet shows, a puppet undergoes a process of transformation through being animated, and is normally manipulated by one puppeteer who hides behind the curtains. Some puppets can be moved electronically. Some advanced puppets can demonstrate detailed expressions with subtle gestures manipulated by more than one puppeteer (2004 movie “Strings”). The gestures a puppet could make are limited by constraints (how the puppet’s limbs were constructed). During the beginning stage of motion simulation research, how designers adjust the motion parameters just likes how the puppeteer to adjust the positions of limbs to simulate the real human’s gesture.

- **Sand table**: Sand tables, such as Greek ancient tool abax [106], were popular on simulating landscape, construction (reservoir and dam), military planning, etc. With the help of physically based modeling technique, simulations can be much closer to the real scenario. This helps a lot on decision making (unlimited times undo) and training (no fear for failure).

In Figure 3.3, I used my proposed coordinate system to organize and represent the tetrad effects of physically based modeling. Size and location of boxes somehow imply the strength of effect and time of these issues.

3.3.3 Spreadsheet Modeling

Accountants have used forms of spreadsheets for hundreds of years. The term “spread sheet” (now “spreadsheet”) also has a long tradition. Reference to its non-computerized version can be found in Eric L. Kohler’s book “A Dictionary for Accountants” [114] and refers to a worksheet providing a two way analysis of accounting data. Richard Mattessich pioneered computerized spreadsheets for business accounting [137]. His later publications [138] contain print-out illustrations and the computer program (later written in FORTRAN IV by his two research assistants, Tom C. Schneider and Paul A. Zitlau). This contribution has variously been recognized in the accounting and related literature.

We have used the term *spreadsheet* to refer to spreadsheet modeling environments. The earliest spreadsheet, VisiCalc, had limited functionality but introduced the world to the concept of the interactive electronic spreadsheet. Later spreadsheets such as Lotus 1-2-3, Microsoft Excel and Quattro Pro greatly expanded the features and developed the spreadsheet into “an effective modeling, prototyping, analysis and presentation tool [81]”. Since their introduction in the late 1970s, spreadsheet programs have transformed the notion of end-user computing, creating a new computational paradigm that offers ease of use and unprecedented modeling power [107].

A spreadsheet is a parametric model. The user builds relationships by entering formulas into cells. A simple example, $b10 = sum(a1 : a9)$ defines a relation: a value in cell b10 is the sum of cells a1 to a9. Whenever one cell (or several cells) among a1 to a9 change, b10's value will be automatically updated. Viewed as model generators, spreadsheet programs have both pros and cons. On the one hand, today's spreadsheet programs are extremely powerful, versatile and user friendly when we consider their humble origins and limited scope. On the other hand, Isakowitz et al. [107] note that spreadsheets suffer from several limitations which typically go unnoticed by novice users: implicit logic, inaccessible model structure, data dependency, and the lack of a unifying model base.

Aiming to teach students in business, economics, and engineering to use spreadsheet modeling efficiently, Monahan focuses on three themes – translation, construction, and interpretation [150]:

- "Complex problems in all of the functional areas of management are *translated* into decision models. Emphasis is on the art of analytical model building.
- *Construction* is the process of building spreadsheet models in applications from the decision models. Skills developed here center on the effective use of the many powerful features of applications.
- *Interpretation* is the conversation of solutions generated in applications into managerially relevant terms that are understandable to those familiar with neither analytical tool nor spreadsheet models."

There are similarities between this structure and physically based modeling's five components (goals, conceptual model, mathematical model, posed problems, and implementation notes) [23]. Both translate posed problems into conceptual models. Knowns and unknowns should be listed in the analysis process. Then, developers should adopt mathematical models to construct parametric models in the systems. Their interpretation is documented in implementation notes. Being two disparate domains, their strategies of translation and construction are different. In spreadsheet modeling, an example problem could be, "formulating a linear program that determines how to meet the auditing demands of three clients each week at minimum cost" [150, p. 124]. Monahan's book suggests a transportation model – how to "ship" products from their source to some other locations at which they are demanded – to solve this problem. There are two sets of constraints: to guarantee that demand is satisfied at all destination points, and to guarantee that no more units are shipped than are available at any point. Through the use of analogous constraints, the problem can be solved.

In spite of the increasing sophistication and power of commercial spreadsheet packages, Grossman [91] argues that this domain still lacks a formal theory or methodology to support the construction and maintenance of spreadsheet models.

Here, I group the literature of spreadsheet modeling into four sections.

Engineering: Most spreadsheet users (modelers) are not information system professionals. Existing software engineering principles have been adapted significantly to form a spreadsheet engineering discipline more sympathetic to spreadsheet modelers [40, 91]. Colver advocates that adopting one approach to spreadsheet development often has negative side effects on the positive aspects of spreadsheet technology such as flexibility and speed of development [117]. Grossman and Ozluk [92] extend previous work on spreadsheet engineering principles, to give a more traditional adaptation of the SDLC and moves away from a best practice approach. This fresh approach gives consideration to the actual use of the final spreadsheet and recommends incorporating users into the development and holding a review of use after implementation.

End-user programming: For many spreadsheet users, the spreadsheet is the programming language. This is especially true for the "end-user" programmers, they are not employed as programmers, but write programs for their own use[159]. A research data from 2000 shows, the number of end-user programmers in the U.S. alone is expected to reach 55 million by 2005, as compared to only 2.75 million professional programmers at that time. [29]. These facts suggest that the spreadsheet is also a particularly significant target for the broader application of programming-language design principles.

Excel today provides user-defined functions by allowing the user to write a function in Visual Basic. Many programming environments that were intended for end-users provide ease of use, but do not support programmers without some kind of "trapdoor" mechanism leading to a different programming language than the easy-to-use top layer. It is not surprising that the end users shy away from software engineering approaches. And that there is a long list of well document spreadsheet-error horror stories, e.g. at the web-page of the European Spreadsheet Risk Interest Group [167]. Several research systems have such supported functions in spreadsheets. They are all either imperative or trapdoors. Examples of trapdoors that are also imperative include Action Graphics [103], Spreadsheet for Images [127], and Penguims [102]. In their 1998's journal, Burnett and Gottfried introduce graphical definitions as the approach that supports direct manipulation and gestures within the spreadsheet paradigm of cells and formulas, "without employing state-modifying sublanguages, macros, or trapdoors to other programming languages" [39].

Error/Irregularities Identification: Although there are already a couple of possible methods to either enforce a systematic development of spreadsheets, according to software engineering principles [107] or to reduce the error rate of already existing spreadsheets, by testing [187] or auditing [195], they are still not widely accepted. Spreadsheet itself is by nature a prototyping tool, led to poor systematic development of spreadsheet programs. Also, testing and auditing approaches fails because the sheer size of the spreadsheets are common in industry. In 2002 their report of a field study auditing the spreadsheets of a large international company. Mittermeir and Clermont [149] examined 78 spreadsheets, with the average spreadsheet containing more than 2400 non-empty cells. Testing the whole spreadsheet, even with the support of current tools and techniques remains a tedious task.

It was found out earlier by Panko [172]that checking a spreadsheet is a time consuming and expensive task. Users can access some methods to assess the risk and the impact of an error in a certain region of the spreadsheet [41], but they are subject to the auditors attitude and might not map to the actual erroneous areas of spreadsheet programs. Markus Clermont proposes a visualization approach to identify certain irregularities of the spreadsheet through an comprehensible abstract visualization of the audited spreadsheet that is always connected with the spreadsheet system's user interface that is familiar to the spreadsheet programmer [55]. Through another auditing approach, Abraham and Erwig [1] present a system called UChecker that detects errors in spreadsheets through carrying out automatic header and unit inference, and reports unit errors to users. Apart from this, Ruthruff et al.'s recent paper [194] presents an empirical study conducted in spreadsheet environment to exam the impact of two separate factors in fault localization techniques that affect technique effectiveness: information base (the type of information maintained by a technique) and mapping (the way a technique maps the information into feedback)

Simulation: *Spreadsheet simulation* refers to the use of a spreadsheet as a platform for representing models and performing simulation experiments. Spreadsheet simulation simply involves using a spreadsheet to represent the underlying data model, do the sampling, perform the model computations and report the results.

Each simulation has its own data analysis requirements [14]. Seila [205] introduces that the cells in a spreadsheet model can be classified by their contents: input to the model, intermediate computations, and outputs from the model and conclude that "most models that can be organized in this way can be simulated in a spreadsheet." Adams et al.'s 2005 work describes the effect of combining hands-on simulation with spreadsheets simulations in learning system design. The evaluation shows that students can easily change various parameters of the Extend simulations using the dialog boxes so student can follow the flow of materials and information and learn from experimentation smoothly [2]. Kumiega and Van Vliet [117] developed a spreadsheet prototyping process that allows the software developing team to deliver for review and comment intermediate-level models based upon clearly defined customer requirements.

3.3.3.1 Tetrad Effects of Spreadsheet Modeling

The four sections noted above categorize spreadsheet modeling literature into strongly interrelated themes. While this four-section structure is not a very satisfying framework for organizing current theories, it serves as a useful structure for understanding the scope of spreadsheet modeling. In addition, it affords the possibility of analyzing media effects more directly without searching for cross linkages. Similar to the analysis of physically based modeling, Figure 3.4 is the representation of spreadsheet modeling's tetrad effects in the coordinate system.

Prospective - Positive What does it enhance?

- **Business management - decision making, accounting**: Modern spreadsheet applications provide a simple, direct, and powerful way for accountants and managers to manipulate and aggregate data directly. Many complex problems of finance, marketing, operations and human resources can be modeled and solved through spreadsheets. Relevant information is important for correct decision-making and good management. Spreadsheets assist people in marshalling this information into useful forms [150].

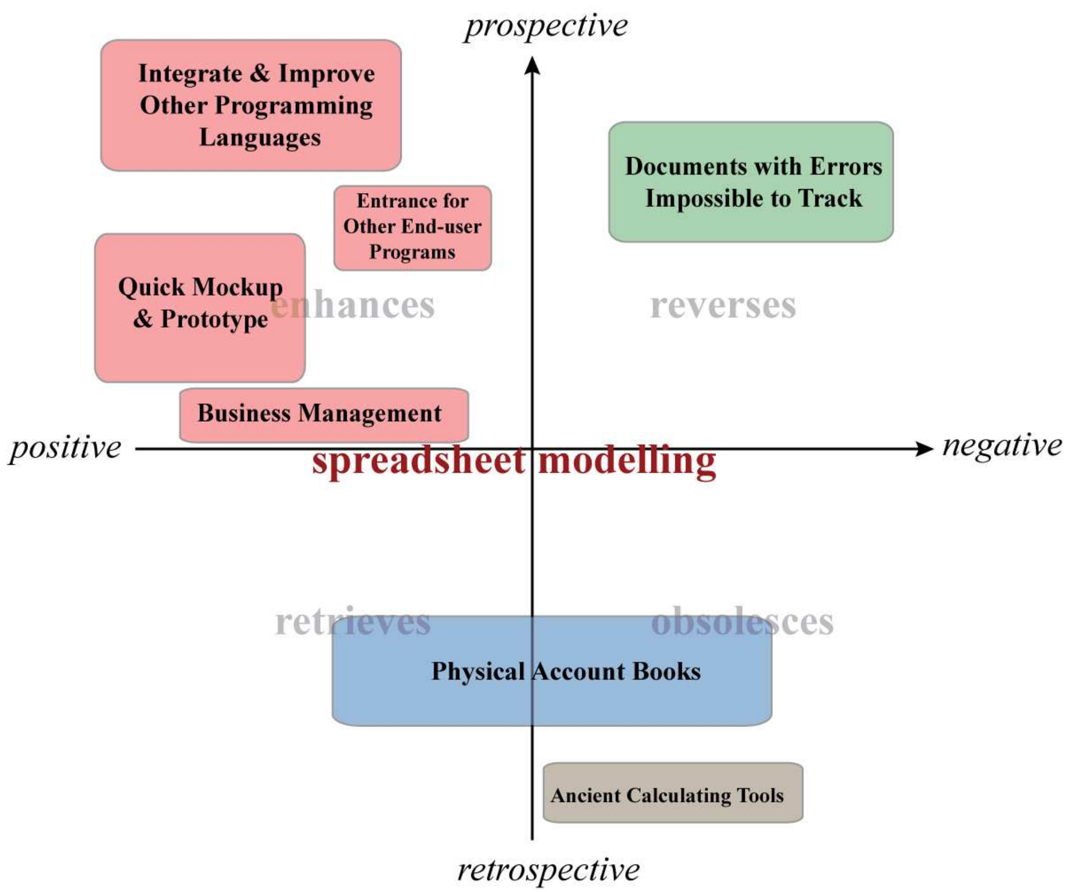

Figure 3.4: Tetrad Effects of Spreadsheet Modeling.

- **Integrate and improve other programming languages**: In the 2004 Oracle white paper, Laker and Shenker [119] introduce the architecture and key components of Oracle BI spreadsheet add-in and emphasized on how Excel users can create and manage directly from Excel and incorporate standard Excel features. They admitted that Excel features such as charts significantly enhance the visualization of the report. Actually not only Excel, many other spreadsheet systems also have the potential to be integrated into other languages, such as AIS's XESS [252].
- **Entrance for other end-user programs**: Being end-user programming environment itself, a spreadsheet system is also an easy entrance for users to learn other kinds of end-user programming systems. For example, Wong and Hong [246] developed an end-user programming tool called Marmite, which lets end-users create so-called mashups that repurpose and combine existing web content and services. They reported that "a user study found that programmers and some spreadsheet users had little difficulty using the system."
- **Quick mock-up or prototype**: Current technologies reviewed in the "simulation" section have the evidence that powerful functions of current spreadsheet systems can deliver quick prototypes for review and comment intermediate-level models for the software developing team [117, 2]. Although there are limitations mentioned by Seila [205], most of mock-ups do not need complex algorithms and large data storage.

Prospective - Negative What does it flip into when pushed to extremes?

- **Documents with errors almost impossible to track**: Survey studies in 1993 shown that each year, tens of millions of managers and professionals around the world created hundreds of millions of spreadsheets [142]. This number of today might have already turned into trillions. On the other hand, European Spreadsheet Risk Interest Group [167] has a long list of well document spreadsheet-error horror stories posted online. There is an early paper about using a framework to merge spreadsheet files [156]. Cells in a spreadsheet are parametric linked. How to archive, compose and reuse the previous spreadsheet files would be a big challenge. Errors existing in current files make the task more difficult.

Retrospective - Negative What does it make obsolete?

- **Ancient calculating tools such as Abacus**: An abacus is a calculating tool for performing arithmetical processes. The Babylonians started to use dust abacus as early as 2400BC [176]. Chinese version of abacus, Suanpan, has been used as the main calculation tool for bookkeeping till the middle of last century. Ancient calculating tools like abacus had been replaced by digital calculator decades ago. There were different kinds of calculators developed to meet various needs such as retirement calculator, load repayment calculator, etc. Some of these were developed upon spreadsheets [215].
- **Physical Account Book**: Accountants have used physical spreadsheets for hundreds of years. In digital spreadsheet modeling, the metaphor of linear programming and column-based matrix layout still originate from their physical ancestors. From the paper based account book to digital bookkeeping, it was a long-term smooth transition. From one perspective, spreadsheets make account books obsolete.

Retrospective - Positive What does it retrieve?

- **Physical Account Book**: From another perspective, spreadsheets also retrieve the culture of account books. They did what account books were doing in the old times. Moreover, they could be complex calculating tool, programming tools, and learning material to other programming.

3.4 Parametric CAD Systems

As a family of computing systems intended to facilitate design activity and amplify design practice, computer aided design (CAD) systems incorporate a project's trajectory from draft to implementation, absorbing and reflecting all decisions taken and changes made [112]. Conventional architectural CAD software comes in two flavors: drawing tools with symbol libraries and "intelligent" tools that offer component-level design (e.g. walls, doors, and windows). With drawing tools, edits to the design may impact hundreds of drawings, which must be done manually and require informal ordering systems [200]. At this 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. Many small mistakes on the drawings are hard to detect. Consequently human error is a critical contributor to lapses in system design [241]. “Intelligent” solutions aim to overcome human limitations by using object-oriented design. However, these approaches are aimed primarily at producing documentation and usually fail to model buildings with innovative form. These restrictions in current systems are impediments to exploratory prototyping and comparison of potential solutions, as every small change has to be manually managed and coordinated between collaborators on the project. In addition, the technological tools deployed today are designed to service the unique needs of each separate discipline, not to integrate information across disciplines to improve accuracy and reduce time [24].

3.4.1 A Propagation-based Modeling System: GenerativeComponents™

In the domain of architecture and building design, parametric modeling software such as Revit, CATIA, and GenerativeComponents have introduced computational mechanisms and interfaces for representing variation in architecture and building design [250]. Revit combines data structures representing building elements with two-way constraints [189]. CATIA imposes a strong abstract structure of data with access to a constraint solver and sophisticated graphical debugging methods [45].

Bentley’s GenerativeComponents™ (GC, shown by Figure 3.5) combines a propagation-based modeling system with a set of geometric types, and a layered structure of access to the system through programming languages of increasing sophistication [5]. In GC, designers build models by creating instances of “feature”, an application primitive that typically takes the form of a geometric object. Features are organized in a graph that models the process that constructs the design computationally. GC provides a symbolic model view (left bottom view in Figure 3.5), in which nodes are features and links are relations between features. The design is generated by updating each node, working from independent features with known values to dependent features with unknown values, in the manner of a propagation based system. Using such a tool, it is possible to develop models that support discrete variation. However, 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 techniques can be explored and tested in actual use.

GC has been used in practice, reaching firms such as Foster and Partners on works such as the British and Smithsonian Museum courtyard roofs and the SwissRE headquarters in London, Arup Sports on the Beijing Olympic Stadium, Kohn Peterson Fox on the World Bank headquarters, Thompson Ventulett Stainback on the Lagoons, and by Morphosis on the Wayne L. Morse U.S. Courthouse. Through ongoing workshops run by the SmartGeometry organization, the community of GC users is evaluating and improving the structure and interface to make it more communicative and supportive for architects, civil engineers and fabricators. These workshops provide an opportunity for us to observe, understand how people use the system to design individually or collaboratively, and suggest new ideas to support the process of collaborative design.

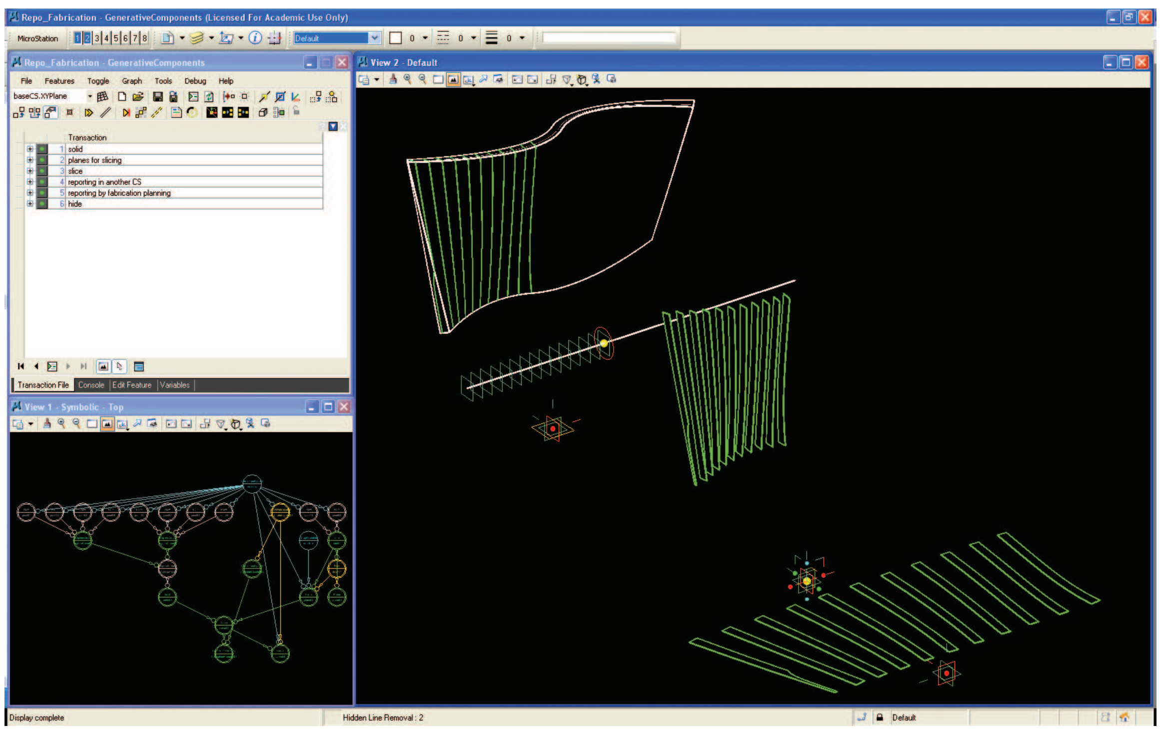

Figure 3.5: Screenshot of a GenerativeComponents Model.
In GenerativeComponents, models are represented in transaction process (left upper), symbolic (left bottom), and 3D geometric (right) views. This model reports the fabrication results through slicing a piece of curved solid.

3.4.2 Parametric CAD Systems Support Design

Many people know that parametric software in architecture can translate and output complex shapes fabrication and production modes. Except this final steps of the design process, the initial design process requires "intuitive" (graphic interface) modeling software. However, if used as an input tool, parametric setups promise an entirely novel approach to the field of architecture, that can be identified through a set of specific characteristics [116]:

Diversity "A parametric design system is based on a hierarchy of variable-controlled dependencies. Each active variable causes the overall system to change its behavior and thereby generate variations without losing the overall coherence and integrity of the model. This is an incredibly powerful tool, which allows designers to test a range of possible applications within a controlled design environment.

Adaptability Which variations are "fitter" to solve a particular design problem? Parametric models can be fine-tuned toward specific environmental pressures. Versions can be tested toward particular performance values. Different regions within the same model can be programmed and constrained individually in order to respond more sensitively. Environmental changes that occur over time at the same location can be addressed by layering functional zones and controlling them through targeted variable manipulations.

Responsiveness/Feedback A parametric model maintains its ability to change (openness) throughout the entire design process. Individual components can be linked in ways that allow constant and immediate feedback throughout the model regardless of scale and hierarchy. The design of the network of dependencies becomes a crucial part of the overall design by highlighting certain relationships over others and thereby allowing for specific local changes to impact the model's global scale and behavior."

The design and construction industries are in the midst of a fundamental transformation towards so-called "digital practice" [24]. This transformation is driven by a confluence of factors: new parametric modeling and digital fabrication technologies, low-cost high-bandwidth communication, and the globalization of practice. A feature of this transformation is that it is being driven by a need to share information, learning, and techniques at an explicit level across firms and schools. Parametric design systems support the process of collaboration and communication in CAD design domain:

Share design process: Parametric design systems record the creation process, retaining the design logic so that collaborators can interpret the originator's work. As a result, other designers can reinforce the sense of design in the group.

Explore design space: Traditional CAD systems only show one instance of a design. In a collaborative environment, participants want to explore different alternatives together and decide on the best choice. Until now, paper-based sketching remains the usual exploration method. In a parametric model, it is straightforward to review different variations through adjusting the parameters. A built-in recording tool can record key steps and stages for further analysis.

Collaborate widely: In many cases, architects have to adjust the design slightly here and there to meet others' needs (such as clients, engineers, fabricators, and contractors). Editing parameters is much more direct and secure than editing the values of an existing artifact.

Facilitate interdisciplinary collaboration: Parametric modeling makes use of explicit computational and mathematical expressions produced by experts in computer science, mathematics and engineering [24], making it possible for them to help architects understand and implement complex geometry problems in their design projects.

Parametric modeling is transforming practice, both within and between firms. But it is evident that this transformative technology imposes new needs for a higher level of support for its communities of practice. I conjecture that design patterns are a good device around which to structure such support.

3.5 Summary

In this chapter I outlined and examined fundamental concepts around *parametric design systems*. Several distinct domains were visited to demonstrate the broad but scattered existence of all kinds of parametric design systems.

I selected physically based modeling and spreadsheet modeling as subjects of a revolution tracing to enrich the understanding of parametric systems. Located in the central domain of computer graphics, physically based modeling incorporates physical characteristics into models to simulate the real world and achieve realism. Many other domains such as earth sciences and musical industry also have a long history of adopting such a method. Originated from ancient account books, spreadsheet modeling reserves its accounting features and turns out to be a successful worldwide end-user programming tool. McLuhan's tetrad was adopted to analyze the media effects of these two system types. I introduced two coordinates – positive-negative and prospective-retrospective – to frame the tetrad so that it may be easier to recognize and analyze media effects.

In comparison to these system domains, parametric CAD systems are young and thin in content. However, some concluded effect ideas are transferrable, such as that technology enhances the collaboration across different methods and branches, a conflict exists between high quality rendering and continuous real-time updates, and that new parametric tools replace old style repetition based work methods. All of these would be valuable supplement knowledge for my research in the parametric CAD modeling systems.

Bentley's GenerativeComponents is the central platform for this PhD research. I reviewed the system's main features in the context of parametric CAD systems. Limitations of the available literature made it hard to further investigate the media effects of parametric CAD systems at the current stage. In Chapter Ten, I will discuss my interpretation of parametric CAD system's media effects based on data collect from a series of research studies.

Chapter 4

Design Patterns

A *design pattern* is a formal representation of a solution to a design problem within a particular field of expertise. Its description includes both problem, solution, and contextual information. In this research, the notion of design pattern is used as a vehicle for understanding and representing solutions to design problems that occur across different contexts and tasks. I review the state of design patterns research below.

4.1 Origins of the Notion of Design Pattern

Early thinking about patterns in design is evidenced within a number of prominent historical publications on architecture, including Leon B. Alberti's "The Ten Books on Architecture" in the 15th century [7], Andrea Palladio's "The Four Books on Architecture" in the 16th century [171], and John Ruskin's "The Seven Lamps of Architecture" in 1849 [193]. However, the contemporary notion of design pattern emerges principally from Christopher Alexander's work [8, 13].

Alexander investigated the process of designing things that display physical order, organization, and form in response to function. As he notes, "this information is hard to handle, it is widespread, diffuse, unorganized. The problems increase in quantity, complexity and difficulty, they also change faster than before ... The unfortunate result is that very few designers have tried to understand the process of design analytically [8, p.4]." By understanding the process of design analytically, he claimed that the use of logical structures to represent design problems could help designers interpret and solve problems systematically. "No complex adaptive system will succeed in adapting in a reasonable amount of time unless the adaption can proceed subsystem by subsystem, each subsystem relatively independent of the other [8, p.41]."

Alexander defines a pattern as, "a three part rule, which express a relation between a certain context, a problem and a solution." He calls the logical structure a *pattern language* and the subsystems *patterns* [13]. He compiled a comprehensive collection of 253 design patterns that appear in urban, buildings, and construction design. [13], and put it into detailed introduction of use in 1979 [10].

Alexander adapted his research in design patterns to a master plan for the University of Oregon[9]. He describes an empirical approach to the implementation of design patterns in a community. As evidenced in the book, his team was particularly concerned about the process of employing patterns. They believed that the process of building and planning in a community will create an environment that meets human needs only if it follows six principles of implementation: organic order - a process that allows the whole to emerge gradually from local acts; participation - putting the 'what' and 'how' of decisions in the hand of users; piecemeal growth, patterns, diagnosis and coordination" [9, p.4-6]. Examining some 250 patterns from Alexander's research [13], they found that about 200 of the 250 patterns were relevant to the university community. About 160 of these 200 dealt with building interiors, rooms, gardens, and building constructions. However, the team noted that, "37 of the 200 patterns relevant to the university are so large in scale that individual projects will not be able to complete them, and they will only appear at all if many different individual projects help to create them in cooperation [13, p.104]." These "large patterns" are difficult to handle and apply. If patterns really organize a language, I wonder if it is possible to use some language tricks such as dividing phrases into words to deal with these big patterns. If not, it is hard to say that these large patterns have any usage.

Alexander's original work saw limited success, and did not not achieved the author's intended goal. However, with the passage of time, design patterns have gained an acceptance in other disciplines. Software engineering in particular has made extensive use of design patterns [83]. Design patterns have also been adopted in workplace design [239], graphical user interfaces [230], web usability [89], socio-technical system design [134], education science [25], participatory design [68] and communication [191, 202].

4.2 Pattern-based Projects

This section introduces how different fields have adapted design patterns.

4.2.1 Object-Oriented Software Patterns

Although design patterns originate in architecture, it is in software engineering that they have found clarity of philosophy, concise expression, and wide utility [249]. Gamma et al. [83] adopted the notion of design pattern to object-oriented software design. Each design pattern systematically names, explains, and evaluates an important and recurring design in object-oriented systems. Patterns aid software engineers in abstracting and clarifying system designs, and increasing the reusability of software components. They note modestly that, "design patterns only capture a fraction of what an expert might know. It doesn't have any application domain-specific patterns, it doesn't tell you how to build user interface, how to write device drivers, or how to use an object-oriented database [83, p.2]." Gamma et al. do not describe the genesis of their patterns, but cite the Model/View/Controller (MVC) triad of classes used to build user interfaces in Smalltalk-80 as a source.

4.2.2 Workplace Design Patterns

As a former student of Alexander, Australian architect Dr. David Week adapted and extended Alexander's work. He restates Alexander's formulation that design patterns should express three fundamental facts [238]:

Context A ⇒ System of Forces B ⇒ Configuration C:

- A is the context in which the rule applies.
- B is the system of forces at work in that context.
- C is the spatial configuration that resolves that system of forces.

This formulation gives every pattern the form of a classical syllogism: if A and B, then C. B is the fact that allows us to infer C from A. Week argues that, "Alexander's pattern language extracted valuable knowledge for the practice of contemporary architecture from historic and traditional buildings," and emphasized that local forces of patterns are the important part that designers need to keep in mind. He encountered a difference between conventional Western patterns of architecture and the local economy or climate or way of life where designed artifacts were employed. The adaptationist paradigm encourages them to change the pattern to increase its specificity to the context.

According to Week, "Alexander was right in his theory of pattern languages, in suggesting that the way we use the environment and the way we make the environment are both intimately bound up in our linguistic understanding. Alexander drew on Whorf for this theory. He is wrong, though, in thinking of a pattern language as a special kind of language. A pattern language is just a subset of ordinary language, focused on the built environment. Language is constitutive – to be a human being is to live in a world that is linguistically given. This hermeneutic view of language stands in contrast to the symbolic view that language is representational [238, p.64]." In short, he emphasizes that without fully understanding the real context and environment of patterns, it is dangerous to *define* a language for the pattern group.

To demonstrate an application of pattern theory, Week published a pocket book entitled "The Culture Driven Workplace" [239] that adapted Alexander's work in a highly pragmatic way. In the book, he describes a number of patterns that can be used in workplace design, from the perspectives of environment, lighting, noise, furniture, space efficiency and office function. These perspectives illustrate potential focii for a workplace designer's attention, and provide solutions for the designer if he/she lacks knowledge or experience about these aspects. He 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."

4.2.3 Graphic User Interface Patterns

Being an interaction designer and software developer, Jenifer Tidwell examines patterns in the domain of graphical user interface (GUI) design [230]. She describes patterns as providing generic design guidance in a format that is consistent and easy to understand, and that conveys knowledge about the attributes of GUI design. Tidwell also wants her patterns to inspire designers and emphasizes on users' read needs, "If you're running short on ideas, or hung up on a difficult design quandary, read over these and see if any of them are applicable. And don't take them as the gospel truth, either – what matters is whether your design works for your users." Her book enumerates patterns for common interface design problems. Each pattern contains practical advice that the user can put to use immediately, plus some good examples to support the explanation. The user can get recommendations, design alternatives, and warnings on when not to use them. Each chapter's introduction describes key design concepts that are often misunderstood, such as affordance, visual hierarchy, navigational distance, and the use of color. Although some problems in GUI design are comparably trivial and detailed, Tidwell tries to name patterns in a succinct way with no more than three words. This naming style is later adopted by my parametric design pattern project.

4.2.4 Interaction Design Patterns

Borchers explains how interface designers can structure and capture user interface design knowledge from their projects and learn to understand each other's design principles and solutions through patterns [30]. Key features of his work include a comprehensive pattern language for interface design in interactive exhibits as well as a thorough introduction to original pattern work and its application in software development. Borchers also provides design techniques and pattern languages for documenting and communicating interaction design patterns using an XML-based notation. He compares different pattern languages and states that an important goal of any design team is to capture the reasons for design decisions and the experience from past projects, to create a corporate memory of design knowledge.

4.2.5 Other Pattern-based Approaches

Design patterns have also been adopted in other domains, to aid in understanding the structure of human problem solving activities.

Distributed Scaffolding Patterns - Tabak applied design patterns to understand scaffolding in learning. He introduced the notion of distributed scaffolding, an emerging approach in the design of supports for rich learning environments intended to help students develop disciplinary ways of knowing, doing, and communicating [225]. Tabak synthesizes research to date, and articulates three patterns of distributed scaffolding and the pedagogical considerations that they target. The Synergy project referred to the characteristic that different components of distributed scaffolding, such as software supports and teacher coaching, address the same learning need and interact with each other to produce a robust form of support. Tabak illustrated these patterns through classroom examples and summarized the scaffolding functions that it can fulfill.

Cooperative Interaction Patterns - CI patterns are regularities in the organization of work, activity, and interaction among participants and around artifacts. Martin and Sommerville employ patterns as a means for reusing findings from previous field studies, enabling analysis and considering designs in new settings [136]. Instead of conceptualizing design patterns as 'reusable templates', their work aims to use them as resources to be "drawn upon, selected and used sensitively and creatively." In both the article a supporting website, CI patterns are described at the level of vignette as being composed of three components: socio-technical configuration in a particular setting (what), two or three comparable examples and implications for achieving the work.

Information Visualization Patterns - Granlund et al. presented a pattern- supported approach to user interface design in the context of information visualization in 2001 [90]. Using an example from the telecommunications domain, they focus on a task/subtask pattern to illustrate how knowledge about a task and an appropriate interaction design solution can be captured and communicated. There are two main differences between task and subtask pattern description. Subtask forces are generally a subset of task forces or derived from them. They are specific to the subtask, and are not divided into components such as Task, User, and Context Forces. This research provides one of the few examples where there is a consideration of linkages and hierarchy among patterns.

Pedagogical Patterns - Pedagogical patterns try to capture expert knowledge of the practice of teaching and learning in a compact form, that can be easily communicated. Bergin et al. [25] introduces pedagogical patterns that deal with the range of instructional techniques in recent project reports. They maintain a website of pedagogical patterns organized on the basis of course sequence.

Collaboration learning Patterns - DiGiano et al. [73] aim to create conceptual tools that help people think about technology-supported collaborative learning. They find patterns useful for guiding technical discussions while providing non-technical templates for generating high-quality collaborative learning activities that make the most of emerging technologies. They use 'collaborative design patterns' to capture common learning situations and tradeoffs in written form, and introduce three families of patterns: “Moon Phrase Patterns”, “Revolutionary War Patterns”, “Salmon Watch Patterns” and the “Bias Circle Pattern”.

Wiki Patterns - Focusing on providing practical advice for encouraging adoption of wiki projects, Mader [133] give wiki users a toolbox of wiki patterns that explain the major stages of wiki adoption and enable newcomers to avoid making common mistakes. The structure of wiki patterns not only includes basic components such as name, related patterns, ‘what is it’, usage, and examples, but also collects users’ comments and editions through an online repository. Wiki patterns are mainly categorized into four groups: “people patterns”, “people anti-patterns”, “adoption patterns”, and “adoption anti-patterns”.

Publications on design patterns, such as Alexander et al.[13], Gamma et al. [83], Week [238] and Tidwell [230], reflect the continual evolution of the design pattern concept. This PhD research does not 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 *completed* pattern language. In fact, Week’s work of several informally defined workplace patterns and Tidwell’s growing pattern team have won recognition from users and other experts. My research will start with several semi-dependent but useful patterns to support parametric design practice.

4.3 Distinctive Characteristics of Patterns

To introduce my research topic to audience who is new to the concept “pattern”, I sometimes used terms such as heuristics, guideline, and strategy to describe and explain. They have some close meanings with the pattern concept. For example, in Monahan 2000’s spreadsheet modeling book “Manage decision making”, he uses the concept of *models* to describe problems with context and discuss the best practice to build, use, and communicate such spreadsheet models [150]. The structure of such models is also close to that of the design patterns. However, design patterns have a number of characteristics that distinguish them from those similar conceptual entities:

Derived from practice - Patterns are designed to capture useful and used practice knowledge from a specific domain [249]. Although some pattern projects claims to search for the "best" solution for certain problems [25], I contend that usefulness and grounding in practice are the most important qualities for patterns. Good patterns may be inspired by literatures or other design guidelines, but whether or not a pattern is useful can only be tested by real users in their working contexts. A pattern is important only if it is useful and has been used [249].

Communicative - Patterns serve as a means to talk about and share experience within a community [249]. Although sometimes the description is not as explicit as a heuristic or as detailed a piece of design guideline, a qualified pattern has a memorizable succinct name, a clear description and sufficient examples to help users understand the problem's context and solution. The explicit structure supports the communication among the pattern users.

Transferable - Patterns are structural and interactional abstractions. There are phenomena that patterns exhibit similar, and sometimes identical properties, in different domains. Patterns invented in one domain can often be transposed into another. Software engineers have not only adopted the notion of design patterns, but have also transferred some pattern ideas from urban design to system architecture. In the process of authoring parametric design patterns, I have likewise referred to object-oriented software patterns. Since problems are understood at a conceptual level, some pattern ideas are transferable across different disciplines.

Hierarchical - Patterns exhibit properties that differentiate them into classes and lead to hierarchies of patterns. For example, Gamma et al. [83] describe three classes of patterns in software systems: creational design patterns, that aid in making a system independent of how its objects are created, composed, and represented; structural patterns, that are concerned with how classes and objects are composed to form larger structures; and, behavioral patterns that are concerned with algorithms that assign responsibilities between objects.

Stable but impermanent - Theoretically, patterns should be very stable because they should serve different problem contexts. For well-defined design problems: especially, in software engineering, "a design pattern is a general reusable solution to a commonly occurring problem in software design [83]". A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. Object-oriented design patterns typically show relationships and interactions between classes or objects, without specifying the final application classes or objects that are involved. Algorithms are not thought of as design patterns, since they solve computational problems rather than design problems. For solving ill-defined problems, especially in design domains, the significance and stability of design patterns are not so obvious. For example, Jennifer Tidwell keeps on collecting UI patterns through operating an online journal because patterns are more concerned on current ideas for thinking and sharing. Without the context of current technologies and culture issues, patterns can be easily out of date and useless.

4.4 Research Questions Around Patterns

While much has been written about design patterns, there are essential elements of knowledge missing that would make patterns a much more useful concept in research:

Research methods to elicit patterns - Most pattern research describes patterns as arising from the analysis and elicitation of information in real practice. However, the procedure and validity of such processes has not been thoroughly elaborated in any of these publications.

Research methods to evaluate patterns - Few publications speak to the process of evaluating pattern work [68, 71]. However, all of them were conducted through designed case studies rather than a practice context. Determining whether patterns work in real practice is still difficult.

Consideration of cognitive issues - Communicating design patterns is an important aspect of their development. Alexander emphasized the importance of providing well-drawn diagrams [10], but few projects consider the cognitive design issues around pattern use and supporting discourse about patterns.

Adopting the pattern construct to encapsulate useful design practice in parametric design systems, this research serves to explore missing issues around the knowledge of patterns. Research methods to elicit and evaluate patterns are first introduced in Chapter Five and then adopted in studies described in chapters Seven and Ten. A discussion of cognitive issues around parametric design patterns follows in Chapter Ten.

Chapter 5

Research Methodology

This PhD research aims at supporting and augmenting design action in parametric CAD systems. The research focuses on a user base comprising architects and civil engineers.

I contend that:

- It is possible to model a designer's learning and working process by patterns in parametric design systems.
- Making such patterns explicit result in improved expert work practices, in better learning material and support collaborative design in parametric design interfaces.

The research work has been divided into four stages, each requiring a different methodological approach. The stages of work are:

1. search for first evidence of pattern-like phenomena through a pilot study
2. gather and elicit evidence for pattern-like thinking through a series of participant observation studies
3. develop parametric design patterns collaboratively
4. evaluate pattern-based tools in use

These studies aim to identify devices for aiding professional designers in executing design work, as well as insights to help developers improve the parametric design application. In the following sections, I evaluate epistemological premises and research methods for these stages.

In general, this research is constituted as a series of qualitative studies of architects and engineers working on real design problems. I employ Bentley's GenerativeComponents^{TM} (GC) parametric CAD application throughout. This application is used in professional practice, in design firms around the world. Through workshops that have taken place over the last six years, I have been afforded with access to a community of GC users who are evaluating and improving the application's structure and interface, to make it more communicative and supportive for professional practice. These events have provided an opportunity to observe and understand how practitioners use the application to design, and develop new ideas to improve the process of parametric modeling.

5.1 Research Methods Framework

Crotty suggested that four questions suffice to describe the fundamental elements of any research process [63, p.8]:

- "What methods do we propose to use (e.g. questionnaire, interview, case study, focus group, etc.)?
- What methodology governs our choice and use of methods (e.g. experimental research, ethnography, etc.)?
- What theoretical perspective underlies the methodology in question (Positivism, Interpretivism, Critical Inquiry, etc.)?
- What epistemology informs this theoretical perspective (Objectivism, Constructionism, Subjectivism and their variants)?"

These four questions show the interrelated levels of decisions that go into the process of designing research. However, defining that paradigms are "basic belief systems based on ontology, epistemological, and methodological assumptions [95, p.107]," Guba and Lincoln listed alternative inquiry paradigms as positivism, postpositivism, constructivism, and critical theory. It seems that the categories of epistemology and theoretical perceptive are sometimes blurred. John Creswell reframed Crotty's model as three questions that require answering in any research design [60, p.5]:

- What knowledge claims are being made by the researcher (including a theoretical perspective)?

- What strategies of inquiry will inform the procedures?
- What methods of data collection and analysis will be used?

This structure of knowledge claims, strategies of inquiry, and methods is used as the research method framework of this PhD research.

5.1.1 Choice of Knowledge Claims

When researchers start a project with certain assumptions about how they will learn and what they will learn during their inquiry, I refer to the claims they make as knowledge claims [60]. These claims might be called paradigms [95], philosophical assumptions, epistemologies, and ontologies [63] under different context. Alternative knowledge claim positions list by Creswell are postpositivism, constructivism, advocacy / participatory, and pragmatism.

I should carefully choose the correct paradigm according to the type of the research and its context. For example, a close related research in the design domain, Susan M. Pistsch's PhD dissertation [174] initially assumed a positivist paradigm, which is the dominant paradigm within the field of urban visualization and modeling where she was working. However, as the process unfolded, she found that it became more difficult to maintain. She outlined the gaps in positivism for this study, including the lack of discussion about the social, administrative and political issues that affect the acceptance and use of a technology, and the limited control available in the environment and the contextual nature of the research [174, p.56]. Constructivism believes that we human beings generate knowledge and meaning based on reflecting on our own experiences. A constructivism position on the research later appeared to be a possible alternative for her for a number of reasons [174]:

- Constructivism is capable of dealing with certain shortcomings of positivism, such as asserting relativist ontology.
- Constructivism allows the researcher to discuss the variety of opinions expressed by participants involved without privileging one over the other based on *truth* claims.
- The mechanism of correlating material with the researched-into allows the reader to be aware of whose construction of the story they are dealing with.

In this research, designers create and give meanings to data as they engage with the world that they are interpreting using parametric modeling systems. Open-ended questions may be used so that participants can express their views. The process of generating meaning is always social, arising in and out of interaction with a community. Therefore, I employ a constructivist paradigm as the principle stance in the research inquiry.

5.1.2 Choice of Research Approach

Knowledge claims, research strategies and methods all contribute to an research approach that tend to be quantitative, qualitative, or mixed. A quantitative approach is one in which the investigator primarily used postpositivism claims for developing knowledge. A qualitative approach is one in which the researcher makes knowledge claims based primarily through a constructivist stance [59]. Through a thorough literature review, this research topic is found to be relatively new. It is exploratory by nature because the content of design patterns will remain unknown and the research outcome is difficult to examine using a quantitative approach. Naturally, the research methodology of this PhD research should be primarily qualitative. However, at the current stage, there are needs for the mixed methods approach.

While the qualitative methods still maintain a dominant position in the approach, some minor quantitative studies such as a survey should be employed sequentially for better understanding of research problems and outcomes. Such integration can enrich the content of study, use different forms of data to support each other and provide sufficient choices for the evaluation stage. The choice of mixed methods approach wants to leave an opening for conducting simple but supportive quantitative analysis to support the principal qualitative studies.

5.1.3 Choice of Inquiry Strategies

Strategies of inquiry (or traditions of inquiry [59]; or methodologies [145]) provide specific direction for procedures in a research design. In Creswell's 2003 book, strategies of inquiry include experiments, surveys, ethnographies, case studies, grounded theory, etc. For each stage of this PhD research, the strategy of inquiry and its specific method might vary. Following are the two main inquiry strategies I adopted in the research.

5.1.3.1 Method of Participant Observation

Suzuki et al. [224] discusses four sources of qualitative data: participant observation, interviews, physical data, and electronic data. They argue that although researchers often use interviews in their work, participant observation, physical and electronic data are also beneficial ways of collecting qualitative data that have been underutilized. DeWalt and DeWalt highlight the advantages of participant observation, including, "enhancing the quality of the data obtained during fieldwork, enhancing the quality of the interpretation of data, and encouraging the formulation of new research questions and hypotheses grounded in on-the-scene observation [72, p.8]."

It is important to recognize that this method combines two somewhat different processes: observation and participation. Spradley [214] suggested a continuum of levels of researcher participation: "The minimal level is nonparticipation, observation only from outside the research setting, and passive participation, where the researcher is present but does not participate or interact. At the other extreme, the researcher plays an active role in activities in order to gain a greater understanding of the cultural norms and mores. Complete participation occurs often when the researcher is already an ordinary participant or a member of a community [214, p.61]". Most studies fall somewhere in the middle of this continuum.

Meyer suggests that participant observation is a way of gaining insight that is distinctly different from the methods traditionally used in the physical sciences. Researchers use their reactions to understand the views of others people and to formulate hypotheses about the reactions of other participants [146]. In this sense, participant observation is a kind of 'disciplined subjectivity'. She advises knowledge engineers to expect to be uncomfortable with having a marginal status, of not being wholly a researcher nor an actual insider while doing participant observation. This qualitative approach has been used in prior research projects to identify the differences between how novices and experienced designers approach design tasks [3], understanding how architects integrate sketching and parametric modeling in the design process [196], and designing an evaluation protocol for learning object development [88].

5.1.3.2 Method of Contextual Inquiry

Contextual inquiry is a qualitative methodology adapted from the fields of psychology, anthropology and sociology. Raven and Flanders describe how technical communicators can use contextual inquiry to gather information about their audiences and their specific needs for online and hardcopy documentation [186]. They describe contextual inquiry as they had used it in a professional work environment, and provide examples. There are three basic principles they emphasize [186]:

1. Data collection must take place in the context of the user's work.

2. The data-gatherer and the user form a partnership to explore issues together.

3. The inquiry is based on a clearly defined set of concerns rather than on a list of specific questions.

Raven and Flanders describe three implementations of contextual inquiry [186]:

1. *work-based interview*, where a user is interviewed while they perform their work

2. *post-observation inquiry*, where the process is observed and an interview is conducted after the completion of the work

3. *artifact walkthrough*, where the interviewer and interviewee go through the procedure of how the artifact is created step by step

Suzuki et al. [224] distinguish participant observation and interview as two different data sources. The method of contextual inquiry collects these two types of data at the same time. The observer can collect participant observation data from subjects immediately during the inquiry process.

Beyer and Holtzblatt [27] provide a practical, hands-on guide for researchers trying to design systems that respond to what users want to do in their work, and for teams that must design products and internal systems. Contextual inquiry is employed as a specific method for gathering field data from users. It is usually done by one interviewer speaking to one interviewee at a time. The aim is to gather as much data as possible from the interviews for later analysis. There are typically four phases in the interview: a traditional interview, the "switch" (which forms a master-apprentice relation), observation, and summarization (where summarize and confirm what has been learned in the interview with the interviewee) to confirm its validity. Beyer and Holtzblatt [27] also outline principles and examples of how data can be gathered in contextual inquiry to advise prototype designs. The advantage of contextual inquiry is that the interviewer can share their findings with the interviewee. However, it is difficult to create a situation that the interview itself does not disturb the interviewee's work practice. To keep a balance between gathering more information and disturbing less on interviewee, this study employed contextual inquiry as a minor technique component of the larger study.

5.2 Strategies for Establishing Reliability and Validity

To paraphrase Morse et al. [153], without rigor, research is fiction. Hence, a great deal of attention is applied to reliability and validity in all research methods. Challenges to rigor in qualitative inquiry interestingly paralleled the blossoming of statistical packages and the development of computing systems in quantitative research. One of the main issues that surround the use of qualitative data is how to determine the quality of the research. The rigor of research work is determined by its *internal validity*, *external validity*, *reliability*, and *objectivity*.

Internal Validity is defined conventionally within the positivist paradigm as the extent to which variations in an outcome or dependent variable can be attributed to controlled variation in an independent variable. Cook and Campbell state that the, "approximate validity with which we infer that a relationship between two variables is causal or that the absence of a relationship implies the absence of cause" [58, p.37]. Assessing internal validity is the central means for ascertaining the *truth value* of a given inquiry. There are a number of putative threats to the internal validity of any inquiry - including history, maturation, testing, instrumentation, statistical regression, differential selection, experimental mortality, and selection.

External Validity is defined by Cook and Campbell positivistically as, "the approximate validity with which we infer that the presumed causal relationship can be generalized to and across alternate measures of the cause and effect and across different types of persons, setting, and times [58, p.37]." LeCompte and Goetz [123] noted that in conventional inquiry, threats to external validity including selection effects, setting effects, history effects, and construct effects.

Reliability in positivist terms respond to questions about the consistency of a given inquiry and is typically a precondition for validity. It refers to a given study's consistency, predictability, dependability, stability, and/or accuracy [58].

Objectivity responds to the positivist demand for neutrality, and requires a demonstration that given inquiry is free of bias, motivations, interests, values, prejudices, and/or perspectives of the inquirer.

Within the framework of logical positivism, the foregoing criteria are reasonable and appropriate because they have their foundational assumptions rooted in the ontological and epistemological framework of that paradigm (model) for inquiry. But the traditional criteria are unworkable for constructivist, responsive approaches on axiomatic grounds [93].

5.2.1 Trustworthiness

What criteria would be meaningful within a constructivist inquiry? To assist researchers and readers to determine the quality of qualitative research, Guba and Lincoln developed the so called *trustworthiness criteria*, that are parallel to the conventional four criteria of validity. They are: *credibility*, *transferability*, *dependability* and *confirmability* [94, p.236-243].

Credibility corresponds to *internal validity* in that the idea of isomorphism between findings and an objective reality is replaced by isomorphism between constructed realities of respondents and the reconstructions attributed to them. Instead of focusing on a presumed 'real' reality, the focus has moved to "establishing the match between the constructed realities of respondents and those realities as represented by the evaluator and attributed to curious shareholders [94, p.239].". Lincoln and Guba suggested several techniques, such as prolonged engagement, persistent observation, peer debriefing, negative case analysis, progressive subjectivity, and member checking, to ensure credibility [128].

Transferability is parallel to *external validity* or *generalizablility*. Guba and Lincoln state that, "rigorously speaking, the positivist paradigm requires both sending and receiving contexts to be at least random samples from the same population [94, p.241]." In comparison to the concept of generalizablility, transferability is always relative and depends entirely on the degree to which salient conditions overlap or match. So, the constructivist does not provide the confidence limits of the study. Instead, s/he provides the complete data base to facilitate transferability judgements.

Dependability is parallel to the conventional criterion of *reliability*, in that it is concerned with the stability of the data over time. But dependability specifically excludes changes that occur because of overt methodological decisions by the evaluator or because of maturing reconstructions. Shift in hypotheses, constructs, and the like are thought to expose studies to unreliability. Guba and Lincoln [94] suggested the technique for documenting the logic of process and method decisions to be the dependability audit.

Confirmability corresponds to the conventional criterion of *objectivity*. There are threats to each of these constructs that must be countered by specific methodological steps in the evaluation design. The presence of these steps is the basis for judging whether the results may be considered trustworthy. Cronbach and Suppes argue that data must be available for inspection and confirmation by outside reviewers of the study [61]. So, the usual technique for confirming the data and interpretations of a given study is the confirmability audit.

In 1981, Guba introduced a procedure for ensuring trustworthiness criteria: "triangulation and member checker (for credibility), thick description (for transferability), leaving an audit trail (for dependability), and triangulation and practicing reflexivity (for confirmability)[93, p.88]." These four trustworthiness criteria create a framework for dealing with qualitative data and addressing concerns with its collection. I employ this framework to ensure validity within my studies. Guba and Lincoln provide an addition set of authenticity criteria – fairness, educative authenticity, catalytic authenticity, and tactical authenticity – that consider the truthfulness of origins, attributions, commitments, and intentions in constructivist research [93]. In light of issues of relevance and volume, I exclude this additional criteria set.

5.2.2 Risks of Post-hoc Approach

Over the past two decades, reliability and validity have been subtly replaced by criteria and standards for evaluation of overall significance, relevance, impact, and utility. Strategies to ensure rigor inherent in the research process itself were supplanted by these new criteria to the extent that, while they continue to be used, they are less likely to be valued or recognized as indices of rigor. Morse et al. [153] were concerned that focus on evaluation strategies that lie outside core research procedures results in a deemphasis on strategies built into each phase of the research, strategies that can act as a self-correction mechanism to ensure the quality of work.

In Latin post-hoc means "after this". Post-hoc analysis, in the context of design and analysis of experiments, refers to the process of looking at data-after the experiment has concluded-for patterns that were not specified a priori. This is also known as data dredging, evoking the sense that the more one looks the more likely something will be found. More subtly, each time a pattern in the data is considered, a statistical test is effectively performed. Morse et al. [153] argued that Guba and Lincoln's approach to evaluate the quality and rigor in qualitative research was post-hoc (evaluative). This shift from constructive (during the process) to evaluative (post-hoc) procedures occurred subtly and incrementally. Now, there is often no distinction between procedures that determine validity in the course of inquiry and those that provide research outcomes with such credentials. The literature on validity has become muddled to the point of making it unrecognizable, as Wolcott notes: "Whatever validity is, I apparently 'have' or 'get' or 'satisfy' or 'demonstrate' or 'establish' it [245, p.121]."

Rigor is supported by tangible evidence using audit trails, member checks, memos, and so forth. If the evaluation is positive, one assumes that the study was rigorous. Morse et al. challenged this assumption and suggest that, "these processes had little to do with the actual attainment of reliability and validity. Contrary to current practices, rigor does not rely on special procedures external to the research process itself" [153]. In short, they argue that while strategies of trustworthiness may be useful in attempting to *evaluate* rigor, they do not in themselves *ensure* rigor. Similarly, while standards are useful for *evaluating* relevance and utility, they do not in themselves *ensure* that the research will be relevant and useful.

5.2.3 Verification Strategies

Verification refers to the mechanisms used during the process of research to incrementally contribute to reliability, validity, and thus the rigor of a study. Morse et al. argue that these mechanisms should be woven into every step of the inquiry to construct a solid product by identifying and correcting errors before they are built in to the developing model and before they subvert the analysis [153]. The process of qualitative research is *iterative* rather than linear, so a good qualitative researcher moves back and forth between design and implementation to ensure congruence among question formulation, literature, recruitment, data collection strategies, and analysis.

How can I check data systematically, maintain focus constantly, and monitor data analysis and interpretation continuously? Morse et al. emphasize on *investigator responsiveness*, meaning that, "the investigator should remain open, use sensitivity, creativity and insight, and be willing to relinquish any ideas that are poorly supported regardless of the excitement and the potential that they first appear to provide [153, p.11]." The investigator intends to be attracted by the 'exciting' ideas emerging during the data collection and analysis process. To judge whether or not such ideas should be relinquished is crucial to the attainment of optimal reliability and validity.

Within the conduct of inquiry itself, Morse et al. list five verification strategies that support rigor [153, p.11-13]:

Methodological coherence Ensures congruence between the research question and the components of the method (ie. - data and analytic procedures). Because the research process may not be linear, the sampling plans may be expanded or change course altogether.

Sample must be appropriate Ensures the efficient and effective saturation of categories, with optimal quality data and minimum dross. *Sampling adequacy* means that sufficient data to account for all aspects of the phenomenon have been obtained [152].

Collecting and analyzing data concurrently This forms a mutual interaction between what is known and what one needs to know. It is also the key strategy to ensure the rigor of iterative interaction between data and analysis.

Thinking theoretically Requires macro-micro perspectives, inching forward without making cognitive leaps, constantly checking and rechecking, and building a solid model. Such a strategy is used primarily because new ideas emerge constantly from data analysis.

Theory development The researcher should move with deliberation between a micro perspective at the data level and a macro conceptual/theoretical understanding. A valid theories theories should be comprehensive, logical, parsimonious, and consistent.

5.3 Summary

This chapter has outlined the rationale and methods used for collecting data in the associated research studies. This approach is the basis for the practice narratives in later chapters.

By adopting a constructivist approach, a fuller exploration of the participants's design process in parametric design system is permitted. I contend that the constructivist theory of perception is consistent with the methodology proposed. The inquiry methods of participant observation and contextual inquiry are particularly critical to achieving this exploration.

Investigator responsiveness concerns on the researcher's objective research attitude, and verification strategies list the details of how to conduct the data collection and analysis through a rigorous approach. I would like to integrate both of them as a system of *verification criteria*. For this PhD research, I want to both ensure the study's rigor and evaluate the relevance and utility during the study process consciously. For this reason, both *trustworthiness criteria* and *verification criteria* would be considered during the four stages of the study and used as the tools to make corrections of methods.

Chapter 6

Search for Evidence of Pattern-like Phenomena

Early stage research typically engages three fundamental steps [59]:

- define the problem to be understood, key concepts and related literature (*what is the problem*)
- conduct a pilot study to demonstrate others' interests on the question (*why it is important*)
- conduct a formal study to gather evidence of research question (*is it a real problem*)

I have discussed my conceptual understanding of the design process, design activity, parametric design systems, design patterns and related literature in Chapters Two, Three, and Four. Here I describe the next phase of my research: a pilot study to evaluate the interest of practitioners in the proposed study.

6.1 How do Others Recognize Importance of Patterns?

Research literature has provided evidence of the existence of patterns in design domains, including architecture and urban design [13], software engineering [83], interior design[239], human- computer interaction [230], computer-supported cooperative work [136], interaction design [30], education science [25, 225], information visualization [90], and communication [73]. Do phenomena suitably described by parametric design patterns exist?

Alexander argues for the existence of patterns in his earliest notes on design [8]. It took him four years to translate the notion of "subsystem" to pattern [12]. Although Gamma et al. [83] did not conduct formal studies to substantiate the wider existence of software patterns, their observations of expert practitioners provided compelling evidence. Object-oriented programming emerged in the 1960s as a means to improve the modularity and re-usability of software. Gamma et al. noted that, "one thing expert designers know not to do is solve every problem from first principles. Rather, they reuse solutions that have worked for them in the past. When they find a good solution, they use it again and again. Such experience is part of what makes them experts." Similarly, Jenifer Tidwell recognized UI interaction patterns in her observation of users: "The applications that are easy to use are designed to be familiar. As long as the parts are recognizable enough, and the relationships among the parts are clear, then people can apply their previous knowledge to a novel interface and figure it out [230]." Alexander discovered evidence of the existence of patterns in his design notes. Gamma et al. and Tidwell found evidence from their observations of existing projects. However, the process of how such evidence should be found was not discussed in their publications.

When a group of people start to adapt a new system, there will be many conversations about how that tool can and should be used. In those conversations, information flows between the tool design and practice. It is through this discourse that users are able to learn. I am confident that this same process of development will occur with parametric CAD system GC as well. In my pilot study, I seek examples of such conversations that will provide insight into the actual dynamic of the adaption process.

6.2 Pilot Study to Investigate Pattern-like Phenomena

In 2007 SmartGeometry workshop, I invited a group of experienced GC users to conduct the pilot study. The study proceeded from the following working hypotheses:

1. According to their teaching and working experience with the parametric CAD system GC, there exist some pattern-like phenomena.

2. There are needs to transfer successful design practice into a form of knowledge that can aid practice and learning.

6.2.1 Research Methods

For a study seeking evidence of design patterns in a new professional domain, I can choose from some simple methods to gather evidence of pattern-like phenomena, as well as the audience's opinions and interests. Some popular methods researchers use includes conducting physical or online surveys, or face-to-face interviews. The purpose of a survey is to generalize from a sample to a population so that inferences can be made about the phenomena in question, or attitude of the population [19]. For example, in the survey's questionnaire, some central questions focus on whether the subject has ever found a solution for a certain kind of problem and reused it with any frequency. However, such a question cannot be asked directly because the subject usually has the intention to satisfy the researcher [44]. To avoid the internal validity issues, it should instead be mixed with other superficial but related questions such as: "have you encountered any problems in the process?", "what kind of problems?", and "what kind of solution did you use?" Because I am able to only survey or interview a small representative sub-population, the study does not provide strong external validity - that is, the ability to be generalized or transferred directly. However, the current aim is to establish compelling evidence for the existence of pattern-like phenomenon. For such a purpose, this approach is acceptable.

An interview can typically elicit more information than a questionnaire and reach a deeper level of details, providing insight into the subjective reactions, opinions, and motivations around individual reasoning. I have opted to conduct this study through a series of open-ended conversational style interviews. During the interview, the same questions are presented to interviewees. The questions remain open and adaptable to the flow of conversation and the interviewee's interests and dispositions. According to Lisa Ede, this approach facilitates faster interviews that can be more easily analyzed and compared [77].

The interview process is as follows:

- Get the interviewees involved in the interview as soon as possible.
- Before asking about controversial matters, first ask about some facts, such as their occupation, type of professional, experience with parametric modeling tools, and their role in the current event.
- Intersperse fact-based questions throughout the interview. To avoid long lists of fact-based questions, I ask about current problems encountered during the tutoring and

possible solutions.

- Ask questions about the present before questions about the past or future. After discussing possible solutions, inquire whether they have used the same strategies to solve other problems before. Ask about their experience in learning from others and what kind of strategy references they were looking forward to have.
- Conclude with a question that enables the interviewee to provide any other information they prefer to share and their impressions of the interview process.

6.2.2 Role of the Researcher

As a researcher, I played two roles in the pilot study: interviewer and data analyst. I began by recruiting and interviewing participants with open-ended questions, and investigating idioms of use from their experience with GC. During the interview, I tried not to lead the conversation, and exercised caution when the interviewee appeared to be echoing similar thoughts. Such an interview enables the researcher to simply "tell it like it is". Data analysis occurs in a bottom-up process. I searched for information that around which stories could be woven. However, merely depending on the interview data, it is hard to reveal a complete and whole story about how a pattern idea was raised and developed.

6.2.3 Research Subjects: Local Developers

I invited the GC software's chief designer Dr. Robert Aish (RA) and five workshop tutors, Jeroen Coenders (JC), Roland Hudson (RH), Axel Kilian (AK), Steven Sanderson (SS), and Rob Woodbury (RW), as the interviewees. These individuals have expertise in architecture or civil engineering design domains, in addition to the use of parametric modeling software. In the investigation of cooperation among CAD users, Nardi and Miller [161] identified a continuum three kinds of users: end users, local developers, and professional programmers. Local developers serve as a resource for end users, training them, developing code for their work. Having broader and deeper knowledge, programmers contribute code to the the programs of end users and local developers, and help them learn new things. According to this user continuum, RA is the "programmer" and five workshop tutors are professional "local developers".

Although their expertise and experience can not be generalized to the larger user population, each of them has tutored tens or even hundreds of new users in the last few years. In each of their own design or tutoring processes, they have encountered many different kinds of problems. Their answers about whether or not having used the same strategies to solve similar kinds of problems are great evidence for pattern's existence.

6.2.4 Data Collection

I conducted the interview study at the SmartGeometry 2007 GC winter workshop. The workshop took four days (January 26-29) at the Hudson Hotel in New York City. Nineteen experienced GC tutors were invited to teach the workshop. During the four days, tutors worked intensively with students (mostly professional architects or civil engineers) on individual projects in assigned rooms. During coffee breaks I was able to individually interview five tutors based on their availability. We usually found a quiet corner in the lobby to avoid possible interruptions. After introducing the purpose of the study and asking them to sign the consent form, I asked prepared questions and recorded the conversation with a digital voice recorder. The length of the conversation depended on the flow of topics and varied from 15 to 30 minutes. After a conversation, I reviewed the recording immediately. If there was any discussion point remained confused or undefined, I would find the interviewee again and reviewed the confused section with him.

6.2.5 Data Analysis

Researchers need to tailor the data analysis beyond more generic approaches to specific types of qualitative research strategies [59]. For this pilot interview study, data analysis was aided by a qualitative data analysis application ATLAS.ti [18]. The software helps researchers uncover and systematically analyze complex phenomena hidden in text and multimedia data. Phenomenological research uses the analysis of significant statements, the generation of meaning units, and the development of an 'essence' description [154]. The study also pays attention to significant statements to weave up stories of idioms of use.

Without transcribing all conversation recordings, I anchored the useful pieces of audio data and coded them to identify significant ideas. For example, shown by Figure 6.3, I firstly marked different useful pieces from tutor SS's interview recording into *quotes*, and assigned those quotes with meaningful *codes*, such as 'anchor the startpoint', 'reference lines', and 'clean the script'. After coding all the recordings, I have generated more than fifty codes. I use the network manager to import related codes and assign links (such as 'is associated with', 'is a part of', or 'is a cause of') among them. The semantic layout arrange the locations of nodes in the window and shows me the flow and structure of ideas. From the right window in Figure 6.1, the cluster of nodes shows me the idiom of use "position anchor". In Table 6.1, I listed all the seven node-link clusters I found from my six-interview recording data and compared their strength of arguments. The four idioms of use are reported in this chapter based on the data analysis.

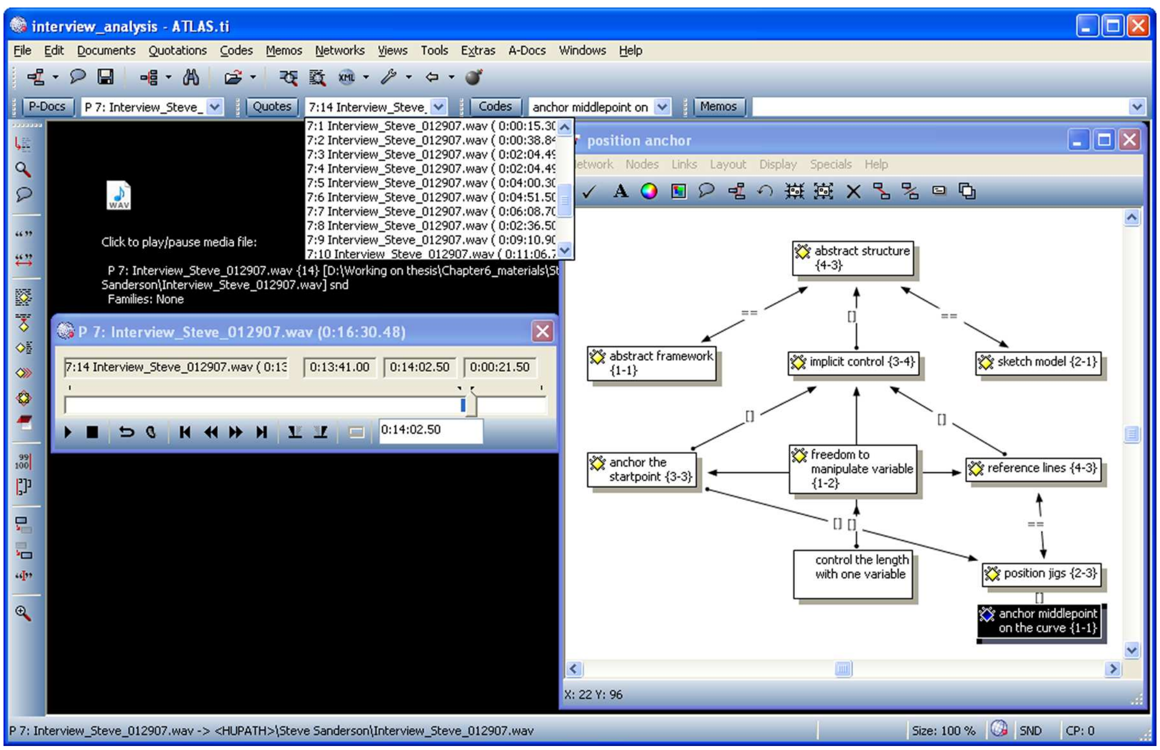

Figure 6.1: Screenshot of Interview Coding Process in ATLAS.ti.

However, during the analysis process I realized that I could not fully rely on ATLAS.ti to give me the insights. It helps to automate and speed up the code process, and provide a more complex way of looking at data. But in many cases, the long list of codes and messy node-link views did lead to confusions (only six recordings generated more than fifty codes). To discuss the pros and cons of computer assisted qualitative data analysis software, John

Seidel argues that there are potential worries of using such tools [204, p.110-113]:

1. it will distance people from their data.
2. it may lead to qualitative data being analyzed quantitatively.
3. it may lead to increasing homogeneity in method of data analysis.
4. it might be a monster and hi-jack the analysis.

I agree with these potential worries. So, I not only pay attention to the selected audio segments, but also listen to their context in the original recordings to avoid misunderstanding of data. In presenting the outcomes in this study, I used *thick description* to describe the idioms of use. According to Clifford Geertz [84], in order to explain the human behavior clearly to an outsider, the researcher should also describe its context such that the behavior becomes meaningful. *Thick description* is a necessary strategy to ensure the trustworthiness criteria *transferability* [93].

6.2.6 Validity Verification

In order to ensure rigor, interview recordings were checked immediately after interviews to form a mutual interaction between myself and the interviewee to make sure both sides are clear with the conversation. Where needed, I employed *post-observation inquiry* or *artifact walkthrough* to review the conversation with the interviewee. This inquiry process is a form of member checking intended to ensure credibility of data. For example, tutor JC said "I would usually program with GC" in the conversation. I was confused whether he wrote scripts in GC or just used GC to run his external code. After checking with him, he made it clearly, "no, I do not use GC's interface because I found myself limited by it. I simply program in C# and then attach the code to GC."

The sample population of this study is very particular (only local developers and the software chief developer, without any end user). Consequently, there may be questions about trustworthiness transferability. In light of the context of this study, I argue that it is not necessarily for the outcome to be transferrable to other populations. It is a learning process from expert users to look for the existence of idioms of use. Searching for commonality in participants' work experience is not the goal of the study at this early stage.

6.3 Pattern-like Phenomena - Idioms of Use

During the interview, RA (chief designer of GC) conveyed the background story of the software: GC was developed in the 1980s as ship building software, then later to design free form architectural shapes in the 1990s. Holding one of the earliest HCI PhD degrees, RA integrated his knowledge about CAD system, software usability, and information visualization into a software prototype of GC in 1996. He stated that after he started to work for Bentley, he had the opportunity to "really start from ground up, to build all the foundation tools to essentially understand what were important and what were missing." He built a number of prototypes, and one of them was called CustomObjects. The CustomObjects prototype was publicly presented for the first time in 2002 at a Bentley systems conference, with its implemented projects from design firms like Foster and Partners, KPF and institutions like the University of Waterloo.

In 2003, the SmartGeometry Group began hosting workshops and put GenerativeComponents (renamed from CustomObjects) to the test. The innovative technology displayed at this conference earned such praise that it became a part of MicroStation™. Bentley introduced new parametric capabilities aimed at enabling and encapsulating design intent. I can see the incremental evolution of GC from the past six years. During my interview with RA, he explained that "my new functions not just came from my own mind. Definitely not, no, actually more from my users, they asked me all kinds of questions, and new ideas came from them as well. I tried to supply new functions to support their needs." Here, I gathered four stories of users' idioms of use. Some of them have grown to be the current functions/features in the system.

6.3.1 Separate Control

RA mentioned three principle areas of interests for enhanced parametric technologies: parametric components, parametric assemblies, and parametric controls. *Parametric components* enable the definition of characteristics, constraints, options, and relationships within building components. *Parametric assemblies* enable the definition of the configuration, options, and relationships within an assembly of building components. More importantly, *parametric controls* enable the manipulation of parameters based on robust design rules and formulas. This contributing technology is essentially focused on the use of rules to create perimetrically derived geometry. The resulting components are generated by processing the defined rules to control their creation, characteristics, sizes, and placement. They are generally at a higher order of complexity than the parameters within any given traditional assembly.

Parametric controls enables new means of form finding, and have been applied to generate and manipulate geometry. An astounding range and complexity of form can be quickly explored, refined, and revisited easily within a 3D model. This is one of the key qualities GC users first realized. Early in 2003, tutor RW found that some users tried to control a piece of model through sliding a point along a separate line segment. The point's position on the segment became a controlling parameter. Users can imagine such a control parameter as driven by virtually any information resource, for example, as in response to solar, wind or acoustic analysis, building program requirements, etc. Then he can use this control parameter to design the geometry, build evaluative tools, or integrate with digital fabrication. However, when the model gets more complex, the simple control model starts to be insufficient.

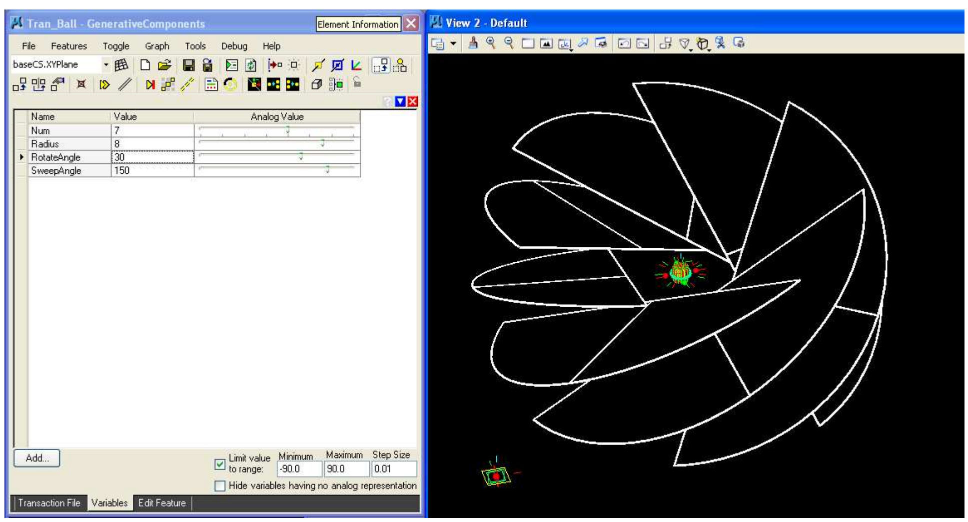

Figure 6.2: Demo of Using Four Graph Variables to Control a Folding Ball.

RW noted that a new function of *Global Variables* was added to GC before the 2004 workshop. Global Variables work in a hierarchical situation. A variable directly applies a number to control all elements or constructions that reference to it. In Figure 6.2, there are four variables (Num, Radius, RotateAngle, and SweepAngle) to control the number of shutters, size of the circle, vertical rotation angle of shutters, and sweep angle of bow shapes in a folding ball. Without direct manipulating the model in the right model window, users can slide the anchors on the "Analog Value" bars to control the transforming condition of the model. Later on, Global Variable was renamed as "Graph Variable" in the software. As soon as the variable function was provided in 2003, users dropped their own sliding tools and adopted the variable function to explore design variations quickly and effectively.

In 2004, RA presented a new function called *Law Curve* during a SmartGeometry workshop. A law curve is a curve (can be defined by control points, functions or other means) inside a rectangle frame (called Law Curve Frame in GC). Any point along the curve has both X and Y coordinates. Given some X values, mapping to the curve, I can get the corresponding Y values. Thus the curve defines the relation between value X and Y. The X coordinate is the independent variable, and the Y coordinate is the dependant variable (based on its given X and the curve). Through such a curve, we can generate a series of (X,Y) value pairs by giving a series of X. Then these (X,Y) pairs can be used to control the model.

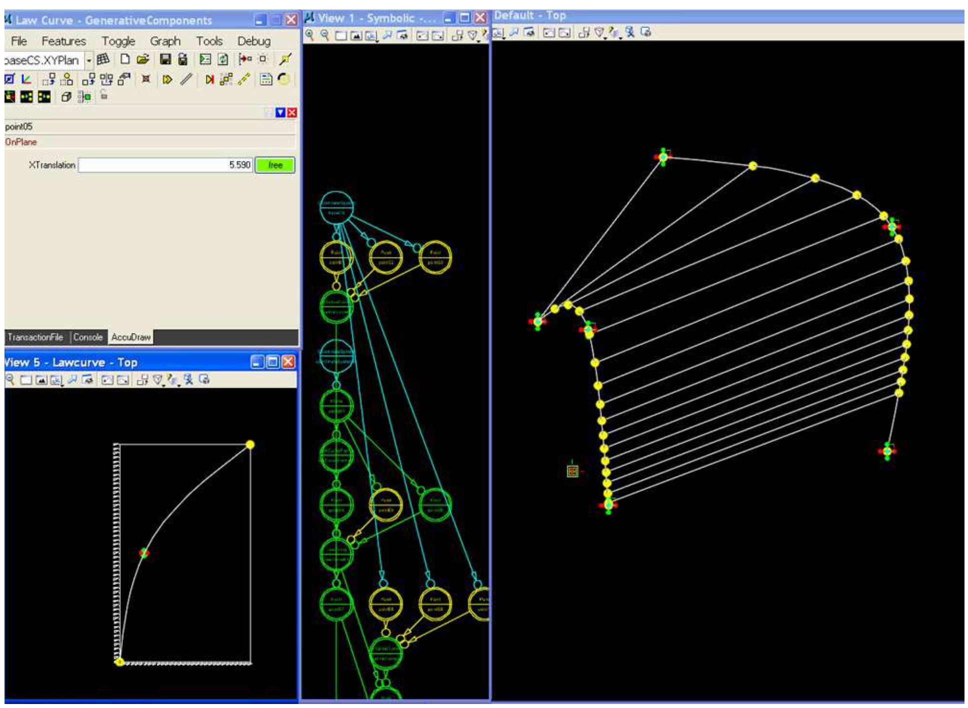

Figure 6.3: Demo of Using a Law Curve to Control a BSpline Surface.

As depicted in Figure 6.3, a Law Curve controls the distribution of points along the curve on the right side. The points along the left curve are evenly distributed. When I adjust the shape of Law Curve in the left window, in the right model window the lacing lines between the two point groups move and adjust. The Law Curve function can be used to control parts of properties in more complicated models. Tutor RH mentioned that whenever the tutor demonstrated this function, workshop participants usually were immediately attracted and tried to integrate it into their models. However, the Law Curve was very difficult to use (although this function becomes easier in later versions of GC). After playing with it for a while, users usually would go back to define their own controllers. As such, we can see that users will give up the system predefined controllers when the controlling mechanism is too complicated to manage.

Tutor RW noted that "We saw them throw out small controlling components all the time. The controlling components they created and used in one model have not been nicely separated from the main parts and can't be reused easily later. Many times, they just keep on reinventing the wheel."

6.3.2 Compiled Features

A very common situation in architectural modeling entails arraying elements across a surface or along a collection of components (e.g. points, curves, and polygons) to form a complex structure. In parametric modeling, users can vary the inputs of the element to make variations of the element. Duplicating such a element and change its shape accordingly can be an effective strategy to build a complex structure. These elements may vary by their absolute position and their spatial relationships with neighborhood repeating elements. Encapsulating such an element properly is an important skill in parametric CAD systems. When CustomObjects was first presented to users, the only way to duplicate a parametric module was through replication. This method is quite limited when we meet the following problem like architect Shane Burger. In 2004 SmartGeometry workshop, he tried to apply a parametric structure onto each triangle on the surface of the dome (Figure 6.4). Instead of applying the structure to all the triangles as a whole, he had to do it manually one by one.

Thus there was a need of a new feature. This feature should work like a "function" in traditional software engineering, but in the 3D graphic format - with a given input, the feature reflects the corresponding geometry. Users should be able to create a separate 3D

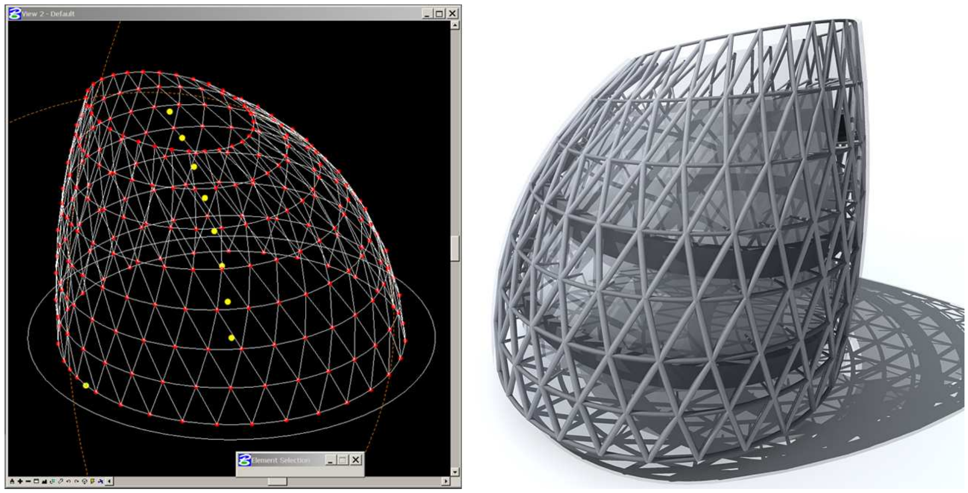

Figure 6.4: Dome Surface Made by Shane Burger to Apply Parametric Modules.

reference model, define some key components (such as a polygon or several variables) as the input, and reference this model freely. By feeding a series of inputs into this model, the user can replicate (or array) the model in one step to create a complex structure with repetitive components. Here the notion of compiled feature arise. As RW recalled, "it was in the 2003 event, the compiled feature idea first came up." Tutor AK also confirmed this fact. He added that when RW and he presented the first version of compiled feature to the group, people in the group didn't get it. When this function appeared in 2003 SmartGeometry workshop, it took a long time for users to understand the steps involved:

> "You have to have a feature working through a single object - like a shape. Everything have to work through that shape. Then you provide the shape as an argument to your feature. If you provide a collections of shapes, and then the feature is put on to each shape."

RA explained the purpose of the function in his interview:

> "You define some intermediate objects which you can populate in your context, and you can create a feature on the local context which you can subtract. Then, because there is a matching between local template in your feature creation and the components in your context, you are able to bridge them though complying a feature."

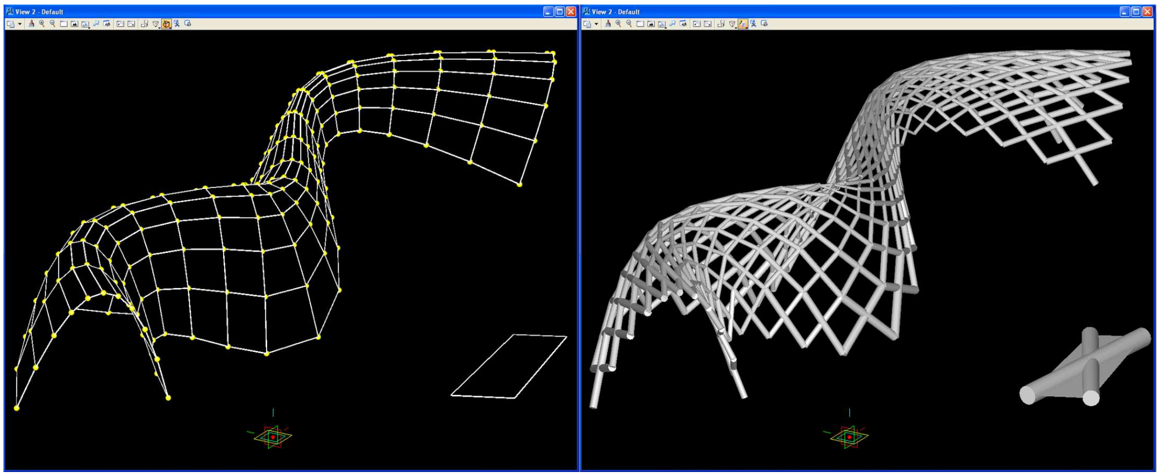

Figure 6.5: Demo of Applying a Cross-cones Feature onto a Collection of Polygons.

AK came up with a demo tutorial of using the new function in 2005. The demonstration (Figure 6.5) displays how to compile a cross-cones feature with a single input of a polygon and apply this feature to a collection of polygons in one step. It was in the 2006 Cambridge workshop, this strategy became a really normal technique for the group to do. Between 2004 and 2006 workshops, the idiom of use took two years to become normalized in the GC user group.

Typically a GC component has multiple inputs, so customizing each one needs a lot of work. This idiom of use enables users to reorganize multiple inputs of a model through a smaller number (preferable one) of abstract proxy objects. Thus make the modeling work more efficient and easier to control.

6.3.3 Position Anchor

In drawing tools such as Adobe Photoshop, rulers and guides are available as references to aid the user in positioning and locating objects. In CAD systems such as Autodesk AutoCAD, we frequently use offset lines and curves to anchor the positions of components. This *anchoring* strategy has also been transferred into parametric modeling environments. Tutor RH said,

> "our students start to build simple abstract frameworks to isolate structure and location from geometric details. I think they obtain such skills from previous

working experience in CAD systems. What we are trying to teach is making the anchors smarter. I mean, they can not only serve as construction lines to locate elements, but can also link more controls to enrich the parametric model."

The curved roof demonstration provided by AK at the 2005 workshop includes a good example of a smart position anchor (Figure 6.6). In order to build a BSplineSurface along a curve, AK assigned a series of coordinate systems by parameters along the curve, used these coordinate systems as anchors to draw connecting vertical and horizonal line segments, and then took the endpoints of these line segments as anchors to apply the surface. These coordinate systems preserve the controls of distribution along the curve, and line segments keep the controls of height and width of the surface. For the surface, they are abstract frameworks, but more importantly, they are necessary steps for constructing the parametric model. As AK mentioned during the interview,

> "You have to build the simple version first, it allows you to understand how the controls work without the distracting detail and slow interaction implied by a larger model. In comparable complex models, you can also change it easily. Many people understand why to do it, but take sometime to learn how to do it in a better way."

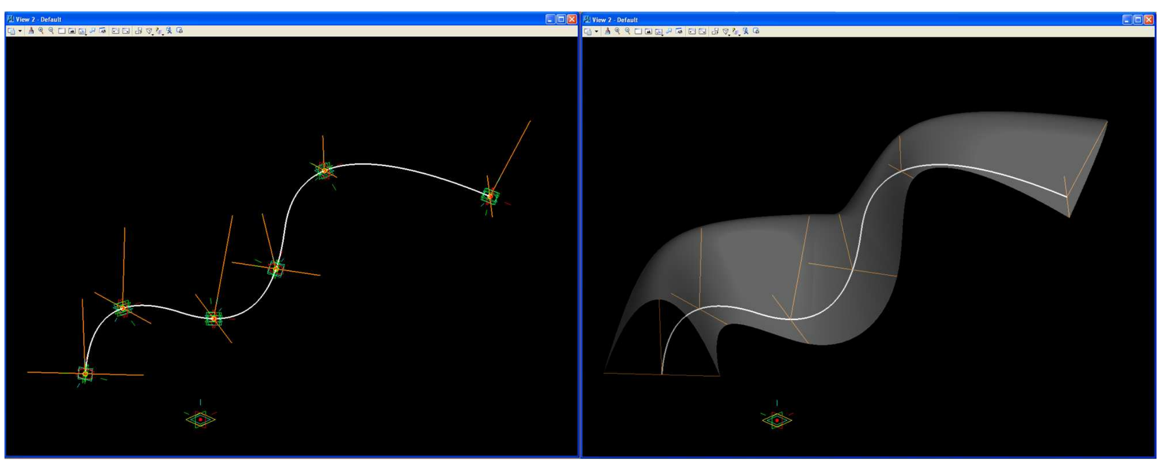

Figure 6.6: Demo of Using Endpoints of Lines to Anchor a BSplineSurface.

6.3.4 Two Dimensional Collections

Many designed artifacts have repeating elements. These elements are anchored on a set of point grids. Examples of such a grid can be points along a curve, a 2D grid on a surface, or a 3D grid in a volume. Beyond such a point grid, GC can also support a grid of line, surface, volume, and even models. Users need to know how to effectively manage such grids.

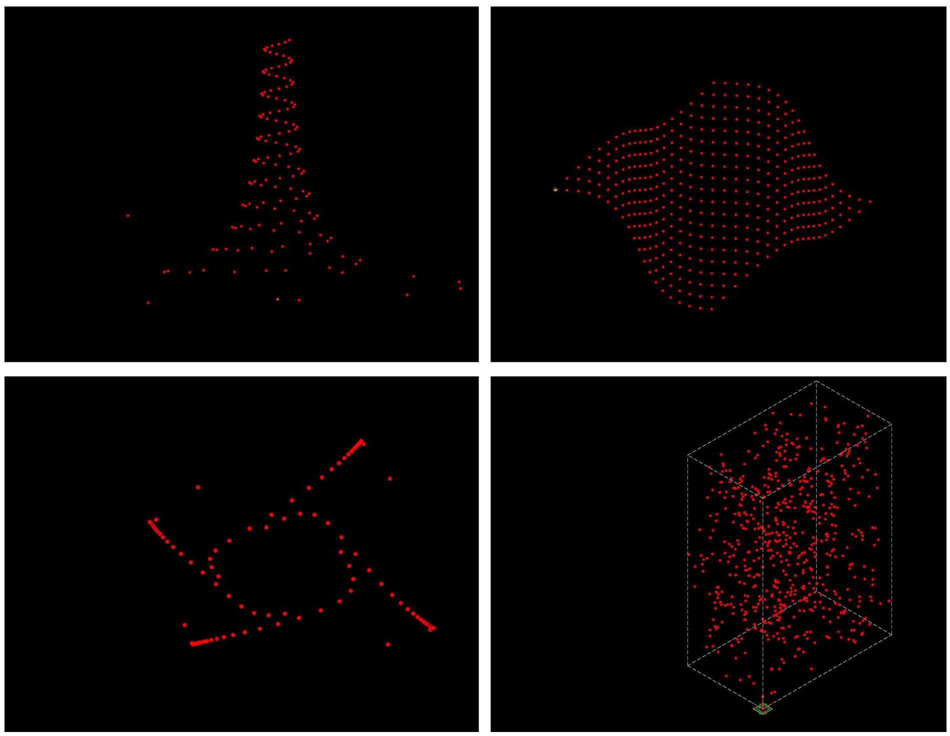

Figure 6.7: A Collection of Points with the Same Logic is Represented as One Point.

During the interview with RA, I talked about the scalability issue of the GC symbolic model and stated that if there are more than 30 symbols in the graph, users would find it difficult to read and understand the relational structure. RA questioned me back:

> "What you are saying is that the overload of symbolic model is actually telling you something... telling you that you need to build structure. Maybe you need to compile a new feature to take a model and encapsulate it into one or two new components. Or, you replicate it, for example, a thousand points on one facade, but logically they are only one idea. If you got a hundred nodes into

> top level graph, then you haven't properly structure your work and you haven't done proper encapsulation. It is telling you that the system is quite happy to deal with the complexity, are you happy to deal with such complexity?"

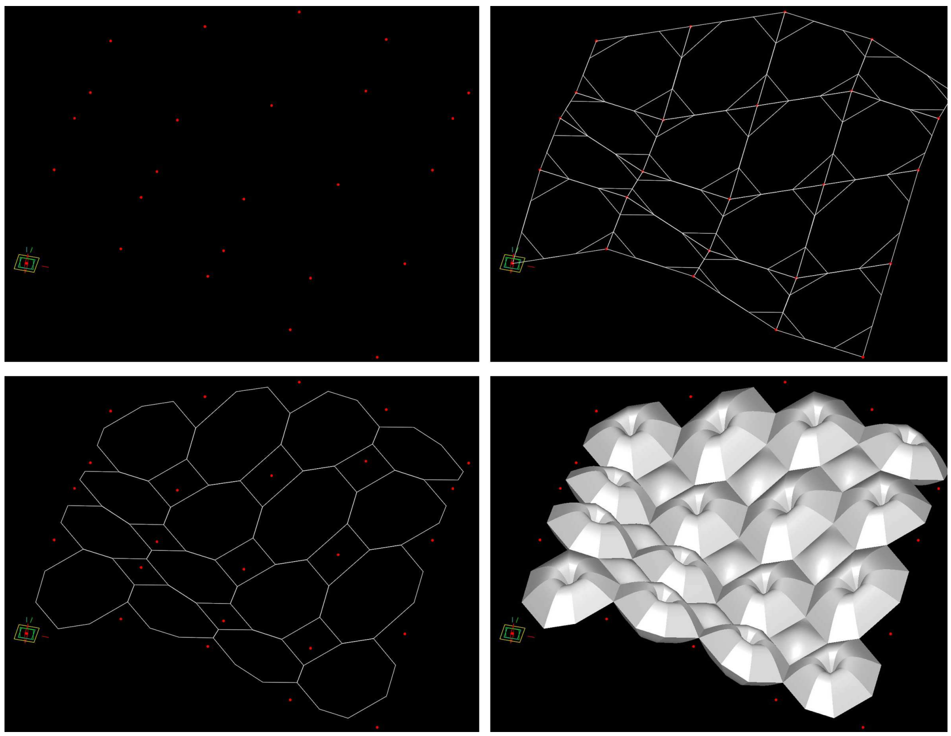

Figure 6.8: Demo of Using Organized Points to Locate Repeating Elements.

As RA said, considering that geometric logic underlines the size and location of a group of points, in a lot of cases, we can use one point (scripted in 1D, 2D or 3D array) to represent the whole collection. Figure 6.7 demonstrates four examples of what one point in GC can look like.

These kinds of point collections can be used to locate repeating elements. Tutor SS described that,

> "the issue I encountered more often is the lack of knowledge in geometric structures. If students have any experience with Catia or SolidWorks, there will be some commonalities they can latch on to. If people are very very good Rhino

> users, they can learn the geometric model of GC a little bit more easily. There are some principles people are lacking in some ways. The most frequent thing I am doing again and again is writing simple two dimension collections. They can't rely on the functionality of GC. They need to do something a little more customized defined for their own projects. I've written that properly five or six times in my group now. Dealing with the 2D collection of points to anchor the location of repeating objects is the major thing I am doing here."

Another tutor JC also said that the most frequent help he provided to the students was, "looping through a point set and selecting stuff from that based on some conditions, then do further construction with the selected result."

In Figure 6.8, a point grid (top right in Figure 6.7) can be used to locate complex repeating elements. In this example, the collection of points was organized to capture intended spatial relationships, and simplify the process of continued model development. This saves time and effort in both modeling and reuse of a model in new context.

6.4 Summary

Through analyzing the interview conversations with the GC chief designer RA and five expert tutors (AK, JC, RH, RW, and SS) , I am able to filter out a list of idioms of use. Some of them have been stated again and again by several different interviewees and accompanied with solid examples. Some are quite abstract at the conceptual level. In Table 6.1, I list these idioms and compare their strength of being reused according to subjects' statements. In above, I selected four idioms of use and explained their stories with examples. In the next part of study, I find other evidence to support these four ideas and finally encapsulate them into four parametric design patterns. In the right column I list the names of patterns implied by the idioms of use.

The intention of this pilot study is to search for first evidence of pattern-like phenomena. From the interview data it is hard to find established or well-identified pattern ideas. Instead, I found different idioms of use with contextual information and solid examples. Focusing on four pattern-like ideas has provided enough evidence for me to proceed to the next level. From the interview, the chief software designer also corroborated the existence of such pattern-like phenomenon. Since he himself has the greatest insight into the software design, his message is encouraging for this research.

Idioms of Use	**Tutors**	**Strength**	**Implied Pattern**
Compiled feature	RA, AK, and RW	Strong	Place Holder
Integrate with other software	JC	Weak	
Position anchor	AK, RW, and RH	Strong	Jig
Separate control	RA, RW, and RH	Medium	Controller
Simplify and rebuild the model	RH, SS, and RA	Medium	
Two dimension collections	SS, JC, and RA	Medium	Jig and Organized Collection of Points
Use symbolic model to understand the geometry	RH and JC	Weak	

Table 6.1: Conclusion of Idioms of Use from Tutor Interviews.

Finally, I also found clear evidence in the data that tutors, as local developers, were unsatisfied with limited resources and current methods to teach GC in workshops. JC thought students were lacking programming skills. SS said that because a lot of users had never tried other parametric modeling tools, it was challenging for them to understand the geometric structures. RH also stated the same problem when he had to teach a small group of students (five late comers) to catch up with the larger team. During the four-day workshop after the pre-training tutorial, most tutors found themselves busy solving similar problems for their group members. They tried to reuse the code they wrote, but sometimes the models are too specific, the code is hard to transfer from one project to another directly.

From the conversations, nearly all the tutors looked forward to having better organized tutorial materials with sufficient examples to learn, refer, and teach. This result answers my second hypothesis about these experts' needs and expectations on pattern-like support.

At the next stage, I will leave the system developer and tutor group, and start to participate in workshop student groups to observe what exactly are taking place in their design and learning practice.

Chapter 7

Gather and Elicit Pattern-like Thinking

According John Creswell's three step theory: *"what is the problem"*, *"why it is important"*, and *"is it a real problem"* [59], I construct this thesis in a narrative structure. The previous chapters review the meaning of key concepts and their related literature scope (*what is the problem*) and introduce a pilot study to demonstrate GC developer and professional tutors' interests on the research question (*why it is important*). In this chapter, I will introduce a series of participant observation studies to search for evidence on pattern-like thinking as the process to understand the real problem (*is it a real problem*).

7.1 How do Others Elicit Pattern-like Thinking?

How do patterns arise? Authors in different research disciplines locate patterns in different ways:

- *Urban Architecture Patterns*: Alexander stressed that patterns should serve urban inhabitants activities because, "towns and buildings will not become alive unless they are made by all the people in society." His system of 253-patterns was sourced primarily from his own group's architectural experience. Ideally, they should have surveyed and searched for patterns more broadly in the urban design industry, to make it more general to meet users' needs under different circumstances.

- *OOP Patterns*: Gamma et al. started to search for patterns from the earliest object-oriented programming language Smalltalk. In the Model/View/Controller(MVC) triad of classes, they were able to figure out patterns such as Observer, Controller, Strategy, Factory Method and Composite [83, p.4-6]. It appears that they use existing OOP languages and, perhaps, their own projects as pattern sources.

- *UI Patterns*: Apart from her pattern book "Designing Interfaces", Jenifer Tidwell has two websites to publish her UI patterns to support the web application design - a sandbox site and an official site. Tidwell [230] does not describe how she gathers and confirms the data to design patterns, but I can see new pattern titles emerging first in a grey color and then being finalized gradually in the sandbox site. It seems that Tidwell built up her patterns by frequent observation of various GUI examples.

- *Pedagogical Patterns*: In a recent project report, Bergin et al. introduce patterns dealing with the diversity of instructional techniques [25]. Their website describes tens of pedagogical patterns grouped according to course sequence. In examining their website, I discerned that some patterns are related to a pattern-writing workshop in Vienna or some existing literature.

- *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 [202]. The patterns authors are either scholars from universities or members of CPSR (Communication Public Sphere Project). Although the collection has not be published as a complete work, this pattern language is built upon literature references or review of real events.

Borchers compares different pattern languages and states that an important goal for any pattern design team is to capture the reasoning behind design decisions and experience from past projects, and to create a corporate memory of design knowledge. He emphasizes that [30], "ideal patterns should be the result of user experience." In the brief review above, most patterns have come from the authors' own experience or existing cases. But, what are the experiences? How was that experience captured? What kind of users were involved? Few authors describe the process of pattern development in detail. To achieve grounded evidence and trustful data, I propose the use of participant observation method and contextual inquiry techniques to identify pattern-like thinking in design practice.

7.2 Participant Observation to Gather Pattern-Like Thinking

7.2.1 Research Methods

Participant observation is one of the central research methods in cultural anthropology. This method includes information gained from participation and observation of the user through explicit recording and analysis of this information [16]. Immersion in the setting affords the researcher with the opportunity to hear, see, and experience reality as the subjects do. Ideally, immersion offers the opportunity to learn directly from one's own experience. Apart from passive observation, I integrated informal contextual inquiry interviews during the participation process.

Ethnographic interview examines the entire ethnographic research cycle from the perspective of interviewing [213]. It is important for a researcher to arrange good interviewing opportunities and present herself during the participant observation. There are basically two interview types: informal and formal. A formal interview occurs at an appointed time and results from a specific request to hold the interview (my pilot study used such an approach). An informal interview occurs whenever you ask someone a question during the course of participant observation [213, p.123]. Beyer and Holtzblatt describes contextual inquiry is an active interview form of study method [27]. In the current research stage, I employed some contextual inquiry interview techniques (such as work-based interview, post-observation inquiry, and artifact walkthrough [186]) in the participant observation study. The advantage is that the interviewer can communicate directly with the interviewee about her findings. However, it is very hard to create a situation that does not disturb the interviewee's work practice. As such, I limit the contextual inquiry process to a smaller, but valuable part of the whole study.

7.2.2 Role of the Researcher

Participant observation combines two different processes: observation and participation. Spradley's typology [214] models a continuum in the "degree of participation" for researchers. Per that typology, this research employs an *active participation* approach. As a researcher, I am 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 events, I acted as a tutor for designers (workshop subjects) while simultaneously observing and discussing their work with them. I provided immediate value through tutoring and added little overhead, as discussion and observation are an essentially part of what already happens in a tutoring session. I became familiar with this software more than three years ago. Having attended GC workshops since 2004 and started to tutor since 2005, I have sufficient knowledge of the application to provide help. I believe my understanding of the context and role enhances the awareness, knowledge and sensitivity to many decisions and issues that would be encountered.

However, as a consequence of previous working experience with GC, I also bring certain biases to this study. Although I have made every effort to ensure objectivity, my biases shape the way I view and understand the data collected and the way experiences are interpreted. For example, my personal experience and skills may interfere with how I understand a subject's problems and actions. For certain kinds of problems, I may already have a set of solutions in mind and thus ignore other alternatives. To avoid such validity issues, while chatting with subjects about their problems, I tried to make my advice suggestive instead of determinative.

7.2.3 Settings of Four Studies

I gathered data through a series of participant observation studies that were held along side four academic events:

1. SmartGeometry conference and workshop (Huston Hotel, New York, January 24-30, 2007)

2. CDRN parametric modeling and digital wood fabrication workshop and symposium (Department of Forest Sciences at the University of British Columbia, Vancouver, February 15-24, 2007)

3. Graduate parametric design studio course (School of Architecture and Landscape Architecture at the the University of British Columbia, Vancouver, January-April, 2007)

4. ACADIA software workshop: GenerativeComponents (Nova Scotia College of Art and Design, Halifax, October 2-3, 2007)

The third event is a graduate studio course in an architectural school. The other three events are all international conference workshops. The length of workshops varied from two to six days. I selected these four events to conduct my study because all the event subjects were professional designers. Attendees had all proposed projects to work on as part of the admission process, and intended to use GC to explore some aspect of that project. Conducting observation studies on prolonged design projects is very difficult. The short length of workshops and courses, and the desire to solve the design problem forced both the subjects and I to be both productive and innovative when we were encountering any design problems. During these events, I collected the following types of data: audio data (recordings of short conversations), graphic data (photos of paper sketches and physical models, model renderings, and related images, and screenshots captured from monitor), and text data (project proposals, GC model files and feature files).

For most of event participants, they have specific design problems that they think are solvable by GC. Especially in those workshops, project proposals were carefully reviewed by the workshop organizers to ensure the workshops can be useful and supportive for the designers and their projects. As a result, design practice in these events was partially driven by the nature of parametric modeling. In some cases, the use of parametric CAD system might affect the design practice, and as a consequence, affect the final design result. It is very hard to distinguish the system-driven effects from design-driven effects. However, I believe that designers do not design to satisfy a tool (although the tool might restrict or channel the design choices). Also, it is not necessarily be a bad thing that user channels his/her design by parametric modeling. Normally there are usually many stages and subgoals involved in the overall design process. GC will be only used to solve problems in some of these stages and subgoals. The data collected and analyzed in these events represent the stage of using the parametric modeling tool to explore and create some components in the overall project.

7.2.3.1 Study 1: SmartGeometry Workshop

The first study was conducted at the SmartGeometry Workshop on January 26-29, 2007. The workshop participants (more than 100, mostly architects and civil engineers) were self-selected into five groups. Every group occupied a separate workroom and was assigned three professional tutors. These participants had been competitively selected by the workshop organizers though adjudication of their project proposals. Prior to the workshop, they had all received no less than three days of training on the basic functions of GC. During the main workshop, all the participants worked intensively on their own laptops from morning till late at night.

In the beginning of the first day, I recruited five volunteer subjects through a public process. Among these five subjects, there were three females and two males, 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. The two students focused on academic studio designs. The other three used the workshop to solve design problems in their current commercial projects. All of them started to learn GC the prior year and it was the very first parametric modeling application they had ever learned.

Apart from three meals and the daily progress report sessions, the subjects spent most of their time working at their computer. 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. In order not to interrupt their thinking, I interviewed the subjects when they were taking a break or during moments of relative calm.

7.2.3.2 Study 2: CDRN Parametric Workshop

The 2007 Parametric Modeling and Digital Wood Fabrication Workshop and Symposium was organized and supported by the Canadian Design Research Network, the UBC School of Architecture and Landscape Architecture, the UBC Centre for Advanced Wood Processing and School of Interactive Arts and Technology at Simon Fraser University in Vancouver. This workshop lasted for ten days (February 15-24, 2007) and comprised three stages. During the first three days, participants took the GC tutorial and started to create parametric models that would later be fabricated in timber. Then in the following two days, they were taught how to use CADwork 3D [43] to design wood joints in their models and how to operate a Hundegger timber process machine to cut their model components from timber. The last five days were devoted to the process of digital model optimization and physical model fabrication and installation. There was an exhibition and reception arranged at the end of the workshop to present and celebrate their achievements.

Participants in the event worked in teams. I recruited three two-member teams (six individuals) as my subjects during the first day tutorial. One team was formed by an architecture school educator and a contemporary artist. The other two teams were organized by architecture graduate students. None of them had used a parametric modeling application before, and this workshop was their first introduction to GC. I collected data during the first four days while the subjects were using GC to develop their parametric models.

Participants were busy cutting and installing their models during the later five days.

7.2.3.3 Study 3: UBC Graduate Design Studio

Professor Oliver Lang invited me to teach an architecture graduate design studio in the 2007 Spring semester in the University of British Columbia. In this course, twelve students were assigned to use GC as a parametric tool to design a responsive facade for the UBC Center of Innovation and Technology. Although they had more than three months to work on the project, only one afternoon each week was available for them to focus on learning GC and to analyze the parametric components with me. In addition, they were also busy working on other parts of the center design and their regular semester course work.

I worked with the students for twelve afternoons that semester. During the first three afternoons, we walked through a number of examples and introduced the functions of GC. Later on, I started to work with students one by one to discuss their individual projects and worked on different problems in their models. I usually spent 20 minutes to half an hour with one student every afternoon. Data were collected during these sessions. Before the discussions, I interviewed them informally about their progress during the past week and any problems they encountered. Then, we usually focused on specific problems and started to explore solutions together.

7.2.3.4 Study 4: ACADIA Software Workshop

The Association for Computer-Aided Design in Architecture (ACADIA) together with CDRN held the ACADIA 2007 international conference in Halifax. The ACADIA software workshop was two days long, and was very intensive. Applicants submitted a statements of interest and project proposal to the workshop. Due to the high number of applications, the workshop organizers screened the submissions to make a highly selected group of attendees based on their proposed projects. Most attendees were architecture educators and professional designers. During the workshop, my senior supervisor Dr. Woodbury and I acted as tutors, taught attendees how to use GC and helped participants with their problems.

I recruited six subjects during the first day tutorial. Two were female, four were male. Apart from one graduate student and one architect, the other four were all architecture educators. The projects they tried to build and the current problems they met were mainly the topics during our discussion. I collected data during the whole two day workshop.

7.2.4 Data Collection

The first study in SmartGeometry workshop has two goals: First, it allows me to gain a preliminary understanding of how experts use and adapt these systems. Second, it helps me to understand the data structure and develop the "codable" moments for a thematic analysis [33]. The information from later three resources was used as the analyzing sources for data triangulation and supplements for pervious findings.

7.2.4.1 Process of Data Collection

I collected the subject's design work as complete as possible. These data includes (if there are): photos of physical models, digital and hand sketches, screenshots of the progress (to avoid disturbing their work, I sometimes used the camera to take photos), successful renderings, script segments, and the feature and transaction files at different stages (in Figure 7.1). Other than these visual materials, during the workshops and the course, with the subjects' permission, I 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 possible solutions towards those problems. The conversation length depended on the topics emerging during the process. These short conversations were recorded by a digital voice recorder. These conversations also made me clear on their design intentions and meaning of their sketches and models.

7.2.4.2 Ethical Consideration

I obtained ethics approval before I launched the study. Participant observation naturally invades the life of the subject [214] and sensitive information such as their roles and progress in real life projects are often revealed. I ensured that no personal information was collected. Subjects were interviewed separately and no information from the interviews was divulged to other subjects. Information was collected and saved in a secure place and subjects have full access to their own information. In the publications and reports, I only use materials they agreed to share.

7.2.5 Data Analysis

Most existing pattern languages do not describe the details of how evidence for the existence of patterns were gathered from practice. As a result, the process of eliciting pattern ideas

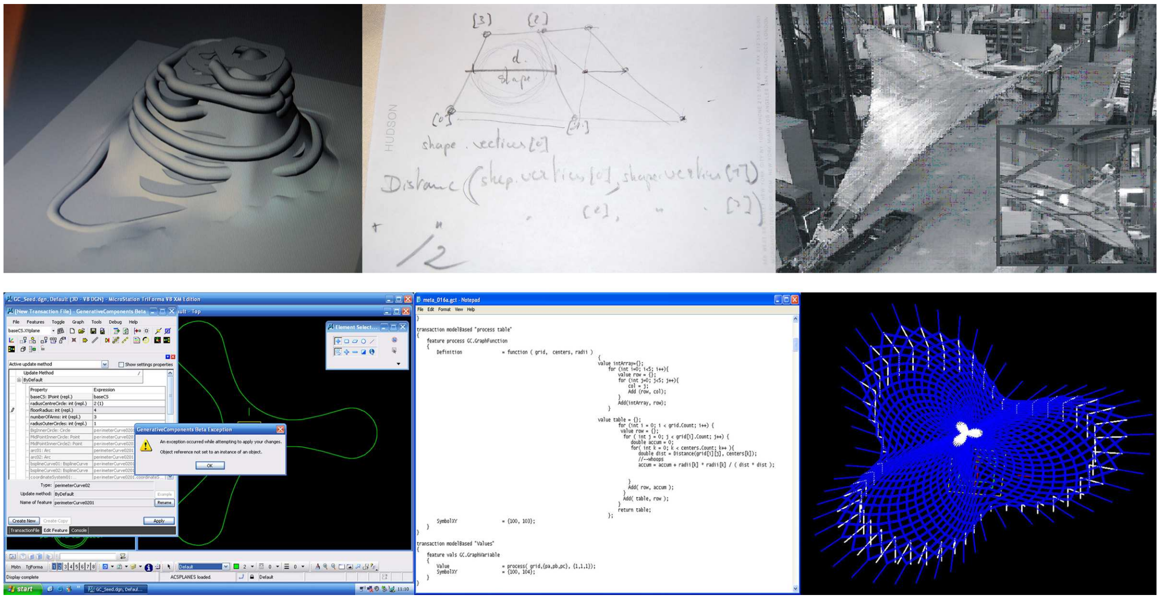

Figure 7.1: Primary Materials Collected from the 2007 SmartGeometry Workshop.

from grounding evidence is also absent in the work. To faithfully record the eliciting process of the parametric design pattern ideas, this study employs methods of thematic data analysis. Thematic analysis is a process for encoding qualitative information. The encoding requires an explicit ‘code’. This may be a list of themes; a complex model with themes, indicators, and qualifications that are causally related. Boyatzis states that [33, p.vii], “a theme is a pattern (this pattern is not the design pattern talked in this paper) found in the information that at the minimum describes and organizes possible observations or at the maximum interprets aspects of the phenomenon.” A good theme is one that captures the qualitative richness of the phenomenon. It should have five elements: a label, a definition of what the theme concerns, a description of how to know when the theme occurs, a description of many qualification or exclusion to the identification of the theme, and examples. Such a “thick description” would be sufficient to maintain the transferability criteria of trustworthiness.

Boyatzis also introduces three approaches to developing themes systematically: theory driven, prior data or prior research driven, and inductive [33, p.29]. Lacking a coding theory for patterns and prior sample research, this research uses an inductive approach (also called data-driven approach). I used ATLAS.ti again to aid the qualitative data analysis process. I anchored material, carried out a compare-and-contrast process to extract observable differences among samples. This process is called “immersion and crystallization” [147] and provides an inquiry audit path to maintain dependability in the study.

However, considering the potential worries of using such a research tool [204] and referring to my data analysis experience in the pilot study, I decided to be more cautious to name and assign new codes to the data. Instead of using very detailed code such as *anchor the startpoint* (in the pilot study), I assign more general code such as *rigid body*, *two dimensions of points*, and *increment change* to the data. I want to use more meaningful codes to help me understand the structure of ideas and figure out the idioms of use.

Transcriptions of important interviews, GC feature and transaction files, screenshots or photos taken during the modeling process, audio recordings of short conversations were imported into ATLAS.ti to create a hermeneutic unit that allowed analysis of each subject’s activities as a whole. I adopted the data-driven approach to develop thematic codes based on sampling and design issues [33]. Figure 7.2 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. Through images such as these, I was able to anchor codes such as *organized collection of points*, *penetrating*, *incremental change of size and shape*, *place holder* and *radiating direction*.

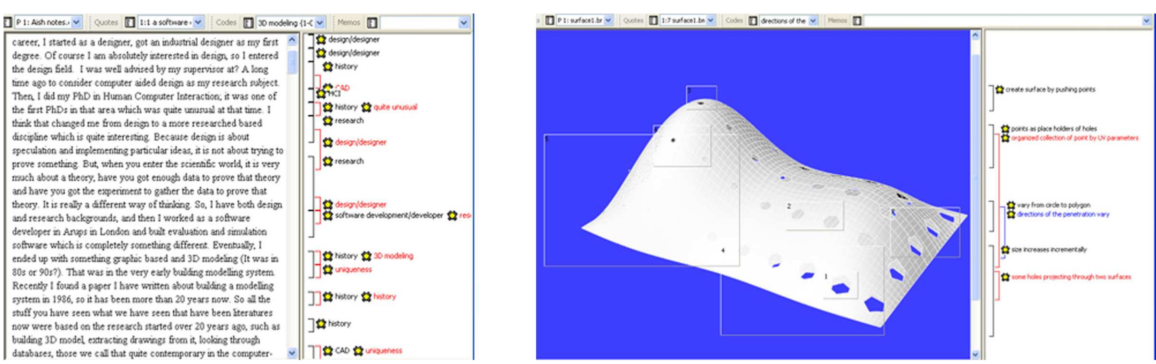

Figure 7.2: Screenshots of the Coding Process in ATLAS.ti.

By reviewing the codes and source materials repeatedly, I was able to locate important themes from the project, reduce the raw information, and generate some insights. Themes were identified as having potential relationships to pattern-related ideas. To determine such a code, we focused on comprehending what the problem was, how the subjects solve the problem and why she/he selected such a solution. In some cases, the problem was clear but the solutions are uncertain. Sometimes the solution has been presented, but the information about the reason to select such a solution was no explicit. The process of thematic analysis and code development helps us to filter out unnecessary information and get important ideas quickly. Themes and codes I created from this unit could be transferred and reused for analysis of subsequence subjects. While analyzing these five subjects, some pattern themes started to emerge.

Triangulation is a method used by researchers to evaluate and establish validity in studies. There are five types of triangulation: *data triangulation*, *investigator triangulation*, *theory triangulation*, *methodological triangulation*, and *environmental triangulation* [96]. Considering the nature of this study (hard to change theory, methodology or environment), we employed data triangulation and investigator triangulation by using different sources of data and inviting another researcher, Yingjie Chen, to code the same data. The data was analyzed through triangulation at two levels. First, because data was collected from different perspectives, such as model scripts, interviews and screenshots, and analyzed as a whole, that data has been naturally triangulated in itself. For example, in a short conversation I asked one subject how she was able to adjust the Z direction of points along a BsplineCurve based on another reference. Both her script file and verbal explanation corroborated the same answer. However, this kind of match does not apply to all the data. In some cases a subject's verbal explanation data do not match with his/her working data, which makes these data sections less useful. Second, to triangulate the findings, I collected data from another six subjects in the CDRN wood fabrication workshop, twelve architecture students in the UBC design studio, and six subjects in the ACADIA Halifax workshop. In the data analysis, we triangulated by looking for data and patterns across workshop groups.

7.2.6 Validity Verification

In this study, I used participant observation method and contextual inquiry techniques to search for grounding evidence for pattern-liking thinking. The four trustworthiness criteria were carefully considered as follows:

- *Credibility*: To conduct participant observation, the researcher must stay in the context to facilitate prolonged engagement in order to overcome the effects of misinformation and distortion [129]. For example, in this study, I worked and interacted with subjects for a long time. The first New York workshop was more than four full days. The second one was a ten-day workshop, and another observation session was a whole semester of design studio course in the architecture department of UBC. Such a prolonged engagement is one means to establish credibility. After each small inquiring conversation, I would ask for re-clarification, conclude the main ideas discussed, and ask for the subject's confirmation. Occurring continuously, both during the data collection and analysis stage, this is the process of member checking. Member checking could be formal and informal. It serves a number of functions, and is crucial for credibility [94, p.239].

- *Transferability*: Slightly different from generalizablility, the burden of proof for claimed transferability is on the subjects. To maintain the transferability of the study, I have provided, "an extensive and careful description of the time, the place, the context, the cultures in which hypotheses were found to be salient" [94, p.241]. This PhD research topic is very specific and observational circumstances are also unique, so I prepared thick descriptions for each of the observed projects.

- *Dependability*: Providing a path for inquiry audit is the method to maintain dependability. In this study, there are various kinds of data such as photos of physical models, digital sketches, and hand sketches, screenshots of existing problems, script segments and successful model, and audio recording of short conversations. All of this information has been saved in copies and preserved confidentially. Such rich information is sufficient for conducting the inquiry audit.

- *Conformability*: In this PhD research, each subject has his/her own project to work on and tries to use the new tool to achieve their goals. There is minor effect the researcher could make upon their work. Conversely, data across different observation events would be triangulated to analyze, so the conformability of study could be kept.

To ensure rigor, Morse et al.'s verification strategies emphasize that samples must be appropriate and data should be collected and analyzed concurrently [153]. I recruited the subjects from limited populations within a limited period of time, so it is hard to guarantee optimal quality data. However, because different forms of data (photos, images, screenshots, voice recording, and script files) were collected for each project, sufficient data was available to account for different aspects of the phenomenon of interest, which meets the requirement of *sampling adequacy* [152]. My data analysis was an on-going process during and after the events. After each conversation, I went back to my desk to download and organize the collected data immediately. I also went through the audio recordings rapidly to ensure there was no obvious misunderstanding in the conversation. When I met any unclarified ideas (sometimes a certain function or a piece of scripts), I wrote them down as notes and checked with the subject during the next conversation. Although it was hard to collect and analyze data at exactly the same, I made sure to conduct the member checking in the same day and initiated the data analysis immediately. Apart from searching for the most obvious ideas and getting member checking during the events, we revisited the data iteratively to search for pattern-like thinking and example ideas for pattern development.

7.3 Pattern Themes - Idioms of Use

In the following I introduce the projects and designers enrolled in the participant observation study. Projects are grouped by similarities, such as form or construction method. To ensure the privacy of individuals, they are identified by a two character abbreviation in the text.

7.3.1 Rigid Transformation

This first theme concerns changing an object's form by translating and/or rotating its rigid components. In physics, a rigid body is an idealization of a solid body of finite size in which deformation is neglected. Although most architectural structures are, in essence, rigid bodies as a whole, most have mechanisms within them - think of a door hinge. Parametric modeling allows designers to control and explore the discrete variation of components. It is very convenient to simulate the elasticity of membrane structure in GC. However, there were quite a few projects aiming to make approximate models of the motion of an assembly of rigid bodies. Such a design intent and similar strategies had been used in the following three projects.

7.3.1.1 LM in Study 3: Bending Sheet Metal

Gradate student LM in UBC design studio tried to simulate a kind of tensioned rigid structure - bending sheet metal. There is a cross cut in the center of a piece of metal sheet polygon, so the model was constructed by a set of four smaller polygons instead of a complete piece. In Figure 7.3, moving the central controlling point perpendicularly away from the polygon plan simulates a pulling force on the central edges of four separated pieces. For each piece, the other three corners remain fixed, so the shape starts to bend out of plan. The key issue is maintaining the lengths of each sides during the bending process, because of the rigidness of sheet iron.

LM noted a reverse thinking process while she was explaining the project to me,

> "Originally, I defined the four small pieces through *Polygon by Vertices*, I mean by the midpoints of the original polygon's four sides. Later on, I realized that after bending, the polygons turned into four BSplineSurfaces. I should use the *BSplineSurface from Boundary Curves* to define these four pieces. Thus, I had to define the bending boundary curves first. The startpoints and length of curves are fixed. The curving direction is partially determined by the controlling point. I should consider the controlling point, decide the curving tension, draw the curves, and then draw the BSplineSurfaces. It is a totally reversed process. I had to open a new file and rebuilt the model again."

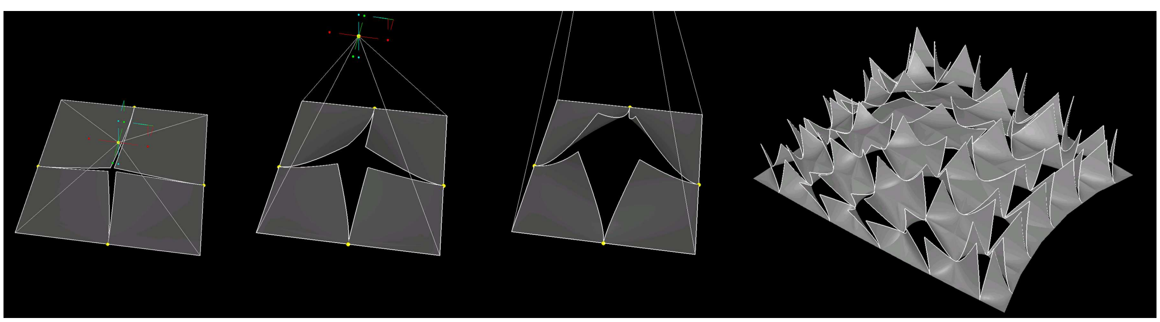

Figure 7.3: LM in Study 3: Bending Sheet Iron.

In LM's model, the sides maintain the same length during transformation. Figuring out the curved paths of the edges being pulling out is an essential process. She used the function *ByTangents* to draw the blending path curves so that she was able to adjust the rigidity through tangents later.

7.3.1.2 ZD in Study 1: Operable Shingled Shading System

In the SmartGeometry workshop, ZD planned to finish an operable singled shading system on an arc surface (Figure 7.4). He used his diagram (bottom left image) and sketches (upper left image) to show me that there are two stages of folding in his system: first, the arc panels organized by shingles can slide horizontally to fold together; second, a sequence of shingles organized a panel. Each shingle can both rotate and slide by the help of control arms to avoid friction and collision during the folding (right images). During the workshop, he was able to achieve the first stage (middle images) in GC.

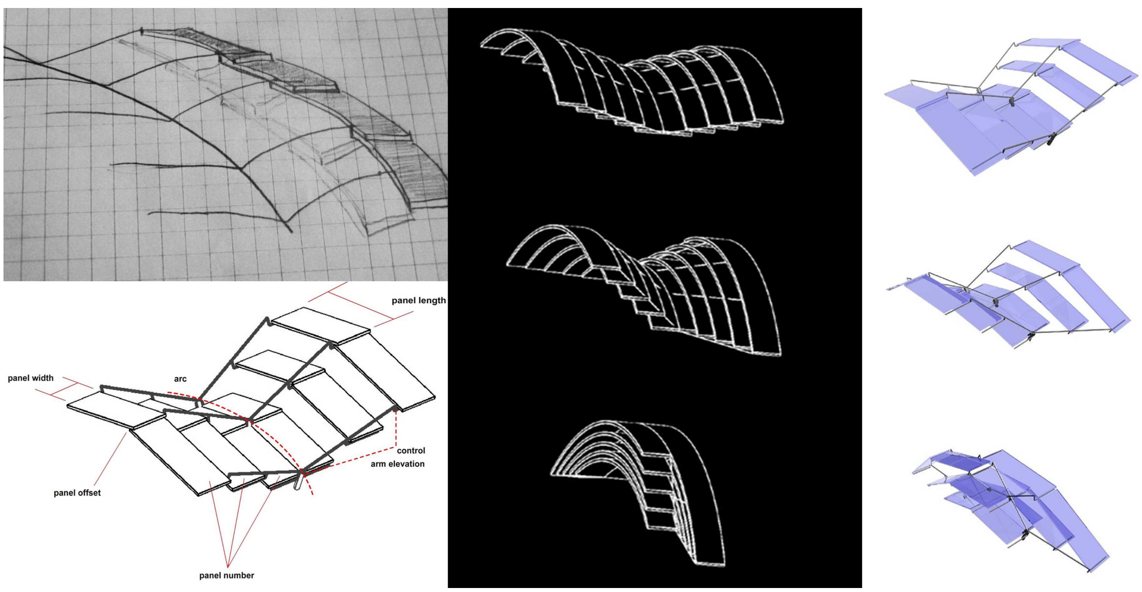

Figure 7.4: ZD in Study 1: Operable Shingled Shading System.

In this shading system, both control arms and shingles are rigid bodies. ZD first abstracted the sequence of shingles into rigid arc panels and anchored the endpoints of controlling arms by *Parameter along the Curve*. For each arc panel, the sliding path is very simple. ZD completed the first stage smoothly. While starting to work on singles, he understood that it was actually a *constraint system*- the sliding and rotation of singles partially determined by the angle and size of their neighbor shingles and partially decided by the overall folding. Because the shingles are rigid, the controlling arms have to rotate slightly to adjust. All of these factors should come from more detailed geometry calculations. ZD was obstructed by the determination of sliding paths of each panels.

7.3.1.3 AK in Study 4: Hexagonal Friction System

AK attended the ACADIA workshop with the assumption that it is possible to apply a hexagonal friction system to a pre-defined BSplineSurface. She used both physical models built with pencils (upper left image in Figure 7.5 and a digital simulation model (upper right image) to demonstrate this possibility. The rigid components in this system are same-length poles which can be neither bended nor overlapped.

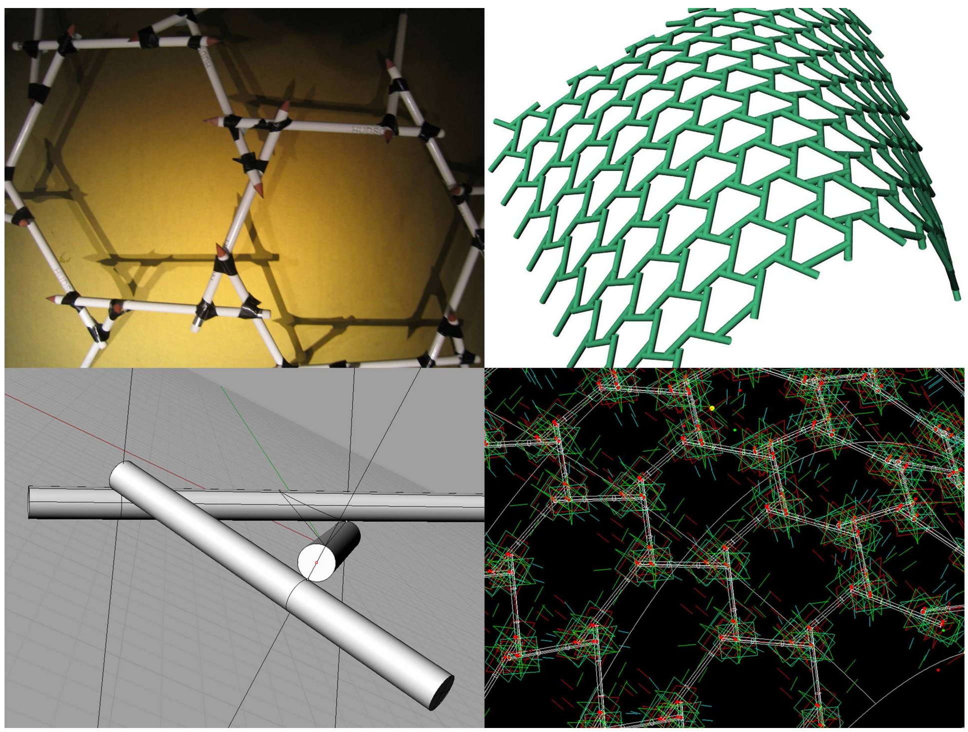

Figure 7.5: AK in Study 4: Hexagonal Friction System.

To avoid collision, AK used the intersected crossing to deal with the joints (bottom left image).

> "I want to allow poles to slide a little bit to fit the curve. I also can adjust the rotation angle between poles. These two options would make a quite flexible structure. In order to do it, I used a set of *Coordinated Systems* to control and

> adjust the poles. But in able to apply on any curved surface, it is very hard for me to define these systems accurately."

Her model could work well when it constructed a single hexagon. In order to link with other hexagons, she had to understand the constraint relations. The main problem in this project is that she started with something too general - trying to apply her idea on a random BSplineSurface. If she had focused on an arc surface as ZD or even a random polygon, she could make use of geometric knowledge and define the rotation and sliding path more directly.

7.3.1.4 Discussion of the Theme

There are two basic features in these three projects: rigid bodies maintain the same shapes and sizes during a transformation, and these bodies rotate or slide along a path or according to rule. So, the central focus of these project is preservation of the rigidity of the components. In order to solve this problem, the designer should first distinguish the rigid bodies from other parts in the models. While the body can be complex, it is typically based on a small collection of simple geometric elements. Except for the position, there must be independent parameters to drive the mechanism. Second, the transformation is constrained by the components. After the first step, the rigid body has one or several geometries as the anchor. Making this anchor transformable is the key. Some properties of this anchor could be variables such as the parameter along the curve, the number of the panels, or the angle of rotation. Being unclear about the anchor transformation is the main reason for AK's failure.

7.3.2 Increment Change

According to the data analysis, this theme concerns that some designers generated elegant whirling or waving effects by accumulating smooth small variations (such as rotation or resizing) along a path or dimension of the model. It was usually achieved by writing a loop function in the script. For each loop, small variation(s) on one or several dimensions were applied. This strategy has been used to simulate whirlwind, construct a wood fence, and build a twisting tower.

7.3.2.1 JH in Study 3: Simulation of Whirlwind

UBC student JH is interested in integrating atmospheric phenomenon into architectural design. Their design studio topic was the responsive facade for UBC Center of Innovation and Technology. He chose to analyze and simulate whirlwinds, and collected some examples such as those depicted in the left photo of Figure 7.6.

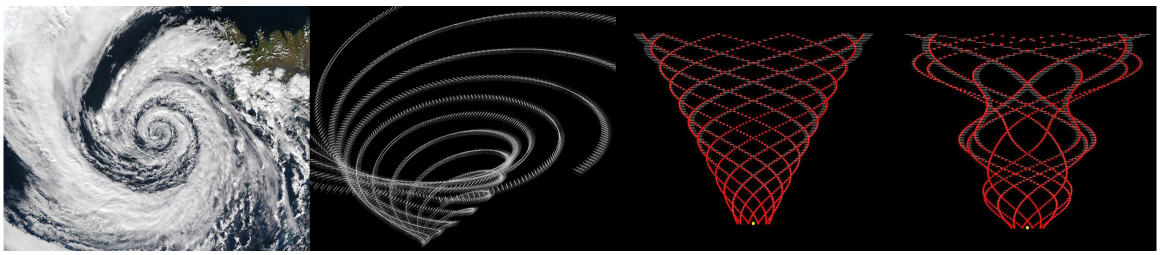

Figure 7.6: JH in Study 3: Simulation of Whirlwind.

JH decided to use a conical spiral to construct his first whirlwind model. Starting from a central base point, he scripted the wind as a rotating sequence of points and used a sequence of line segments to mark these points. For each point, an increment is added to its height, radius, and rotating angle over its previous point. Through a simple algorithm, the whirlwind model was formed quickly (middle images) in GC. To add more variation, JH started to make the increments inconsistent. Instead of setting the increments as global variables, he changed some increments to random values from a range or input values from an external Excel sheet. The whirlwind instantly had many variations. The right image in Figure 7.6 is just one alternative. JH said that he would select some alternatives and use them as patterns in the facade design.

7.3.2.2 LP and EJ in Study 2: Wood Fence Sculpture

The wood fence sculpture in the CDRN fabrication workshop was a collaborative project. A landscape artist designed it in GC and built it up with the help of an architecture educator EJ. Joints of all the components were further designed in CADwork 3D [43]. They used the Hundegger machine to cut timber to get all the components. To make these sculpture pieces easier to assemble, they marked one side of the beams and all the connectors pieces. The final procedure of screwing up all the components and hanging it up for demonstration took them a whole day. Figure 7.7 shows in-progress fabrication photos of the wood fence.

Figure 7.7: LP and EJ in Study 2: Wood Fence Sculpture.

On one hand, the geometry model in this project is very simple - same radius rotation around one axis. There are only increments in two dimensions between the beams: the rotation angle and the distance between the beams. On the other hand, there are some serious fabrication requirements: the size of beam, the shape of joints, and the smallest length that can be cut by the Hundegger machine are fixed. As a result, LP and EJ had to simplify and optimize their GC models again and again to make it fabricatable.

7.3.2.3 DF and PV in Study 1: Twisted Tower Project

DF and PV worked on the twisted tower project in the SmartGeometry workshop collaboratively. PV focused on the plan model and DF was developing the vertical changes. In the tower model, each floor plan rotates and shrinks while it gets higher (in Figure 7.8). The algorithm of this project is actually very similar to the whirlwind project: starting from a central base point, increments in three dimensions of radius, height, and rotation angle. However, because the building floor model is much more complex than a line segment, DF

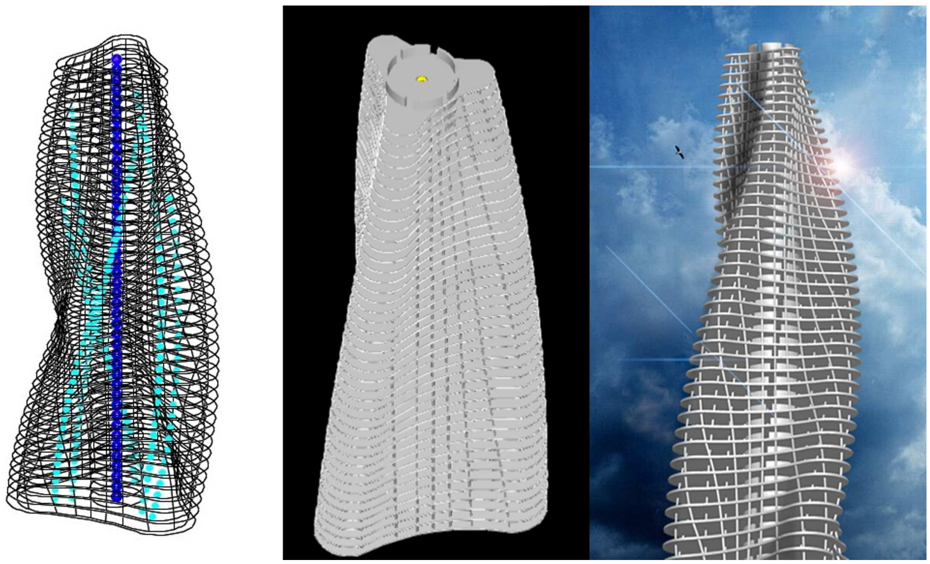

Figure 7.8: DF and PV in Study 1: Twisted Tower Project.

had to package the whole floor as a compiled feature and kept the increments as inputs to reapply on the rotating skeleton.

7.3.2.4 Discussion of the Theme

The prior three projects all chose the same approach of adjusting sub-elements gradually while moving along some dimension(s). Algorithms in whirlwind and tower models linked the values of rotation degree, model size and height together. In the wood fence project, each beam rotates and changes 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 along. The path can either be divided by points at specified T values and then attach the item to each 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 the basic item's parameters with the path's T value or between parameters with the number of steps) to explore design variations.

7.3.3 Responsive Effects

This theme focuses on making an object respond to the proximity of another object. It represents a common design intent in contemporary sustainable architecture structures. The essential idea here is to connect an *interactor* (driving component) to a *result* (outcome component) through a *reference* (other parametrically related components).

7.3.3.1 DH in Study 3: Responsive Matrix

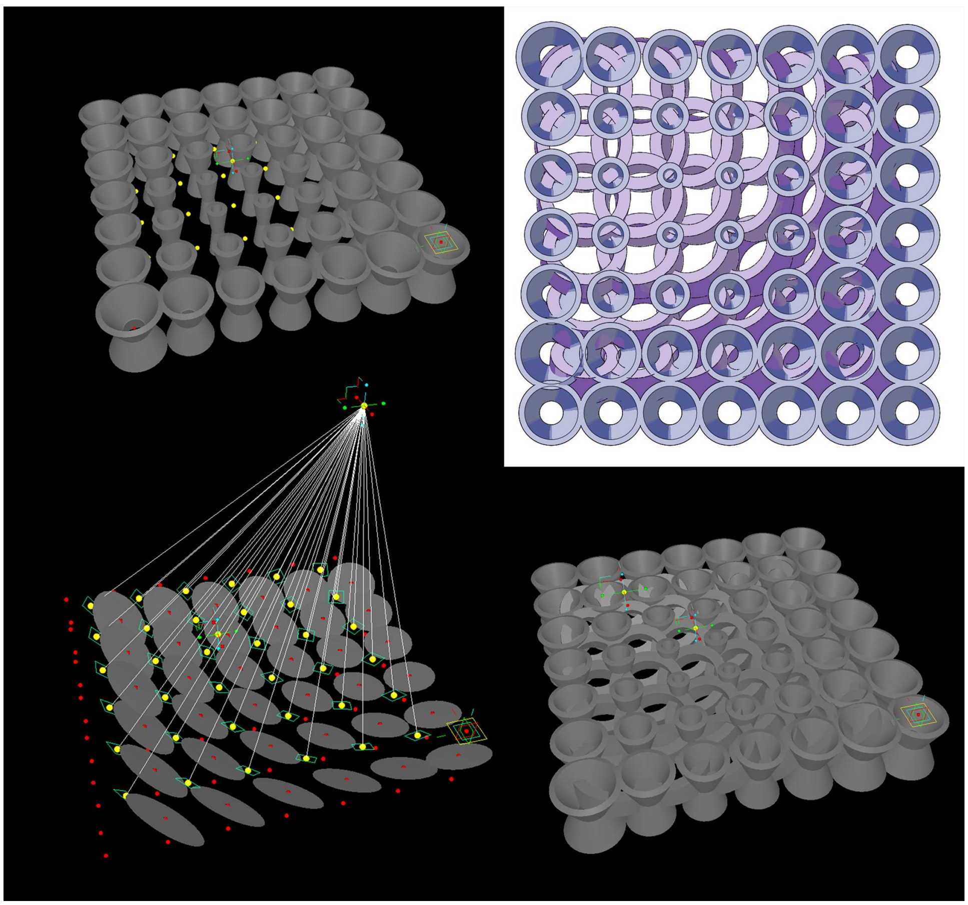

Figure 7.9: DH in Study 3: Responsive Matrix.

To design a responsive facade in the design studio, DH started by creating a responsive matrix as a module for the future facade(in Figure 7.9). In the matrix, there are two sets of objects: trumpet-shape objects and ring-shape objects. The rings fill in the gaps between the trumpets. The two groups are partially overlapped and conceptually independent.

DH used a free 3D point as the reference. According to the movement of this reference, there are two kinds of responsiveness: first, when the reference point moves closer, the size of trumpets will shrink (upper left image); second, the set of rings can rotate smoothly and will always face the reference point (bottom left image). While moving the reference point, the two systems work together and form discrete variations. Some results are open and loose, and some are closed and snapped solidly. For the trumpets, their *interactor* is the distance between the reference point and each trumpet's center point. For the rings, their orientation are controlled by the direction lines linking between the reference point and their corner points. DH told me that this project was inspired by a responsive roof example I showed to the class. In my example, when the reference point moves around, the heights of a matrix set of points increase and decrease according to their distance from the reference. As a result, the roof surface constructed by these points waves with the movement of the reference. DH wanted to explore new possibilities, so he tried to explore the rotation and size variation.

7.3.3.2 FE in Study 1: Multiple Responsive System

FE's responsive system in the 2007 SmartGeometry is conceptually similar to DH' s responsive matrix. The system is also organized by a collection of responsive components and a free reference point. Each component looks very simple - a BSplineSurface with thickness constructed by the function of *fromClosedCurve*(the base is a square-shape closed curve and the upper opening is a circle-like closed curve). However, such a simple structure can be responsive to the moving reference in four ways: the overall height, the opening's orientation, the size of the opening and vertical variation of the opening (in Figure 7.10). The opening's orientation and size of opening adopt the same approach as DH's matrix. Furthermore, FE stretched the shape of opening to be more responsive to the outside reference point. After applying such a component to a matrix of squares, he got a multiple responsive system.

The implementation of this project is an interesting three-step exploration. In the first stage, the opening in the start stage was a circle. FE first set the distance between the reference point and component center as an *interactor*. He linked the height of the component and opening radius to this interactor. But FE was unsatisfied: "with all the openings are all facing up, my model looks less responsive to the moving point". FE began to play

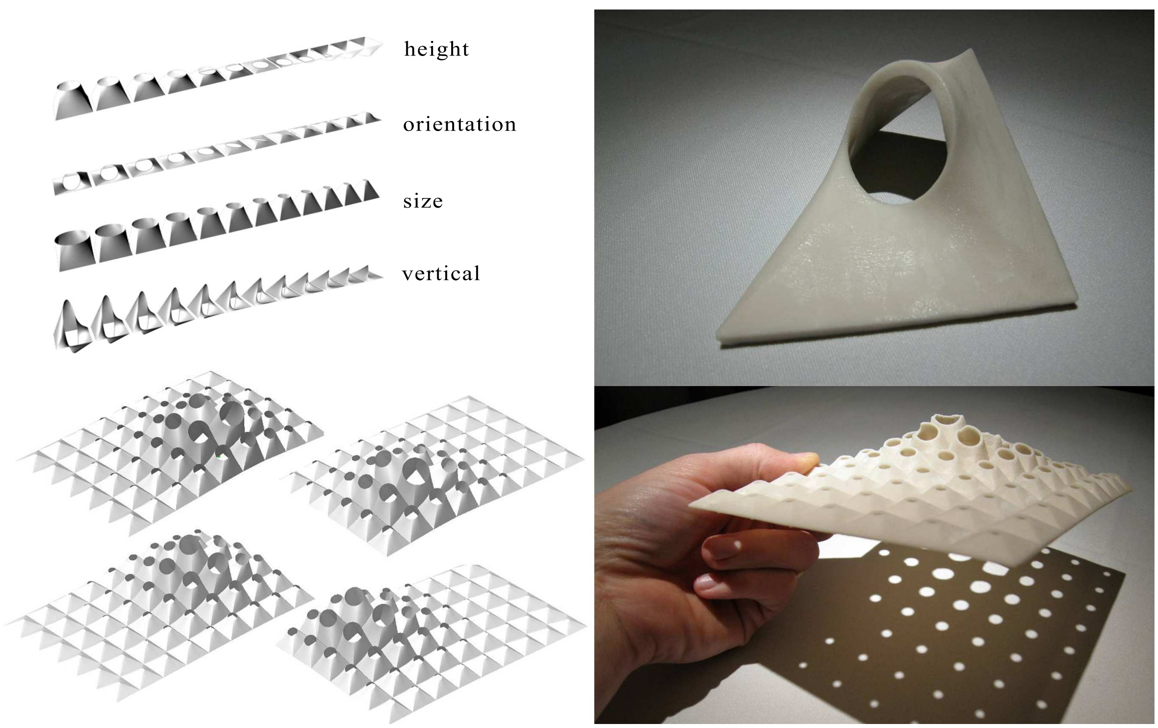

Figure 7.10: FE in Study 1: Multiple Responsive System.

with the opening's orientation. In a new file, He first determined the orientation direction, then assigned a *Coordinated System* whose XY plane always faced the reference point. The closed curve was drawn on the XY plane so that the opening orientation problem was solved. The verticals change on the opening is another step of innovation FE made based on the model at the second stage. Based on the responsive coordinate system, he added another interactor between the height and distance, then drew the closed curve as the opening. In the final 3D printed model, we can see the openings are no longer smooth curves, but in some cases, the shapes have become awkward. Holding the model in his hand, FE was still thinking about an approach to make the simple component more responsive to a moving point outside.

7.3.3.3 MD and SP in Study 1: Smart Roof System

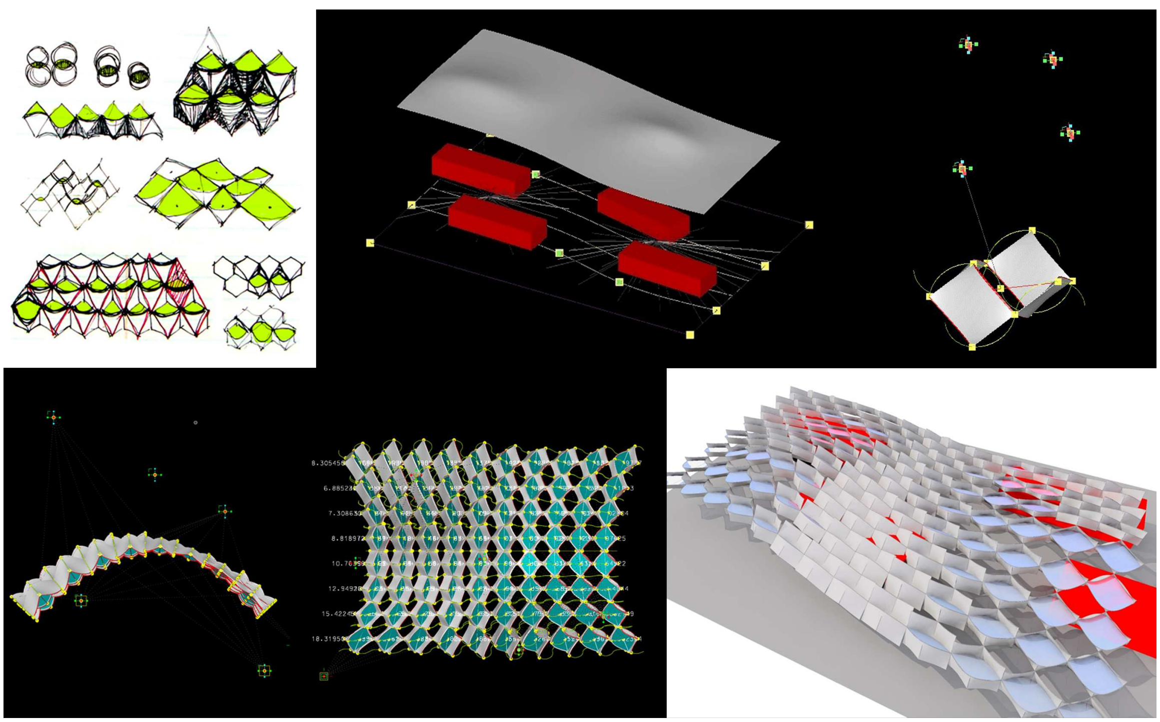

Figure 7.11: MD and SP in Study 1: Smart Roof System.

The responsive roof system MD and SP worked on in the SmartGeometry workshop was complicated. The middle image in Figure 7.11 shows a sketch model of the system: the roof surface is responsive to the four red blocks below. These red blocks represent the hall areas where people gather and communicate. The designers want to design the roof with a kind of responsive structure so that natural sunlight can be sufficient for such gathering areas. They designed the roof by organizing a collection of cells. A roof cell will open up when there is a red block close to it. It will open to the maximum if the red block locates right below it. The overall outcome creates a waving texture on the roof surface. Ideally, when the hall areas move, the roof cells will adjust their openness to meet natural lighting needs.

MD and SP had a detailed design plan for the cells. They first sketched them on paper (upper left image) and then sketched two neighboring pieces in a simple GC model (upper right image). The openable membrane is anchored on two snapped circles. The circles slide between each other to react to the outside distance change. Then the membrane surface bends and opens up. MD and SP first compiled the small structure into a feature and applied on a sequence of circles, and then applied the sequence feature to a surface. The final product (bottom right image) beautifully demonstrated their project design intent.

7.3.3.4 Discussion of the Theme

All of these projects proposed the metaphor of *response* in which one part of a design depends upon the state of another. Reversing the perspective, is as if part of a design becomes a tool for controlling and shaping the other. The essence of parametric modeling is defining object properties in terms of the properties of upstream objects. One particular problem arises - the object definition becomes circular if the relation between the object and its upstream precedents is based on proximity. Because circularity is not allowed in a propagation graph, some designers make an object respond to the proximity of another object. The essential solution of this pattern theme is to connect an interactor to a result through a reference. The trick is to join the interactor and the result with an intermediary and usually fixed object, which we call a reference. The interactor and the reference interact to produce the result.

7.3.4 Branching Variable

The data revealed that another recurring theme entailed trying to keep a flexible number of branches originating from the same central point to explore possible variations.

7.3.4.1 RS in Study 1: Flexible Aviation Museum

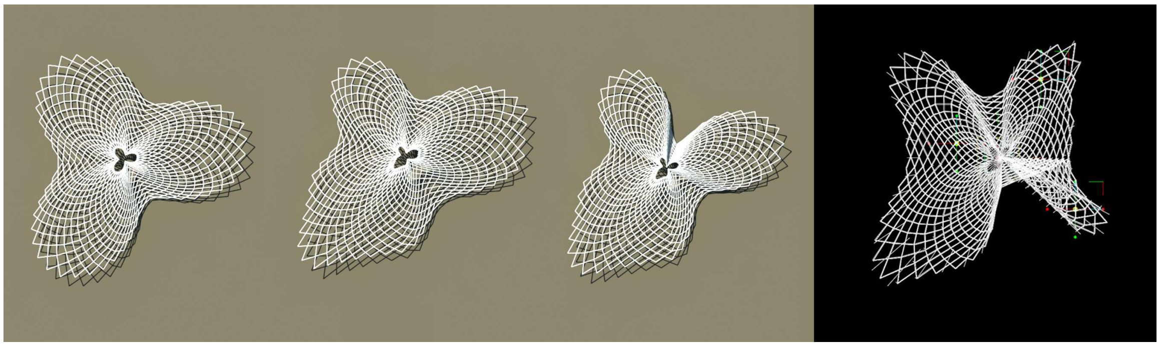

Figure 7.12: RS in Study 1: Flexible Aviation Museum.

In Figure 7.12, RS developed a flexible roof for an aviation museum. The roof structure is built upon three coordinate systems branching from the same central point. Although the systems' X directions target differently, they belong to the same set and are controlled by a variable as the number of branches. During the second and third day of the workshop, RS experienced some scripting difficulties to link up structures between the coordinate systems. He explained that if these three systems are independent, it was very easy to build the weaving structures between them. However, he wanted to control all the coordinated systems as a whole and build a single waving surface to cover all branches. With the help from professional tutors, RS achieved his design goal. While adjusting variables, the three branches can vary their capacity shapes to accommodate different sizes of aircrafts (left three images). Through changing the branching number variable from three to four, the model adds up one new branch (right image).

7.3.4.2 DF and PV in Study 1: Flexible Twisted Tower Plan

In the twisted tower project, PV worked on the floor plan component for DF to apply on the twist skeleton. In DF's tower model, three increments are the radius size, height and rotation angle. For each of the plan, the rotation angle and height can be controlled by the central coordinated system which DF defined as a series. The plan size was provided by PV's model.

Apart from leaving the plan's size flexible to adjust, PV also created a plan that had a

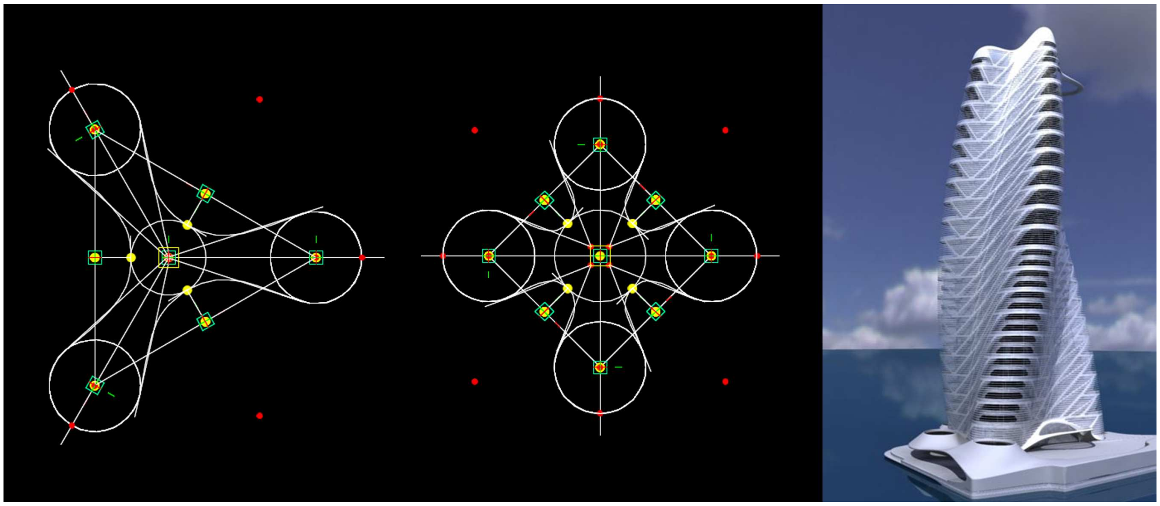

Figure 7.13: DF and PV in Study 1: Flexible Twisted Tower Plan.

flexible number of branches. In the left two figures of Figure 7.13, PV's floor plan ordinates from axis lines starting from the central point. All the plan details such as the locations of end circles and the tangent lines are all built based upon these axis lines. When there are more branches, all the details also amplified correspondingly.

7.3.4.3 AS in Study 3: Limacon Equation-based Geometries

In geometry, a Limacon is a heart-shaped mathematical curve. It is defined as a roulette formed when a circle rolls around the outside of circles of equal radius. The equation in Cartesian coordinates is $(x^2+y^2-ax)^2 = b^2(x^2+y^2)$. Student AS is interested in exploring the variations of Limacon equations and organizing them on a lacework-like structure to be a part of the building facade.

There were two stages in the project: Cartesian coordinates planning and Z direction stacking. In the left images of Figure 7.14, AS drew the plan patterns of the lacework based on a Limacon equation. For all the points, their current Z direction values are zero. Very similar to RS and PV's projects, he also keeps the number of branches as a global variable. The plan can have the different structures when this variable changes. The next stage is to stack the lacework. AS added the Z direction values to the points also according to the Limacon equation. By adjusting the a and b values, the curves in the lacework demonstrate different exquisite forms (middle figures in Figure 7.14). During my data collection, AS still had some scripting problems to connect the beginning and end part of curves, so there was a break at the left corner of the lacework. AS gave the curves a thickness, complied one layer as features, and stacked a few of such layers together. At the end, he constructed an artistic facade.

Figure 7.14: AS in Study 3: Limacon Equation-based Geometries.

7.3.4.4 Discussion of the Theme

Shown by these three examples, there is an idiom of use that all of these projects set a variable as the number of branches and keep the flexibility for future alternatives. This variable can be used to calculate 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 variations for designers to explore.

7.3.5 Variations of Compiled Feature

In the Chapter Six "search for first evidence of pattern-like phenomena", I discussed one idiom of use the function *Compiled Feature* and its background story. It was a function urgently needed by GC users and later provided by the system developers. In the workshop demo created by tutor AK, the designer can use a polygon as a place holder to compile the feature details and transfer all the details to a group of polygons. At the current research stage, I was able to find many instances of this theme: using such a strategy to construct complex structures with hierarchies. In the previous eleven examples I used to describe other pattern themes, there are seven projects using the compiled feature as a strategy:

LM in Study 3: Bending Sheet Metal (Figure 7.3): The feature's inputs are a polygon and an outside controlling point. Output information are the four bendable BSplineSufaces.

ZD in Study 1: Operable Shingled Shading System (Figure 7.4): For the successful first stage, the feature's inputs are an arc and the single width. Outputs are a sequence of shingle panels and controlling arms.

AK in Study 4: Hexagonal Friction System (Figure 7.5): It was an unsuccessful project, AK originally planned to use a hexagon as the input and the outputs would be the snapped pole structure.

DF and PV in Study 1: Twisted Tower Project (Figure 7.8): In this collaborative project, PV compiled his floor plane as a feature and passed it on to DF. The input is the central coordinated system and its outputs are the detailed layout on each floor of the twisted tower.

DH in Study 3: Responsive Matrix (Figure 7.9): Two kinds of features compiled in this matrix: trumpet shape objects and ring shape objects. For each of them, their inputs are the starting coordinate systems and the outside reference point.

FE in Study 4: Multiple Responsive System (Figure 7.10): The responsive cell is the feature output and the inputs are the base polygon and a outside reference point.

MD and SP in Study 1: Smart Roof System (Figure 7.11): There are two stages in the project, MD and SP used the snapped two circles and outside reference points as the inputs of the feature. The feature's outputs are the bendable membranes. In the second stage, a sequence of polygons served as the inputs, and outputs are the sequence of membrane structure.

7.3.5.1 Discussion of the Theme

The overall intent of this strategy is to use proxy objects to organize complex inputs when making collections. In above seven projects, the proxy object can be an arc, a pair of circles, a polygon and a coordinate system. They were used to describe multiple inputs and duplicate the information of a model in an efficient way. Now, it is possible to conclude the situation, reason, and procedure of using such a strategy:

Situation of Use: Copying the model, one copy for each part, and adjusting the inputs to the copies is an effective strategy. Typically a part has multiple inputs—customizing each one is a lot of work. Use this pattern when you are able to describe the multiple inputs to an model through a smaller number (preferably one) of abstract proxy objects.

Reason: If this module requires multiple point-like inputs themselves defined on the target, organizing these inputs is sure to be complex and error prone. If the user can define the inputs to the complex module through a simple construct such as a polygon, it is often much easier to place the module.

Procedures: The mechanism for the strategy has two parts. The first is the proxy object itself: a simple object that carries the inputs for the module. For example, a rectangular module may require four input points, one for each corner. A four-sided polygon can act as a proxy for these points: each of the vertices of the polygon provides one of the points. The proxy simplifies the arguments needed for the module: instead of four points, the user only need one polygon. The second part of the mechanism is to relate the proxy object to the model. For example, a polygon proxy can be placed using a rectangular array of points by relating the polygon's vertices to the points P_{ij}, P_{i+1j}, P_{i+1j+1} and P_{ij+1}. The code required to place a generic object such as a polygon is more simple and reusable than the code for a specific module.

Later on, the detailed discussions of themes were transferred as *Intent*, *Use When*, *Why*, and *How* in the pattern structure in Chapter Eight. Some pattern samples were built upon the simplification of collected projects.

7.4 Summary

This chapter introduces four participant observation studies to search for evidence of pattern-like thinking. Through the data analysis from the multimedia raw data collected from these event, we are able to elicit several pattern themes: rigid transformation, increment change, responsive effects, branching variable, and variations of compiled feature. For each of them, there were real practice examples to demonstrate and present the intent and strategy idea. In the latter part of study, I use them as raw ideas of patterns and start to develop parametric design patterns from these themes and their related practice. In our pattern responsory, patterns from these five themes were named as *Transformer*, *Increment*, *Reactor*, *Brancher* and *Place Holder*.

Chapter 8

Author and Communicate Patterns

In Chapters Six and Seven, we described how we encountered pattern-like phenomena and thinking in users' parametric design work, and that it was possible to elicit such patterns. Here we investigate structures to present these parametric design patterns, along with devices to communicate them. Chapter Eight and Nine are dedicated to discussion of the structure of design patterns and the system design to support pattern authoring and communication. This chapter reviews other pattern structures and nails down one single structure to represent parametric design patterns. For each pattern, an iterative process leads to the generating and polishing of the pattern content. During collaborative pattern authoring, we shared our outcomes with other GC users through a variety of methods.

8.1 Structure Patterns

We want to use a sophisticated and stable structure to construct and represent parametric design patterns. There are many existing pattern languages with established structures. We review and integrat these forms to build up a structure that fits our needs.

8.1.1 How do Others Structure Patterns?

Alexander [10] defines a pattern as a three-part construct: a certain context, a problem and a solution. In his 1977 book, a pattern is represented in the same format: a picture illustrating a typical example, an introductory paragraph setting the context, a title describing the essence of the problem, a problem body, a paragraph describing the solution, and a diagram of the solution. Gamma et al. [83, p.6] used graphical notations to describe design patterns and argue that concrete examples are essential. In the book, the Object-Oriented Programming (OOP) pattern structure includes intent, aliases, motivation, applicability, structure, participants, collaborations, consequences, implementation, sample code, known uses, and related patterns. The structure of Bergin et al.'s pedagogical patterns is a little vague [25]. There is no subtitle to label paragraphs into pattern components. For most patterns, they at least have the three components: problem, context, and solution. Some patterns include examples to support the arguments.

Informal Expression	**Formal Expression**
	A simple *name* by which to remember the pattern.
It states a problem with its context	The *conflict* or *problem* the pattern is aimed at resolving.
It proposes a *solution*	The *strategy* proposed by the pattern, often stated as a simple rule.
The *solution* is spatial and physical	*Argument* that sets out the thinking behind the rule, and discussing nuances of its implementation, often with links and references to other patterns.
	A *diagram* illustrating the rule.

Table 8.1: Comparison of Week's Pattern: Informal and Formal Expressions.

Week [238] introduces two ways to express his workplace design patterns: an informal structure and a formal structure. I compare them in Table 8.1. In his book of "the Culture

Driven Workplace: Using Your Company's Knowledge to Design the Office" [239], both of the two structures were used to present his workplace design patterns.

Tidwell's User Interaction (UI) patterns have a clear and strong structure: name, diagram (usually made by example screenshots), what, use when, why, how, and examples [230]. This structure makes her book and website easy to read and navigate.

8.1.2 Structure of Parametric Design Patterns

Our review shows that patterns can be presented both in a serious structure and as a set of flexible ideas. However, clear structure with high readability can help readers to revisit and use the pattern content easily and efficiently. Building on widely accepted OOP pattern structures [83] and UI patterns [230], we proposes the following structure for parametric modeling patterns:

- **Name** - a noun describing the general function of the pattern briefly and vividly.
- **Diagram** - a graphic representation of the pattern.
- **Intent** - a one-sentence description of the pattern's design goal
- **Use When** - a scenario consisting of a problem and intended context
- **Why** - the reasons to use this pattern.
- **How** - the details of how to adopt the pattern to solve the given problem.
- **Samples** - illustrations of how the pattern can be used in different contexts.
- **Related Patterns** - connections to other patterns.

With such a structure, we can easily communicate or educate people using *Name* as the reference of a specific pattern. *Use When*, *Why*, and *How* tells users under what circumstance they should use this pattern, why they should use such a pattern and how to use it.

8.1.3 Pattern Naming

When a design is completed by a team, pattern names will form a vocabulary they can share. This makes it necessary for pattern names to be both easy to remember and highly descriptive. In Alexander's pattern language, some examples are *Window Place* (helps define where windows should go in a room) and *A Place to Wait* (helps define the characteristics of bus stops and hospital waiting rooms) [13]. The names of Tidwell's UI patterns are also both vivid and descriptive, such as *One-window Drilldown* (linear arrangement for many pages or panels), *Movable Pieces* (self-evident pieces do not need to be laid out precisely) and *Step-by-step Instruction* (helps to organize long or complicated task).

Ideally, parametric design patterns should be named in a similarly descriptive way. However, when we tried to describe complex patterns using short names, we began to realize the difficulty of this objective. We also tried to avoid the direct linkage between a name and a very specific problem context because patterns should be generaliable to be applied in different problems with similar logic. Using Gamma et al.'s software engineering pattern names (such as *Singleton* pattern, *Bridge* pattern, and *Proxy* pattern) [83] as a reference, we decided to also name our patterns in an abstract way to ease communication. Pattern names are usually just one or two terms, and if possible, an explicit and succinct noun ended up with "er" to imply the most extraordinary action in the pattern. Examples of such names are *Controller*, *Goal Seeker*, *Reporter*, and *Transformer*.

8.2 Author Patterns

8.2.1 Collaborative Authoring of Parametric Design Patterns

Pattern-like themes are selected and listed as raw pattern ideas from the previous studies. Our next major task to is to convert such raw pattern ideas into the explicit eight-element structure and coded themes. From the data analysis, some fundamental information has been ready for authoring. For example, in the theme discussion of "variations of compiled feature" in Chapter Seven, the information of *Intent*, *Use When*, *Why*, and *How* has been revealed by analyzing its situation, reason, and procedure. However, problems, solutions, and context in parametric design have many variations. It is difficult for an individual to provide sufficient information for all aspects of a single pattern. In order to maintain the objectivity of writing and provide for inspiration in the design, the authoring process should be collaborative. Built upon intense dialogue over a prolonged period of time, motivated by a desire to transform knowledge, and integrate knowledge from multiple disciplines, collaborations will result in innovation [110].

Several facets of the pattern structure require collaborative effort.

- One part is about the writing in problem, context, and reasons (*use when* and *why*). Coded themes provide the information that one solution serves for one kind of problems. What are the variations of this problem, what is the complete description of its context and what are the main reasons for use such a solution. Although there might be a little related information recorded in the studies, one author's intelligence and such little information are not sufficient for writing these parts.

- The second part requiring collaborative efforts is *how*, which usually organized by one or two paragraphs to explain the details of how to solve the problem explicitly. It is curial to ensure that the solution illustrated is clear, understandable, and applicable to a group of similar problems. This piece of writing should be reviewed and polished by different authors.

- The third part that needs teamwork is the idea contribution of *samples*. Coded models in the studies are definitely good resource for sample creation. However, samples in patterns should be simple and self-expressive examples to demonstrate different variations of the problem. Using a complicated project model as a sample is not a suitable choice. We usually equip a pattern with more than five samples to demonstrate the variety. Brain storming to select best sample ideas is also an important collaborative task.

In many cases, team members share creative ideas with each other during discussions [124]. One pattern may be written by a single author, but sufficient discussions around its content would be useful to enhance the pattern quality, expand coverage of the solution, and ensure rigor.

There were five main members in the pattern authoring team (in Figure 8.1). The supervisor, Dr. Woodbury, directed both the content and system development teams. Three researchers, Roham Sheikhoeslami, Maryam Mokhtarmaleki and I, were responsible for authoring pattern content, and one researcher, Yingjie Chen, focused on programming the repository to support authoring and communication. I was in the content authoring team and also participated in the repository system design and development. In the authoring team, each author focused on developing different pattern ideas. Our weekly meeting is an active show-and-tell process. Apart from components which require definite collaboration (*Use When*, *Why*, *How*, and *Samples*), our discussion sometimes touched upon very specific issues such as the expressiveness of a diagram or the sequences of animations. At times, the discussion would get stuck in investigations of better solutions for a single sample. Pattern authors voluntarily proofread patterns for each other and contributed new sample ideas. While working independently at his/her own desktop, we also used the repository system to leave annotations online. In some urgent cases, we wrote emails to the author mailing list to ask for help or to discuss possible solutions. Such a collaborative authoring process started in May 2007 and continued to the end of 2007.

Figure 8.1: Parametric Design Pattern Authoring Team.
(from left to right: Yingjie Victor Chen, Roham Sheikholeslami, Maryam Mokhtarmaleki, Dr. Robert F. Woodbury, and Zhenyu Cheryl Qian)

8.2.2 XML Template for structured Authoring

During the authoring process, it is critical to preserve and represent the semantic relations among documents from different patterns in a repository. Structured documents associate explicit semantics with content, but authoring rigorously structured documents is a very difficult task [181]. Tim Berners-Lee argues that, "the concept of machine-understandable documents does not imply some magical artificial intelligence which allows machines to comprehend human mumblings. It only indicates a machine's ability to solve a well-defined problem by performing well-defined operations on existing well-defined data. Instead of asking machines to understand people's language, it involves asking people to make the extra effort" [26]. The semantic web offers a collection of models and languages (XML, XML Schema, RDF, OWL, SPARQL etc.) to use "well-defined data" to handle various aspects of document semantics. Each model or language plays a different role, and organized into a semantic web layer [99]. But people have to spend "extra effort" to produce data, makes the semantic web not growing as fast as some would hope. Most document authors will not spend additional time to formally model data thoughts that they have already expressed in natural language or described through drawings and pictures.

Structure provides context for content. We focus on the first levels of the semantic web, namely XML and its schema languages, trying to help authors to safely produce "well-defined data" while they are writing documents. XML's goal is to explicitly represent the logical structure of documents and data. With this structure, programs can securely and unambiguously identify the various parts of a document or a data base, and find hints about the type of information located in each part. Our authoring repository aimed to provide authors with minimal tools making it both easy and efficient to author specific structured data. In a team, pattern authors drafted the XML template's first version. Then this template was edited and improved during the development process. The next chapter will introduce this part in detail.

8.2.3 Current Parametric Design Patterns

At the time of writing this thesis, we have completed the development of thirteen parametric design patterns (in Table 8.2):

For each pattern, we provide five to eight samples with script files and Flash animations online, allowing users to interact with these patterns directly. Pattern authoring is an on-going process. At the time of writing, we are still in the process of completing on the *Recorder*, *Grouper*, *Linear Mapping*, *Random Choice*, *Clear Name*, and *Sampling*. With new pattern ideas coming in from the GC user community, the repository will be enriched step by step.

8.3 Communicate Patterns

The first straightforward method to communicate patterns is to publish them, either in print or online. Alexander et al. [13], Gamma et al. [83], Maiden et al. [134], Week

Pattern Names	**Intent of Patterns**
Controller	Control (a part of) a model through a simple separate model.
Goal Seeker	Change an input until a chosen output meets a threshold.
Increment	Make one or several values of the model grow gradually.
Jig	Build simple abstract frameworks to isolate structure and location from geometric detail.
Mapping	Use a function in a new domain and range.
Organized Collection of Points	Organize collections of point-like objects to locate repeating elements.
Place Holder	Use proxy objects to organize complex inputs when making collections.
Projection	Produce a transformation of an object in another geometric context.
Reactor	Make an object respond to the proximity of another object.
Recursion	Create a pattern by recursively replicating a motif.
Reporter	Re-present (abstract or transform) information from a model.
Selector	Select members of a collection that have specified properties.
Transformer	Change an object's form by translating and/or rotating its rigid components.

Table 8.2: List of Current Parametric Design Patterns.

[239], Graham [89], and Tidwell [230] have published their pattern collections or pattern languages into books. Some on-going projects such as pedagogical patterns [25], cooperative interaction patterns [136], and pattern structure patterns [244] are in print form. Some pattern languages rely on online contribution such as communication revolution patterns [202]. Unavoidably, we see Alexander et al. and Tidwell's work being posted online to invite wider communication as well. Some outcomes of this study have been published in papers [177, 179, 180] and online [248]. Due to versatility and coverage, online publishing opens more perspectives as well challenges, including how to enhance the readability of patterns and engage discussions and contribution. Chen et al. [49] reported the design process and possibilities of the online pattern repository.

Another direct communication channel is delivering pattern-based tutorials in classrooms and workshops. Under this circumstance, patterns suddenly turn into *the form of patterns*. Instead of talking about concepts, theories, or functions, these patterns are teaching strategies. Therefore, introducing or describing individual patterns one by one is not proper. A better approach would be walking through examples that have integrated the adoption of several patterns, and then address the reasons for such adoption. The detailed content of the patterns is left for the practitioners to explore and tryout. In this kind of communication, the problem, context, and solution should be presented and discussed as a whole. Without any part of them, the pattern is incomplete and hard to understand. We tutored two GC workshops (Munich, Germany, March 2008, and Minnesota, USA, October 2008) using patterns and archived successful outcomes. Such pattern-based tutorials and their outcomes also form part of the evaluation of this study. Details are discussed in Chapter Ten.

We tried to encourage discourse during the pattern authoring process, through face-to-face and online means. The repository was designed to serve a large user community, to support discussion and improvement of the pattern project. We posted a link to our pattern repository in the GC user forums and invited voices from different designer groups. We received useful feedback from the forum discussions, and more interestingly, we noticed a large population of "silent pattern readers" when we conducted tutorials in the workshops. After interviewing such readers, we realized that our repository is one of the websites with the richest content about GC, as it provides examples and script files.

8.4 Summary

This chapter introduces one part of the pattern authoring process: determination of pattern structure, the what and why of collaborative authoring, and the design of templates to aid use and communication. To invite discussions around the project, we tried different communication methods to publish and deliver pattern ideas. The authoring process was supported by an online repository system we developed in parallel. The next chapter introduces the repository and its development.

Chapter 9

Support Pattern Authoring and Communication

In this chapter we report on our experience with designing a human-centric repository to support pattern-based authoring and communication. The users of www.designpatterns.ca are designers who want to explore, communicate and share parametric modeling knowledge and strategies. Aiming to support such a community, we provided succinct functions to meet their needs, kept the effort of packaging and publishing to a minimum, and left space for user innovation. We used agile methods to negotiate process and design with users regularly. The chapter records and reviews the process we followed and lessons we learned in the process of repository development.

9.1 Support Pattern Authoring

There were five main members in the pattern authoring team: a research supervisor, three researchers including myself who authored patterns, and one researcher who implemented the repository. The supervisor and I served as intermediaries between the users and developer. The research team met weekly to discuss their work and project issues. The first 15 minutes of every meeting was dedicated to discussing the repository. Users talked about new needs, reported bugs, and suggested better ideas on operation and interface. In short, we developed the system by adding or fixing functions according to users' requests during the entire period of content authoring. We used the agile software development method to develop both pattern content and repository system.

Agile development methods attempt to achieve faster and nimbler software development processes. They were first published by a group of software practitioners and consultants in 2001. The fundamental principles of agile methods include the following [24]:

- Close and regular cooperation between users and developer.
- Users and developers work in the same lab to meet face-to-face as the primary communication.
- All group members are motivated and are trusted.
- Continuous attention to technical excellence and good design.
- Simplicity is always the highest priority.
- As the project proceeds, the users may continuously change their requirements. Any late changes are welcomed.
- The working software with changes is delivered frequently.

There are many kinds of existing agile methods such as extreme programming, Scrum, Crystal family of methodologies, feature driven development and the rational unified process [24]. For us, the critical aspects of agile methods are simplicity and speed, which match our design goal of the new pattern repository. We selected a loose form of Dynamic System Development Method (DSDM). The fundamental idea behind DSDM is that, instead of first fixing the amount of functionality in a product, "it is preferable to fix time and resources and adjust the functionality accordingly [216]". Our limited development resources essentially pegged time at a constant, so we had to reach a compromise between what we wanted and what we could achieve at each stage. In addition to using agile methods we took a deliberate design stance to include no functionality without evident user need. Even such a simple feature as a private area for authors was introduced only when it became evident that we needed it.

9.2 How to Support Different Users in Pattern Communication?

9.2.1 Types of Target Users

In addition to supporting the authoring process, the repository aims to teach and communicate common strategies used in parametric modeling. Target users are undergraduate and graduate students, architects, civil engineers, constructors, industrial designers, or any designer interested in parametric modeling. Traditionally, architecture is taught in studios, with instruction taking place on an individual or small group basis. We chose to work within this tradition. Thus our patterns are not structured as formal course materials - this would guarantee their irrelevance. Rather they are devices to structure dialogue around good experience and provide useful "tricks" experts have and novices need. Also following in the studio tradition, it is very common for new users to post new findings and communicate with each other or experts. This repository needs to make it easy for them to put their work online as new patterns.

On the other hand, we want the repository to be simple to use. Most of our users are expert CAD users. Being simple does not mean that the system has few functions as possible. Rather, it means that tasks are clear and easily accomplished in the work contexts in which they occur. In general, it is impossible to understand the tasks that are actually needed at the outset of system design. This occurs because the act of using a system changes the tasks that the system must support. Therefore, the tasks supported and the resulting system must converge over time if the system is to be successful.

We began with the user roles we developed for the A·VI·RE (A VIsual REte) project [178] and reformulated the use cases for each role in the patterns context. There are four user roles in our system: *curator*, *author*, *critic* and *visitor*. The traditional definition of "curation" is the process of identification and organization of artworks or museum objects in a collection in order to further knowledge. In this repository, the curator discussed the flow of the content, arranged the sequence of patterns and edited the links across different objects. This part of work requires authoring at the system level with more privileges. Some high level or fundamental tasks include designing the structure and form of design patterns (determined at the beginning and during the process of pattern development), server administration, and technical consideration. Most decisions were made by the development team during the weekly meetings. Generally, our development team took the role of curator.

9.2.2 Use cases for Roles

For the other three roles (authors, critics and visitors), we have developed use cases by observing, meeting, and talking to users during the development period. Below I report on three families of use cases, each focused on a particular role.

9.2.2.1 Use case family one: an author

> "I learned the theories of spatial computing in my graduate school. During the past three years, I have used GenerativeComponents frequently in my own design projects. We have our discussion forum and need more participants in the community to share ideas. Through authoring some pattern patterns, I want to show new users the significant potentials of parametric modeling and share my experience with examples. Having a busy life, I can't afford a lot of time on organizing and formatting the outcomes. I want to get concentrated on the text writing and the example creation. If I made any mistake even after publication, I want to be able to fix it quickly and directly." - one tutor in 2007 SmartGeometry Workshop

By observing the authors, I saw three main parts in the pattern authoring process: preparing materials, organizing the content, and publishing the pattern. To prepare for a pattern, the author needs to write introduction text, draw an expressive diagram, build several parametric examples in GC, and capture a sequence of screenshots. After most of these materials are ready, they will start to organize all these materials into the outcome: a document (mostly a webpage or MSWord document) composed by text, diagram, images, links to source files, and sometimes movies or animations, and then publish it. The main intellectual effort of authoring is in preparation, a task mostly beyond automated assistance. We focused on supporting the acts of composing and publishing. Our tools for this included both scripts within GC and website features, for example, a simple "drag + drop" uploading function. A key lesson here is that the system implementation is likely to cross the boundary between online tools and the off-line process of creating patterns. We should carefully choose the boundary between the online and off-line tools by consider ease of use, implementation cost, and system securities.

9.2.2.2 *Use case family two: a critic*

> "I am familiar with the theories and applications of parametric modeling. I am quite interested in the area and have attended the SmartGeometry workshop before. Although I do not have a lot of implementation experience to share, I think my background knowledge can provide some useful advice to those patterns. Firstly, I want to be able to download all the source files of a pattern so that I can have a close look at the script code. But I hate to use right clicks and download items one by one. I also want to ask questions, leave comments or suggest better solutions on the ones I am interested in. In the future when I get enough knowledge about parametric modeling, I wish I can also participate and contribute." - one participant in 2007 SmartGeometry Workshop

Critics are the group of "serious readers". They are usually interested in the mechanism (algorithm and source code) and want to leave comments. To make their job simple, we should allow them to download a pattern as a package. They should be able to jump across different patterns, leave annotations and become an author or even a curator.

9.2.2.3 *Use case family three: a visitor*

> "My architecture classmate told me that this repository was interesting. Before that, I have never heard of adopting parametric modeling in the design projects, but I definitely want to see new and fun stuff. I do not have the time to install a new application, learn its basic functions, and run a file to see how it works. So, if I can understand the thing directly while I am just reading a webpage, it would be brilliant. If it is really as exciting as described, I would go to tell my other friends." - one participant in 2008 SmartGeometry Workshop.

For "visiting" users, we aimed to make the system easy and friendly as an interactive reference book. They should to be able to easily understand each pattern by reading its description, and looking at the 3D model to quickly understand what the pattern does. Then they can look in detail at the source code to understand the 'how' of the pattern if interested. Also, as a computer application, GC is quite large, the installation process is complicated, and the learning process is challenging. The webpage should aim to simulate the outcome taking place in GC so that users do not have to try it by themselves. The interface design should be clean and the text should be very easy to read. The system should also be able to welcome a large increase in the number of users in the future. The size of multimedia documents should be relatively small so that users can download and view them without much delay.

In the repository, each of the four kinds of users has distinct needs. Our solution was to clearly separate roles in the interface design and implement such a design in an effective way.

9.3 Repository Design to Support Authoring and Communication

9.3.1 Start from a FTP based system

Our system originates from a simple FTP based repository. In a SmartGeometry workshop held in February 2006, participants needed a centralized place to share their on-going work and present their outcomes at the end of the workshop. We already had a powerful multimedia repository A·VI·RE ready for use. However, the workshop was so intensive that no time was available for participants to organize their work into well-design PowerPoint files or webpages. Few users were willing to use the standard HTML form's file upload function to upload many files one by one. They were also reluctant to use tools (e.g. WinZip) to compress files (some participants are not familiar with computer tools apart from CAD applications). Their most familiar and preferred way is use "drag + drop" to copy files directly from one window to another. Thus, we decided to create a FTP space in a web server and use server side technology to generate webpages.

In this system, each participant can create his/her own folder, use "drag + drop" to upload all their files from a local folder to the ftp site through their browser (such as Internet Explore 6) directly. Meanwhile, s/he could write a text file in any text editor (such as Notepad or TextPad) and name it as README to describe his/her work. A server page (written in ASP.net) automatically generates a webpage based on the uploaded files. In this webpage, text from the README file is placed on the top, uploaded images are aligned in a sequence and source files are listed as external links at the end. Although the system is extremely simple (the author can not even control the layout of rendered webpage), it turned out to be quite successful, supporting more than 60 participants to upload, share, and present their work in a four day workshop. After the workshop, we improved the README file into structured text. We introduced some Wiki rules to provided limited authoring flexibility. People can use simple html tags (if they know any) and several simple rules to compose the webpage. This simple FTP/Web repository served hundreds of participants in the following GC workshops for a year.

During year 2007, the FTP space became clumsy for the workshop groups because the population of GC users increased rapidly. This created both authoring and retrieval problems. Technically, there were several security issues. It became difficult to manage many FTP username and passwords directly. New versions of web browsers no longer support uploading through "drag + drop". Users have to open an FTP client application to upload files. To save our user's time and enhance system security, we integrated a third party drag and drop Java Applet into the system. Currently this system is still serving the SmartGeometry workshops. Participants use it to collect and communicate their works. This experience encouraged us to build a repository system with better functions based on the same design principles - simple, quick and useful. The A·VI·RE roles, the "drag + drop" functionality were the primary technical features taken from our prior repository projects.

9.3.2 Design Choices for Different Users

9.3.2.1 For authors: an XML template

In order to help authors to correctly produce "well-defined data" while they are writing patterns, the development team suggested using XML as the template for authoring at the beginning of the project. Content authors drafted the XML template's first version. Metadata and tags are defined based on the pattern (design patterns) syntax. The XML template is as follows (at the time of writing):

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="pattern.xsl"?>
<pattern id="">
    <name>Pattern Name Here</name>
    <alter_name>Alternative Name1</alter_name>
    <alter_name>Alternative Name2</alter_name>
    <diagram>
```

```
    <img src="Diagram_PatternName.gif" width="200"
       height="200"/>
</diagram>
<what>
    <paragraph><em>Intent</em> states a one-sentence
       description of goals behind pattern <patternName>
       Reactor</patternName> and its relation to pattern
           <patternLink>Controller</patternLink>
    </paragraph>
</what>
<when>
    <paragraph><em>Use When</em> describes a scenario
       consisting of a problem and a context.
    </paragraph>
</when>
<why>
    <paragraph><em>Why</em> states the reasons to use this
        pattern.
    </paragraph>
</why>
<how>
    <paragraph><em>How</em> explains the details of how to
        adopt the pattern to solve the given problem.
    </paragraph>
    <paragraph>Second paragraph of <em>How</em>.
    </paragraph>
    <pre>pre-formatted text (Code) here</pre>
</how>
<samples>
    <sample id="">
        <name>PatternName_SampleName</name>
        <when>
            <paragraph>Context of the sample
            </paragraph>
        </when>
        <how>
            <paragraph>Details of the sample
```

```
                    </paragraph>
                </how>
                    <a href="Samples/SampleName/Sample.gct"> Click
                        to see SampleName.gct
                    </a>
                    <animation src="Samples/ SampleName/
                       Animation_SampleName.swf"/>
            </sample>
        </samples>
        <relatedPatterns>
            <relatedPattern ref="">
                <paragraph><patternName><patternLink>   Controller
                   </patternLink></patternName> is related to this
                    pattern
                </paragraph>
            </relatedPattern>
            <relatedPattern ref=""/>
            </relatedPattern>
        </relatedPatterns>
    </pattern>
```

XSLT is used to display XML structured to this template into readable WebPages. Currently we provide one standard XSLT file for all patterns. In the future, it is possible to customize the XSLT for certain patterns to display them in different forms of presentation.

There is not much syntax for writing a pattern, but we found that our authors still did not want to remember grammar, syntax, or hierarchy of file organization. While authoring a new pattern, they never started from scratch. They usually duplicated an existing pattern into a new folder to use it as a template. However, the files and text in the old pattern may not have everything ready for the new pattern (e.g. some tags were deleted because they were not used in that pattern). We also continuously developed new syntax based on authors' requests. Remembering all the syntax and rules, or retrieving new or missing tags, is still extra effort.

Therefore, we created a template pattern containing the most recent package definition. The package contains the main control XML files, file hierarchy of images, animations and source files. In the template XML file, every possible tag (with sample text), rules and external file links are included. The author does not need to remember syntax. S/he just has to download the package, delete unnecessary parts and replace files with their real content. To author a new pattern, s/he always can use the most updated package. Considering that XML may still be too technical for some GC users, we kept some features from the FTP based predecessor repository. The user can create a folder in the system, which is the same as making a new pattern, and "drag + drop" files into the folder. The system will pick these files up and display them as much as possible. Thus we made this repository more generic.

9.3.2.2 For authors: manual coding instead of WYSIWYG GUI

The core of the pattern is the text file containing explanatory and external file links. It must be displayed as a webpage. To create such a document, a GUI interface with WYSIWYG would be welcomed by many novice users. But for experienced users, command line interaction, desktop text editors or scripts enable them to work in a more flexible and efficient way. Most users have their own favorite text editor, so we do not need to provide a web based GUI interface or editor. After negotiating with content authors, we decided to use a simple GUI to create, upload and delete patterns in the system. The pattern itself is organized and written in a user's own text editor. Files, images, and text are created and packaged locally. Authors can upload and publish the content as a whole or in parts to the server through a drag and drop web interface. Minor changes can be made in the simple online editing window. While it is possible to edit directly online, our authors far preferred the faster offline process.

9.3.2.3 For authors: naming convention and file hierarchy

At the beginning of the project, we set simple rules for the syntax of the XML file and folder hierarchy. After a few weeks, the content authoring team found it essential to make the file organization explicit and meaningful. Then we formally wrote down naming conventions and posted them in the public space. These rules were also improved and edited during the following weeks.

To collect feedback and comments from external users, the repository was open to the public from the outset. With more and more colleagues and friends discovering the site, the project team realized that they need a private place for themselves to store temporary files and internal documents. We thus separated the system into one part that is open to the public and another that is password protected. The naming convention document, the

XML template, and other development information were moved into the protected space. Only registered content authors are able to view them.

9.3.2.4 For Visitors: Animations of 3D Models Online

The need to present 3D models more vividly online arose only after authors had created several patterns. The models being conveyed are parametric - their 3D form and appearance changes when their parameters are adjusted. To help people understand the problem in a 2D screen, the model should at least be shown from different perspectives.

An "ideal" solution allows users to manipulate the model directly in a 3D environment online. However it is hard to achieve in an inexpensive way with current technologies and resources. Although learners can download source files and run them on their local machines, it still interrupts the flow of reading. Thus we decided to use animation to demonstrate the dynamic aspects of the 3D models. The animation is implemented as a Marcromedia Flash .swf file, because Flash is provided as a standard plug-in for nearly all web browsers. As people may not familiar with the creation of Flash animations or Actionscript, we devised a solution to have the Flash file generate an animation by automatically picking up a series of images from a folder (Figure 9.1).

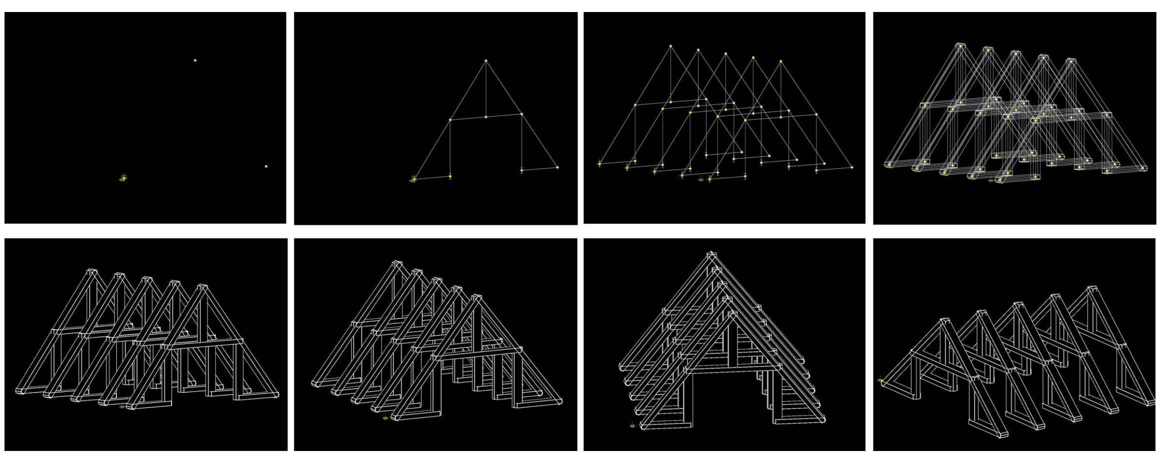

Figure 9.1: Several Frames of Images to Compose one Animation in a Pattern Sample.

Creating an animation now becomes easy. Authors first capture the screenshots as frames of the animation. This may be a manual process or may be scripted inside the parametric modeling system. A team member developed an auto-screen-capture module. Users simply insert the module into their source file. After initiating the module, updates of the screen caused by the parametric model are captured and saved into a sequence of images. To provide more control of the animation, we designed a simple XML file (also can be modified in the template) to define the appearance of the animation (display sequence and time interval). Flash automatically reads the XML file and images and displays them in the webpage as an animation.

A sample animation control XML file follows:

```
<gallery timer="1.5" order="sequential" fadetime="0.5"
   looping="yes" xpos="0" ypos="0" scale="0.5">
    <img src="PlHo_Hedgehog_00000.jpg"/>
    <img src="PlHo_Hedgehog_00001.jpg"/>
   ...
    <img src="PlHo_Hedgehog_00020.jpg"/>
</gallery>
```

Later, authors reported that they wanted multiple animations for one sample in a pattern. Therefore, in the naming convention part, we created a new rule for naming the Flash and XML file - the Flash file would only read the XML file with exactly the same name and pickup images listed in that XML file. Thus, the authors had more freedom to create animations efficiently and directly.

9.3.2.5 For critics and other users

At the beginning, we focused on the needs of authors and visitors. With more colleagues visiting the system, we found that we needed to facilitate critiques. The first critiques came internally from the project team. Content authors read others' work and left suggestions. Although we had face-to-face meetings, we still needed to write done suggestions as they occurred to us. Some external visitors also wanted to contribute new ideas or leave comments on patterns. We created a registration module and a discussion board. External users can register and discuss patterns of interest, or any part of the repository. In each of the patterns, it is also possible to leave annotations directly.

9.3.2.6 Remove and simplify functions

Having more functions does not necessarily make a system work better. Inherited from the FTP based repository, some features of this system were not welcomed by users. For example, in the FTP repository, files in the server are organized in the folder tree structure, and the navigation interface reflects the tree structure. Users are able to "go back to parent folder" or "back to root". The development team kept the folder navigation tree in the interface at the beginning. But in such a pattern repository, visitors do not need to know how files are organized from the server side and this actually confuses users. The development team received complaints several times in group meetings. So we removed this folder navigation interface from this system. Although one developer still thinks it was a very useful feature, no end user felt inconvenienced by its removal. To reduce the users' cognitive load and confusion, the system should provide the users exactly and only what they need.

9.3.3 Repository Design Potential Extensions

The next step we want our system to achieve is allowing GC developers are implementing key features of designpatterns.ca directly inside the modeler. The *auto-documenter* will remove the need both to author and upload objects. The essential idea is to provide pattern authoring tools as configurable modeling elements. By changing the preference settings on such tools, modelers will be able to directly create patterns and have them uploaded to a chosen site on demand. The repository will become merely an extension of work being done. By changing the XSLT transformation on the stream of such work arriving at the repository, we can repurpose content in a variety of formats.

Although the current system was built to host one certain type of parametric design pattern, we want to extend this project to handle more in the future. Currently we use a simple XML metadata to organize our patterns. We can extend our XML definition to handle a wider range of specifications, standards, and developments. The next research group goal is to fit our current XML metadata into IEEE Learning Objects Metadata (LOM). Due to the flexibility and extensibility of XML standard, this migration should be fairly easy with minimum impact to this research group and pattern authoring.

9.4 Summary

The design pattern repository system is aimed at users who are busy with their own professional work. Working and sharing their own work with the system is only a tiny part of their professional life. They do not have time to explore and learn features of the system.

If a contributor meets any problem during his or her first visit, he or she is unlikely to take a second try. Simplicity and directness is a key part of encouraging contributions from a wide audience.

Currently, the designpattern.ca repository is neither a sophisticated multimedia repository nor a complete content authoring tool. It offers basic functionality and aims to provide simple and sufficient support to its users/authors. Because of the timeline of the project, we did not have enough time to build the system before the content development stared. We chose the DSDM agile method and communicated with our users regularly and frequently during the entire process. The chapter summarizes our experience of designing this pattern repository and authoring platform. Trying to create a simple and elegant system for all the users, we observed users, analyzed use cases and outlined essential needs of target users.

During development, we continued to receive feedback and advice from the GC community. In the following evaluation study, this repository served as the key platform to teach and communicate parametric design patterns with our participants.

Chapter 10

Evaluation of parametric design patterns in Use

To answer the third question underlying the research process "*how well are we doing?*" [135], this PhD study demands a realistic and researched evaluation of how our patterns are working in relation to our stated purpose and goals. The literature review in Chapter Two constructs a cognitive framework for examining and understanding whether patterns meet design process requirements and whether they support design activity.

In this chapter, I introduce two studies, conducted at the 2008 SmartGeometry and 2008 ACADIA software workshops, that evaluated parametric design patterns through the proposed framework. Chapter Three adopts McLuhan's tetrad of media effects [143] to evaluate parametric systems in physically based modeling and spreadsheet modeling. In the later part, I discuss pattern based parametric CAD systems through the lens of media effects.

The evaluation studies were all based on current thirteen completed parametric design patterns. Here, I believe that lessons I learned from these studies will be useful to guide future pattern development and existing pattern revision.

10.1 How do Others Evaluate Patterns?

It is comparatively difficult to find evaluative studies or summary discussions of patterns' qualities in established pattern languages. Typically, authors focus their effort on describing patterns and pattern languages, compiling descriptive content into 'pattern dictionaries' without providing evaluation. In more recent conference literature, a few authors have started to discuss the process of evaluating patterns.

In his work on the communication revolution pattern language [202], Schuler provides a long discussion of openness, fairness, and effectiveness as inter-related aspects of processing and producing patterns [203]. Connelly et al. [56] describe the extension and evaluation of prior work on a pattern language for safety in user interface development. They performed an evaluation of their pattern language using a safety-critical user interface for a hypothetical radiation therapy machine as a design case study [56]. Although the authors characterize the research as a qualitative study (p. 47), what they describe was instead an experiment run through the process of designing the interface for the therapy machine using patterns. The study did not involve a subject and no data was collected during the process. While the study may provide further understanding of the usage of patterns, it does not suffice as a trustable evaluative study. Ronteltap et al. [191] describe their process of designing a pattern language to aid in designing productive learning conversations. They ran two sets of experiments to gather feedback during the development process. Details of these experiments were not provided in the paper.

Two studies encountered aimed at evaluating the effectiveness of pattern-based projects.

A quantitative pattern evaluation study: Deng et al. [71] adopt a comprehensive evaluation framework named DECIDE to conduct a formative evaluation of their prototype MUIP (Management of User Interface Patterns). DECIDE focuses on determining the specific goals of a particular evaluation and then choosing an appropriate evaluation technique [208] Using the DECIDE framework, Deng et al. translate the main goals into a set of questions. Three user-based techniques were selected for the evaluation: observation, questionnaire, and interaction logging. They invited five computer science staff members who were familiar with MUIP to perform five scenarios. Each participant explored the system for ten minutes, then an observer recorded data from scenarios using a template. The observer also took notes about requests for help and any other comments. Once the participants completed all the tasks, they answered a questionnaire that included both closed and open- ended questions. Most closed questions in the questionnaire adopted the Likert Scale for responses. The final result were not so positive for the interface design, but confirmed that MUIP was helpful for managing and manipulating UI literature.

A qualitative pattern evaluation study: A CHI short paper [68] presents an evaluation of pattern languages as tools for participatory design, based on three criteria derived from the work of Christopher Alexander [13, 10, 11]: empowering users, generative design, and life-enhancing outcomes. Design exercises were conducted with six participants, ranging in experience from a retired non-web-user to a trainee web designer. Two participants were given the pattern language to read for one hour in advance of their design session. Two were introduced to the language at the beginning of the session. Two were introduced to individual patterns or groups of patterns as the design session progressed. In all cases, the participants were asked to develop a paper prototype of a travel website using the pattern language. All the participants were told that following the patterns was not compulsory, and that the illustrations shown were examples only and not definitive 'best practice'. The sessions lasted between one and two hours, and were videotaped. All participants were interviewed following the task. Dearden et al. [68] outlined their observations in relation to their three criteria and discussed insufficiency of data and limited evidence of the pattern language's success.

Both of these studies adopted evaluation criteria from external literatures and designed tasks for the participants to achieve. At the end, they were able to evaluate observations and outcomes based on methodological criteria and derive limited conclusions. These examples demonstrate that it is possible to design and implement different kinds of studies that evaluate pattern outcomes. Although not wholly satisfactory, the structure and methods employed provide some inspiration and a point of reference for this study.

Why are there so few studies that evaluate patterns? Why do most studies give poor results? We argue that we should not expect short term results from patterns. A pattern, as a particular mental construct, does not show explicit structure in human practice. That is why we need to search for pattern-like phenomena to elicit pattern themes. In addition, after being delivered to designers, pattern ideas are only gradually embedded in human reflection. Thus, the benefits of patterns would only be observable in multiple work sessions or after days, weeks, or even months. As such, a long time frame is required to see pattern effects. However, we are provided only with short time frame events to conduct our evaluation studies. Instead of trying to discern stabilities of patterns, we should be able to detect early signals of pattern use. The principle underlining it is human's reflection process. Short time frame studies do not tend to show something significant. But from early signals, we expect to see the integration process of pattern ideas and designer's existing design skills.

It is very hard to judge whether or not a group of patterns are 'successful' even we are able to collect the data of users' long term reflective thinking upon patterns. There are too many aspects to consider the 'success', such as well written, well represented, well communicated, or widely used. For this research, I delimit that being *useful* is the key aspect for the "success" of parametric design patterns. As I mentioned in the second research question, patterns should be useful while serving as learning materials, scaling up problems, and being communicated among users. In the stage of evaluation study, I expect to see firstly patterns can be practically used in real design projects, being cognitively easy to interpret and functionally supportive to solve design problems.

In my evaluation studies, I have also adopted evaluative criteria from external literatures. Considering the variety of design tasks, I want to investigate what exactly takes place in real design practice. Instead of assigning fixed tasks to participants, I choose to observe their activities when they are working on their own projects.

10.2 Study to Evaluate Parametric Design Patterns

10.2.1 Research Methods

Similar to the previous studies aimed at gathering evidence of pattern-like thinking, this evaluation also employs participant observation method and contextual inquiry techniques to elicit feedback from participants when they are working on their projects. These two methods can provide rich data from participants without seriously interrupting their work flows.

Suggested by Dr. Lyn Bartram, I brought the method of *journaling study* into the second study to enrich our understanding of the participants' design context. Journaling is a highly personal form of written self-expression, and journaling study was useful to investigate phenomena that were hard to observe and annotate explicitly [64, 101]. Writing in a journal enables people to learn more about themselves and better express themselves creatively without having to face the comments and criticism of others. There are different forms of journaling. Schiano et al.'s study [199] took blogging, a popular online communication activity, as a medium for personal journaling. In the context of architecture, many architects have the habit of carrying a sketch book around to record their creative ideas.

These sketches also act as a medium of self-expression and communication. Collecting and analyzing such 'journals' provides contextual information about their design practice. Although it is impossible for me to conduct prolonged observation studies, the sketches can inform me facts about how patterns are involved in designers' practice in a longer term.

10.2.2 Role of the Researcher

We employed *active participation* from Spradley's participant observation continuum [214]. We, as researchers, are engaged in most of the same activities as our subjects as a means to learn the rules of their behavior.

During the beginning of workshops, Dr. Woodbury and I gave a series of tutorials with examples composed of some pattern concepts and samples to participants. Also the pattern repository was introduced to them. Participants are free to reference and use strategies/-solutions in the repository, and they are also free to copy, paste and modify the source code from any of the samples. Apart from providing them with the script files we were demonstrating on the central projector, we went around the room to answer questions as they arose. During the formal workshop, we acted as tutors for designers (workshop participants), simultaneously observing and discussing how they were working. We posed little overhead on any particular designer as the discussion and observation is essentially what already happens in a tutoring session.

10.2.3 Settings of Two Evaluation Studies

10.2.3.1 Study A: SmartGeometry 2008 Workshop

The first evaluation study was conducted at the Arabella Sheraton Grand Hotel in Munich Germany from February 29th to March 3rd 2008. During the pre-training workshop, Dr. Woodbury and I delivered tutorials to more than seventy participants in a standard classroom setting (Figure 10.1). The participants (more than 150 and mostly architects and civil engineers) were selected into six groups (structure, environment, fabrication, form, computation, and architecture). Every group occupied a work room and had three professional tutors (Figure 10.2). 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 pre-workshop that introduced the basic functions of GC. Participants used their own laptops during the intensive four-day session.

Figure 10.1: Settings of the SmartGeometry 2008 Pre-training Workshop.

Six subjects from the pre-training workshop who had prior experience with GC volunteered for the study. I tried to keep the sample distribution similar to the whole population of the workshop. In term of gender, the subjects includes one female and five males, between the ages of 25 and 45. The female-male ratio in the event was approximately 18% – 28 out of 154. In term of occupation, three of the subjects were graduate students, two were industrial professionals and one was a design firm director. In term of profession, five were architects and one was a civil engineer. In term of their goals for this workshop, four planned to use the workshop to solve design problems in current commercial projects. Two focused on academic studio designs. All of these subjects had learned GC for at least 3 months and it was the first parametric modeling application they had ever used.

Figure 10.2: Settings of the SmartGeometry 2008 Individual Workshop.

In advance of the workshop, each participant had submitted a project proposal centered on one or more design problems in a current architectural project. The proposed projects necessitated parametric modeling, such as solar analysis and panel arrangement on a free-form structure. While all of the architecture projects come from teams, only one representative of the project was in the workshop. In one case, two members from the same work team participated (GF and HC) – they worked on separate parts of the design project and had to integrate them at the end.

Apart from meals and presentation sessions after dinner, the subjects spent most of their time on the design projects, aiming to complete them by the end of the workshop. They sometimes had short chats with their teammates and neighbors or negotiated solutions with their tutors (Figure 10.2). In order not to interrupt their work, we interviewed the participants by request, when they were taking a break or during moments of relative calm. We engaged them in short conversations (3-10 minutes) two or three times a day. We chatted about their progress, problems they encountered, and potential solutions toward those problems. The conversation length depended on the topics emerging during the process. Audio from these conversations was recorded digitally. We also collected other data such as photos of physical models, digital sketches and hand sketches, screenshots of existing problems, script segments and successful models, and all GC feature and transaction files at each stage.

Data was not only collected from these recruited subjects. Since many participants were already get familiar with us from the pre-training workshop, some (including tutors) approached us for possible solutions to their problems, to share their progress, or chat about their work. With their permission we recorded these conversations. By the end of the workshop, we also had the necessary data from four other projects. In total, we collected data from ten subjects in all.

10.2.3.2 Study B: ACADIA 2008 Software Workshop

The ACADIA 2008 conference seeks to identify and examine current trends in digital design technologies. The three-day GC software workshop was held in a meeting room in Walter Library at the University of Minnesota from October 13rd to 15th.

In order to register the workshop, every participant was required to submit an statement of interest centered on design problems in a current architectural project. Workshop organizers selected twenty participants after a review of submitted statements. Similar to the ACADIA 2007 workshop, most selected participants were architectural educators or professional architects. Because of the conference theme, many participants had express an interest in exploring the modeling or imitation of biological structures using parametric CAD systems.

In this event, we tried to integrate the method of journaling study of sketches. We first asked the conference organizers for permission to read the statements of interest participants submitted, to understand their design intent. Among those statements, I selected six participants as potential subjects. In the invitation emails, I asked them to bring their sketchbook to the workshop. Luckily, all of the candidates agreed to participate in the study. The subjects were four architecture educators and two professional designers. However, all of them are males because of the significant imbalance of gender in this workshop – only one female graduate student among all the twenty participants. They had all heard of or tried GC and other parametric software in advanced of attending the workshop, but never took any serious training in parametric modeling.

Figure 10.3: Settings of the ACADIA 2008 Software Workshop.

During the three-day workshop, four tutors worked with the group (Figure 10.3): Dr. Woodbury and I from Simon Fraser University, Kaustuv DeBiswas and Makai Smith from Bentley. On the first day we delivered GC tutorials integrated with our pattern examples. Then we encouraged participants to go ahead to work on their proposed projects over the following two days. They are free to use any resources from our pattern repository. With enough tutors around, participants can get help directly whenever they met some problems.

The outcomes of workshop were organized and presented by Dr. Woodbury in the later conference.

10.2.4 Data Collection

The data collection process is very similar to the participant observation studies that gathered pattern-like thinking. I amended my ethics form twice and obtained the approvals on January 31 and September 23, 2008. After my subjects signed their individual consent forms, I engaged them in short conversations two or three times a day. During the conversations, we chatted about the progress on their projects, problems they have met and potential solutions towards those problems. Sometimes subjects used sketches to explain detailed problems to me. These short conversations were recorded digitally. Information such as photos of hand sketches, physical models, screenshots of existing problems, script segments and successful models, and all feature and transaction files were collected at each stage.

Two weeks after the ACADIA workshop, I contacted these subjects and presented them with a short email questionnaire to check on their subsequent progress with GC. I asked if they were able to provide new hand sketches and model scripts for me to analyze post-event affects. The sample email was prepared in Appendix A.1.

Having spent nearly one week at the ACADIA conference, most participants returned to their already busy lives. Also, without support from the workshop, some participants did not continue their work in GC. I was able to elicit replies from four subjects and had prolonged communication with two of them.

In these two workshop studies, we ensured that no personal information was collected. Subjects were interviewed separately and no information from the interviews was divulged to other subjects. Information was collected and saved in a secure storage and subjects can fully access to their own information. In the publications and reports, I only use materials they agreed to share.

10.2.5 Data Analysis and Evaluation Criteria

I used ATLAS.ti 6.0 organize and analyze data in this study. Data included text (GC feature and transaction files), images (photos of sketches and physical models, screenshots or photos captured during the process) and audio data (recordings of short conversations). All data belonging to a single subject's activities or one design project was packaged together into a single unit. In such a unit, there are usually more than twenty documents categorized by name, media type, author, creation time, modified time, and origin path in the hard disk drive. From the two workshop events, I collected and built sixteen units.

Boyatzis introduces three approaches to developing themes systematically: theory driven, prior data or prior research driven, and inductive [33, p.29]. My pilot study to search for phenomena and participant observation studies to gather pattern theme all adopted the inductive (data-driven) approach because of the lack of coding theory. In term of these two evaluative studies, the data analysis goal is to search for early signals of how patterns cognitively exist in designers' work, communication, and practice. Facing nearly two hundred files, I chose to use a theory driven approach (top-down approach) to analyze data. The cognitive design frameworks I constructed in Chapter Two is the analysis theory I use to analyze and evaluate parametric design patterns. I investigated data in two cognitive directions:

- Do patterns meet the cognitive requirements of the design process?
  - Provide contextual information in the structure
  - The quality of language-based and graphic-based representations
  - Available to conduct the cognitive work analysis
- Do patterns cognitively support the design practice?
  - Support parametric CAD tool learning
  - Support building up the expertise of scaling up
  - Support collaborative design practice

According to this framework, I firstly defined fifteen basic codes: *context-struct*, *context-system*, *represent-text*, *represent-graph*, *cwa-goal*, *cwa-around*, *learning-reflect*, *learning-scaffold*, *scale-direct*, *scale-unaware*, *scale-other*, *collab-struct*, *collab-system*, *collab-internal*, and *collab-external*. For example, in a conversation recording, a subject's statement related to the readability and connection of the pattern's structure will be coded as 'context-struct'. If this information is specifically related to the *Name* part in the pattern structure, the higher level 'context-struct-name' code would be assigned. An image to demonstrate a Flash animation's effects would be coded as 'represent-graph-animation'. We also use the thirteen pattern names as basic codes, such as *pt-cont* (*Controller*), *pt-plho* (*Place Holder*), and *pt-repo* (*Reporter*). A phenomenon in a script file to use scripts of sample Arms in pattern *Transformer* can be coded as *context-struct-sample*, *represent-text-script*, *learning-reflect*, and *pt-tran-arms*. All files were coded at least two rounds to make sure we interpreted and coded the data sufficiently. During the coding process, we realized that there were instances that participants suggested new pattern ideas. A basic code of *pt-new* was added to the list to anchor such conditions. In total, there are twenty-nine basic code. Any other codes all have to add extensions upon these basic ones.

Two coders (Yingjie Chen and I) were employed in the coding process. Naturally, different coders see different things in the data. When differences arose, we discussed the data until some agreement was reached. In some cases, this was resolved with a new code. In other cases the data was coded into more than one category. Since the basic codes have already been linked and structured, it is straightforward for me report our findings with respect to the cognitive framework outlined in Chapter Two.

ATLAS.ti only helps us to manage, trace, and connect pieces of ideas. My evaluative discussions are still reported directly from the original data. For example, to evaluate subjects' feedbacks on patterns' graphic-based representations, I imported all the documents and extended-codes related with basic code *represent-graph*. Through looking at their linkage networks, I realized that they were mainly divided into two parts: diagram and animation. Then I went back to the original files, listened to the audio, checked all the connected files (such as images and scripts), and wrote the evaluations for these two parts separately. I believe *thick description* of ideas is more useful to represent the qualitative research work. Limited by the length of this work, I only select the most obvious segment from projects to report.

10.2.6 Validity Verification

In these two studies, I used participant observation method, informal interview, and journaling study techniques to search for evaluative information. Four trustworthiness criteria were carefully considered:

- *Credibility*: During the two workshops, I worked and interacted with subjects in the same environment for a full seven and three days. After each conversation, I conducted member checking, asking for re-clarification, summarizing the main ideas discussed, and asking for the subject's confirmation. This is a crucial step in maintaining credibility.

- *Transferability*: Although problems, solutions and opinions of individual projects are not directly transferable across different cases, I tried to create an extensive and careful description for each of the observed projects. This process made another coder's analysis process easier.

- *Dependability*: With both multimedia data and thick project descriptions, I could provide sufficient information for conducting the inquiry audit.

- *Conformability*: Two kinds of triangulations were involved in the data analysis: data triangulation across different projects and different events, and investigator transvaluation. These two methods were used to ensure the conformability of evaluative study.

I used the verification strategies of *collecting and analyzing data concurrently* and *thinking theoretically* to ensure the rigor of study. Apart from conducting *member checking* during the short conversations, I also tried to read and digest the collected data during the breaks between conversations. Although such digesting processes were not serious data analysis, I was able to form a mutual interaction between what is known and what I needed to know in the next conversation. This made the iterative interaction between data and analysis possible. Before the evaluative study, I also built up the cognitive framework to explore the data. I was focusing on looking at the users's feedback on the cognitive aspects of patterns and whether or not patterns were supportive. With such a structure built in mind, I checked and rechecked my data constantly and inched forward without making cognitive leaps.

10.3 Evaluative Discussion: Being Cognitive

I conducted two rounds of evaluative discussion for two different directions. The first round focuses on the cognitive construction of design patterns. The second round of evaluative discussion focuses on whether or not design patterns can support design activities. I used three perspectives, *context*, *representation* and *work analysis*, as a cognitive framework to help me understand using parametric design patterns in design process. To enhance the efficiency and communication of design activity in parametric CAD systems, the structure of design patterns and their related activities should meet the cognitive design requirements. The following discussions are based on the activity facts released by the data. When I introduce the projects and designers enrolled in the evaluation studies, subjects are referred to by a two character abbreviation in the text to ensure the privacy of individuals.

10.3.1 Contextual Aspects of Design Patterns

In comparing three HCI approaches to the study of context, *activity theory*, *situated action models*, *distributed cognition*, I chose activity theory as the main theoretical perspective for discussing contextual issues around pattern structure, since this theory commits to understand things from the users' points of view and aims to search for persistent structure in activities. According to the activity theory, *persistent structures* stretch across situations and activities through time and space that cannot be properly described as simply an aspect of a particular situation [118]. In the data analysis, across different activities we search for persistent structures of pattern use. To narrow down and focus the research scope, we emphasize on analyzing the data in terms of the pattern structure components. Across different patterns, different projects, different users, and different activities, we investigate the similarities in acceptance of pattern components.

I also used distributed cognition to understand the social context of pattern use. Distributed cognition asserts as a unit of analysis a cognitive system composed of individuals and the artifacts they use [104]. The design of pattern repository serves as the communication platform for pattern users and parametric design patterns. In this part of data analysis, we shift the focus to the repository evaluation because distributed cognition focuses on the transformation and coordination between individuals and artifacts. During this stage of analysis, we pay more attention to information such as how ideas transfer from the system to users, from users to users, and how information is carried on during the time.

At the current stage, I did not use situation action models in evaluation studies because of the following reasons:

Contingency focus: While comparing activity theory with situated action models, Nardi noted that "in activity theory, the structuring of activity is determined by human intentionality before the unfolding in a particular situation; in situated action, activity can be known only as it plays out in situ [160]". In other words, situated action models emphasizes the emergent and contingent nature of human activity. Design goals and plans cannot even be realized until after the activity has taken place [218]. In my research context, parametric design is mainly goal-driven activities. Workshop participants prepared detailed design proposals to attend the events and the design intentionality is strongly embedded in their behaviors. The contingency focus of situated action models is not suitable in such a context.

Empirical methods: The research methods used in situated action model are difficult to apply to investigate real life design process. For example, Suchman and Trigg cataloged their empirical study methods are "(1) a stationary video camera to record behavior and conversation; (2) "shadowing" or following around an individual to study his or her movements; (3) tracing of artifacts and instrumenting of computers to audit usage, and (4) event-based analysis tracking individual tasks at different locations in a given setting [220]". They took interviews as "more or less unreliable accounts of idealized or rationalized behavior [218]". Although it might be possible to video record the whole real project design process (even for days), it is very difficult to 'shadow' and trace all the related activities without interrupt the designer's thinking and creation flow.

Goal of evaluation: Interpreting and using patterns in practice is a long term reflective thinking process. The main goal of this evaluation study is to investigate early signals of pattern use in real design practice. We are more interested in persistent structures across activities than outlier cases. Situated action models' interests on "moment-by-moment interactions between actors, and between actors and the environments of their action [218]" do not match our current evaluation goal.

Considering these three reasons, I chose to only use activity theory and distributed cognition to construct the contextual cognitive evaluation of patterns in use.

10.3.1.1 Contextual Components in Pattern Structure

Using widely accepted OOP [83] and UI patterns [230] as a point of reference, we proposed a structure for describing parametric design patterns and applied it on all of thirteen patterns. That structure included the properties of *Name*, *Diagram*, *Intent*, *Use When*, *Why*, *How*, *Samples*, and *Related Patterns*. Our first evaluation objective was to investigate whether this structure supported users in achieving their goal-directed actions.

During the two workshops, participants could access our pattern repository online. We also distributed a DVD on which all the thirteen patterns were packaged. Most of the pre-training workshop participants copied the content of the DVD to their laptops and were able to read the patterns and try the samples directly. We noticed that subjects deploy different strategies in using patterns for working and communicating. There are some feedbacks upon the pattern structure.

***Names* identify transferable ideas in communication:** From the second day of the SmartGeometry 2008 pre-training workshop, participants started to use pattern names in their conversations. For example, when I started the observation study and interviewed my subjects about their planned projects, both MT and CB mentioned several pattern names in their interviews to describe their plan. Those names served as a shortcut for us to dive inside the detailed discussion directly. Since MT mentioned the *Transformer* pattern, I asked her which part of the model was rigid and what the transforming path looked like. She showed me her prepared diagrams (upper images in Figure 10.4 directly and tried to explain which pole was rotating on a certain plan and the relation between α and β angles.

When subject GF was discussing the collaborative twist tower project with his teammate HC, HC advised GF to use two patterns, *Place Holder* and *Increment*, for their model transition. HC was more familiar with the pattern repository, but was so focused on his own part that he was not able to explain these patterns to GF in detail. Subsequently, GF searched for the named patterns in the repository and learned about them himself.

However, we observed that some participants misinterpreted the meaning of the patterns and used the names incorrectly. For example, pattern *Recursion* was mixed with a two-dimensional loop in the function, and setting a global variable was understood as the *Controller* pattern. In some cases, participants also confused the name of the sample with the name of the pattern, or created new names based on the appearance of the sample. For example, "roof pattern" and "paper folding pattern". We believe that this is not necessarily bad because certain new names were so vivid that participants immediately associated the name with the "officially" recognized pattern idea. Concepts of the patterns are much more important than the names, although it is clear that, at least in a team, having a unique name was important.

During the pattern authoring, we referred to the Gamma et al.'s software engineering pattern naming style [83] and collaboratively named our patterns with explicit and succinct nouns. Considering the users' feedback, we found most of the patterns had been smoothly integrated into the user communication. However, we also realized that pattern names acted as an essential role in the discussions. We should pay attention that whether or not there are some "natural" names, terms, or metaphors emerging through the language of users so that we can use to improve some existing names or name new patterns.

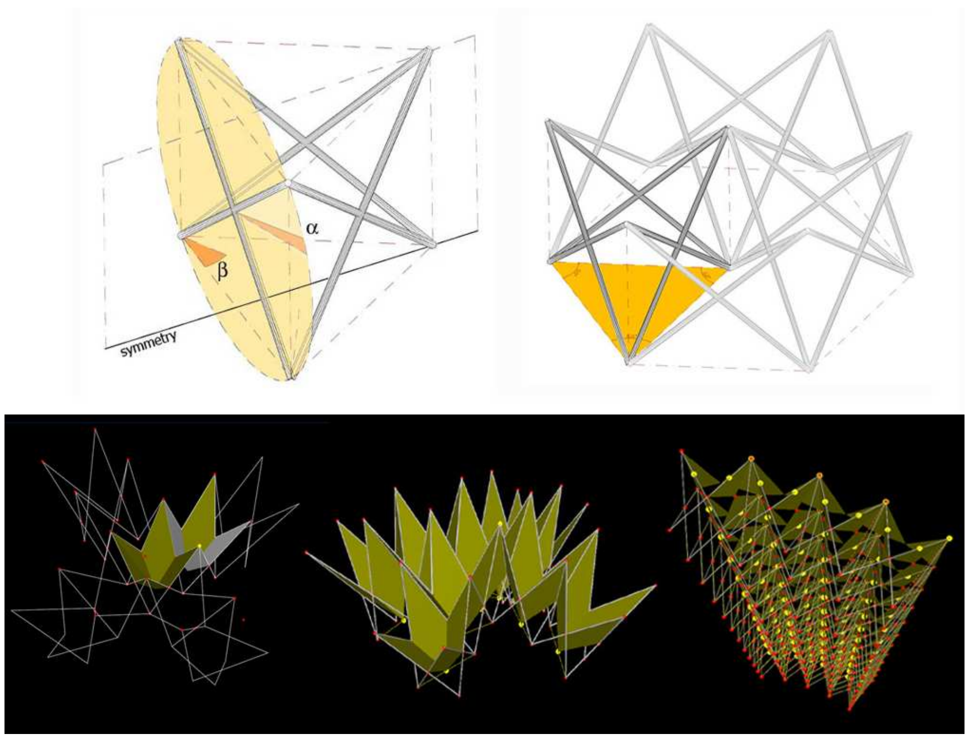

Figure 10.4: MT in SmartGeometry 2008: Rigid Foldable Structure.

***Samples* are critical elements in pattern exchange:** During the interviews or coffee breaks, when participants started to talk about patterns with us, they quickly engaged in detailed discussions of certain pattern samples or even certain transactions in samples. For example, the first chat between JG in ACADIA workshop and me was to answer his question about a sample script in the *Reactor* pattern (This sample was outside of the pre-training tutorial. JG looked into the sample to check if it was appropriate for his project). "I can see some direct connections between my model and the *Vector Field* sample," JG compared his sketches (in Figure 10.5) and the sample side by side to me:

> "See? all the line segments rotate to target the reference point in your example. In my model, my lines not only rotate but also lifted up to target that point. It is logically the same! Show me how can I change the scripts to do that."

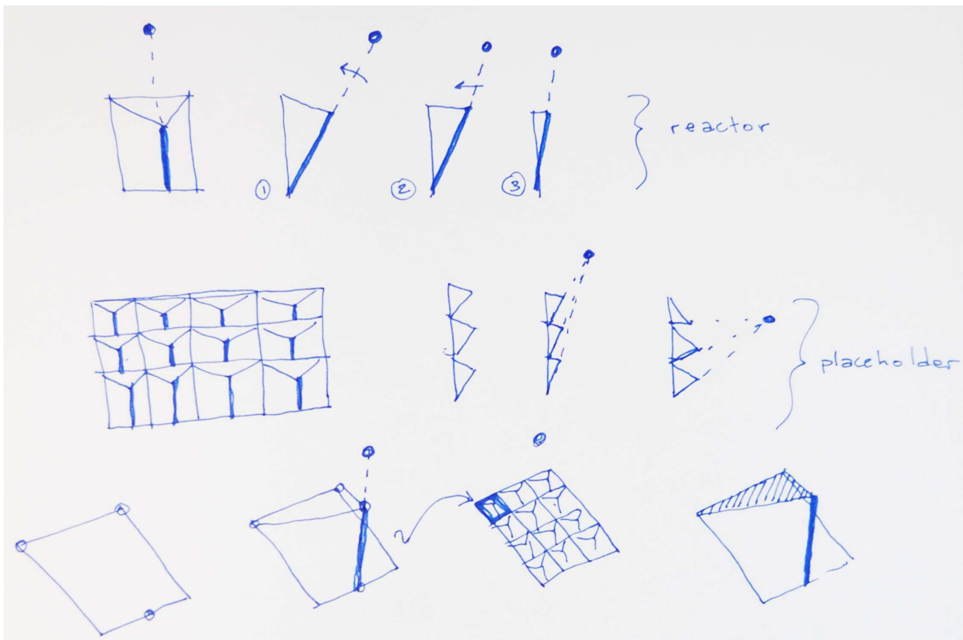

Figure 10.5: JG in ACADIA 2008: Responsive Vector Field.

The dynamic aspect of activity theory is that the constituents of activity are not fixed but can change as conditions change. Thus, the context is not an outer container or shell inside of which people behave. Rather, people consciously and deliberately generate contexts (activities) in part through their own objects. In JG's responsive vector project, our discussion quickly fixated on several lines of code in a sample script and focused on the discussion of specific issues - how to move the endpoints of lines away from the XY plane and whether or not the length of line segments should be fixed. In JG' s sketches, pattern names such as *Reactor* and *Place Holder* were annotated. He was able to work on certain small problems without losing the context of the whole project.

***Related Patterns* need further interpretation:** Pattern families have their own hierarchies. Although many pattern authors aim to build pattern languages, the "grammar" dimension was typically inexplicit. In this study, the hierarchy of our patterns has not emerged clearly. We used *Related Patterns* to annotate the connections among different patterns. However, this component caused confusions.

In the second day of the ACADIA workshop, subject JC questioned me straightforwardly, "What do you mean by related patterns? Why only several patterns have related patterns? Should they be used in a sequence? or, is one a component of another? If not, show me how to do that..."

In the SmartGeometry workshop, GF was confused by the *Jig* pattern, which is related to the *Controller* pattern. In our authors' understanding, jigs not only serve as the simple abstract frameworks to isolate structure and location from geometric details, but also aim in controlling some details of the structure. So they can be a kind of implicit "controller". GF thought jigs were a fixed structure and was not aware that other components could be adjusted freely until I edited one sample to demonstrate the possibility.

There is not enough explanation provided to introduce *Related Patterns* and their connections. Such underdeveloped patterns sometimes confused or misled our users.

According to the data analysis, feedback received from subjects emphasized the *Name*, *Samples* and *Related Patterns* properties of the pattern structure. There is a persistent structure in participants' communication: discussing strategies with pattern names (or sample names and features) and then diving into problem solving with one pattern sample directly. On the other hand, in the pattern structure *Related Patterns* serves a negative role to provide contextual information.

10.3.1.2 Pattern Repository Facilitates Collaborative Manipulation and Information Dissemination

In the theory of distributed cognition, Hutchins [105] discusses "collaborative manipulation", the process by which we take advantage of artifacts designed by others, sharing good ideas across time and space. From the collected data, we saw different forms of collaborative manipulation and information dissemination arising from the pattern repository and its publication:

Online Publication Advertises the Repository: We tried to encourage discourse around the pattern authoring process through both face-to-face and online interaction. We posted the URL to the pattern repository in different GC user forums and invited individuals from different designer groups. The response from the pre-training workshop indicated that the outcome was quite positive. Among the 70 participants, approximately 20 of them had visited the repository before. Some of them learned about it from discussion forums. Some heard about the repository from paper presentations made at CAD conferences or invited talks. One case is of particular interest. Subject MK found our repository through a Google search,

> "I learn software through self-teaching. As usual, I was trying to teach myself the software GenerativeComponents two months ago. I can't find any good tutorial materials from their public website, I mean, with examples to follow. So I did a Google search for .gct files. The website which holds the biggest amount of .gct files is your repository. The site is simple and clear, for me."

Diffusion within Groups: In the SmartGeometry 2008 workshop, there were 154 participants and 24 senior tutors. More than 70 participants and 7-8 tutors (who arrived earlier) attended the pre-training workshop. In the first day of the formal workshop, we had to introduce the pattern repository and offer the pattern DVD. From the second day, there was no such need. Many participants introduced themselves and said that they had heard about this repository and had copied the DVD content to their laptops. For example, during a coffee break, I mentioned our repository to architect JS. She quickly realized that she already got it,

> "No thanks! No need for the DVD. I have already got your pattern files. XXX told me that they were good stuff..."

During the middle of the workshop, one of the SmartGeometry directors heard about the pattern discussions from the larger group and came up to us to ask for detailed information about the repository.

Suggestions of New Patterns: We authored patterns based on the practice observation data gathered in 2007's four GC events. In 2008's events, our patterns also inspired new ideas, not only from participants but also senior tutors who had instructed GC workshops for years.

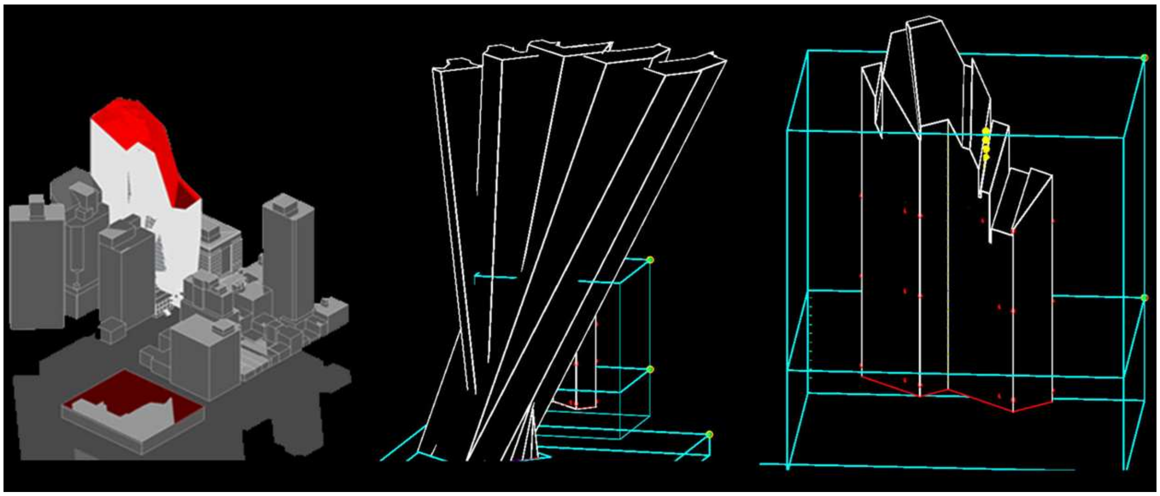

Figure 10.6: DS in SmartGeometry 2008: Sunshine Shaping Tower Project.

- Subject DS was working on determining the shape of building through analyzing the sunshine and shadow relations. He failed many times when he was trying to determine the proper building elevation through calculation. At the end, he got the accurate shape through uniting all possible sunshine rays and trimming the tower as a whole (Figure 10.6). DS argued that it was an approach of thinking reversely in parametric modeling and recommended this new pattern idea to us.
- Subject KT suggested a pattern idea "mediator" in which item B can inherit some (not all) properties of item A and connect with item C. In such a pattern, all the properties of A could be preserved, and a connection between A and C could be built up. HC describes the idea briefly to me during a lunch and I wasn't able to collect any detailed sketches or examples for this idea.
- Senior tutor SB asked us if we have a pattern "data organizer". In several cases he met in the group, he had to re-organize the data structure of an index or a collection, such as sorting a linear index of random points to a two-dimensional collection based on the Z translations of points. When another tutor MS dropped by our conversation, he said he was working on something similar and demonstrated what he was able to do through a small GC file.

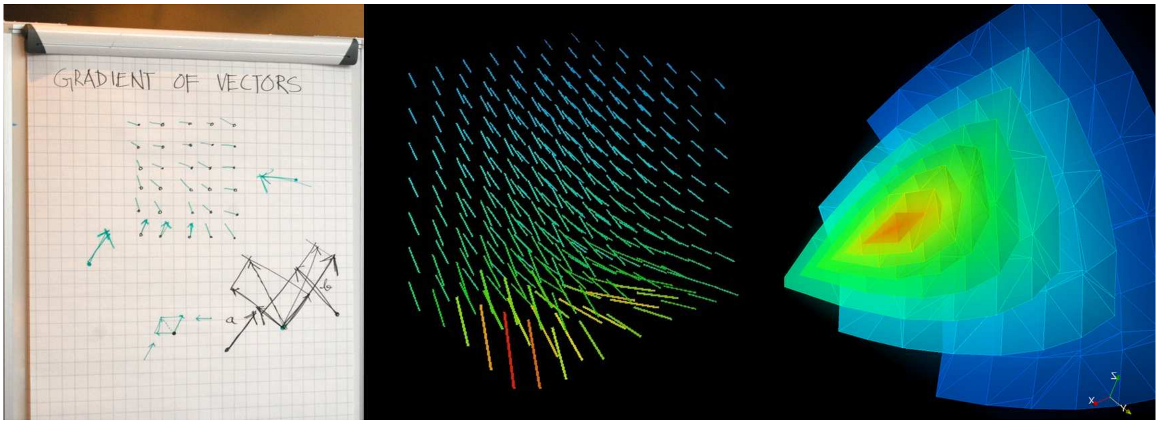

Figure 10.7: KD and CB in SmartGeometry 2008: Potential Pattern *Gradient of Vectors*.

- Senior tutor KD was responsible for the environment group. He realized that there were common needs to calculate a collection of vectors based on more than two influence sources. On the room's white board he drew the diagram of "gradient of vectors" (left image in Figure 10.7) and distributed his sample file around the group. We got to know this through subject CB. CB did some further implementations based on KD's sample - analyzing the gradient of vectors from a three-dimensional perspective and representing such as gradient through colored layers (Figure 10.7). CB said that it was really a useful pattern idea.

10.3.2 Representations of Design Patterns

Scaife and Roger's theory of *external cognition* makes the distinction between language-based external representations and graphic-based external representations [197]. Parametric design patterns employ both kinds of representations in their structure. How do these representations work in the real design practice?

10.3.2.1 Language-Based Representations

Participants mainly scanned through the pattern repository for solutions. Samples provided concrete code that help participants, as amateur programmers, to understand the overall pattern by following its transactions step by step. We devoted much effort to writing the explanatory text: components of *Intent*, *Use When*, *Why*, and *How* are all such text. However, we realized it was the *Samples* that actually bridged between specific design ideas and the generality of patterns and helped participants adapt in design practice.

During the short-term conversations, participants mainly talked about the details of samples. Even while discussing the samples, we noticed that most participants spent much more time viewing the animations of samples than reading their text explanations. MH from the UN Studio told me that he scanned through the animations in our repository to look for useful scripts:

> "I want to enjoy the writing part. But honestly, finding what it can do is much more important for me than finding how to do it. I mean, in that environment, I have to find quick solutions. Basically I scanned through all of your animations first, then I may download a couple of them to see how you made it work. Only if I failed understand the concept or the strategy, I would return to read the text explanation. I think that only happened once or twice."

"A picture is worth a thousand words" is an adage from an old proverb. In light of the fact that the user group in question is comprised of visual-oriented designers, it is not surprising that they would scan the graphical representations rather than read. Also as MH emphasized, working in such an intensive environment, their only goal in using the repository is to search for meaningful solutions. Graphics are comparably more self-expressive than text. Text-based script files are quite popular because they are directly related to solutions. While preparing these scripts, we tried to annotate each step with proper names, arrange a balanced symbolic model view, and cleaned out all unnecessary components. These efforts had been acknowledged because most of participants found the scripts clean and easy to follow. Some participants even copied and pasted part of the sample scripts directly into their projects.

10.3.2.2 Graphics-Based Representations

In the scope of graphics-based representations, the classes of animations and static diagrams are considered distinctive, in so far as they have been identified as having different characteristics in the literature [197]. From the cognitive interactivity perspective, animations have the benefits of cognitive tracing and interactivity. Diagrams are comparably easy to produce and useful because of their explicitness and visibility. In each pattern, we used animations to represent the outcomes of pattern samples and a diagram to present the general pattern idea.

Sample Animations: Shortcut for Solution Scan When we were designing the pattern repository system, we wanted to present the patterns online in 3D. An "ideal" solution would have been to allow users to manipulate the samples directly online, but that was developmentally expensive. As a compromise, we decided on animated transitions with Macromedia Flash™, as animation is a well-established way to guide a user through process and change [22]. We devised a solution to have the Flash file generate an animation by automatically picking up a series of images from a folder to ease creation and editing by designers. Simple as it is, it presents a series of rapidly changing static displays and gives the illusion of temporal and spatial movement.

From the observation, I can see participants spent more time on viewing the animations than reading the details of the scripts. Just as MK in the ACADIA workshop said, "your animations make the 'scanning through the patterns' possible." Designers can quickly judge which part is related and which part is useless when they scan through the animations. A lot of reading time was saved and most participants really appreciated this feature. The animation here is proved to be useful for synthesis.

Pattern Diagram: Less Useful for Meaning Exchange Among the eight parts of a pattern structure, *Names* and *Samples* have been found to be very useful. Participants also spent time reading textual descriptions (*Intent*, *Use When*, *Why*, *How*, and *Related Patterns*) when they wanted to understand our interpretation of patterns. One subject even commented that the structure was very clear. However, we did not find any evidence favoring the pattern diagram. These diagrams provide a graphic representation of a pattern that suggests is structure and meaning. This seems ironic because both the diagram authors and readers are visual people. In comparison with the animation representation, the diagram is obviously less accepted.

As pattern authors, one problem we already noticed was that the style of diagram drawing was not consistent. These diagrams were drafted by three different authors and the meanings were sometimes very abstract. We did not have an opportunity to standardize the style of diagrams. Whether or not a set of consistent and meaningful diagrams will gain more readers is unknown.

The literature around external cognition also describes a "cognitive interactivity" analysis framework to exam the interactivity of technologies. At the current stage, our patterns are still presented in a static form, so the interactivity would be the system's future conceptual design consideration.

10.3.3 Cognitive Work Analysis of Using Patterns

Kim J. Vicente's Cognitive Work Analysis (CWA) recognizes the essential adaptive role that humans play in safely controlling the unanticipated in complex socio- technical systems [236]. More importantly, it aids in distinguishing between goal-directed actions and work-around activities. Nearly all the subjects I observed have clearly proposed tasks to work on. As such, we expect that their work in the workshop is largely goal-directed. In fact, the observational data indicated that conditions varied significantly. Here, I introduce this framework into the data analysis to demonstrate two distinctive cases in pattern using practice.

10.3.3.1 Goal-Directed Activities

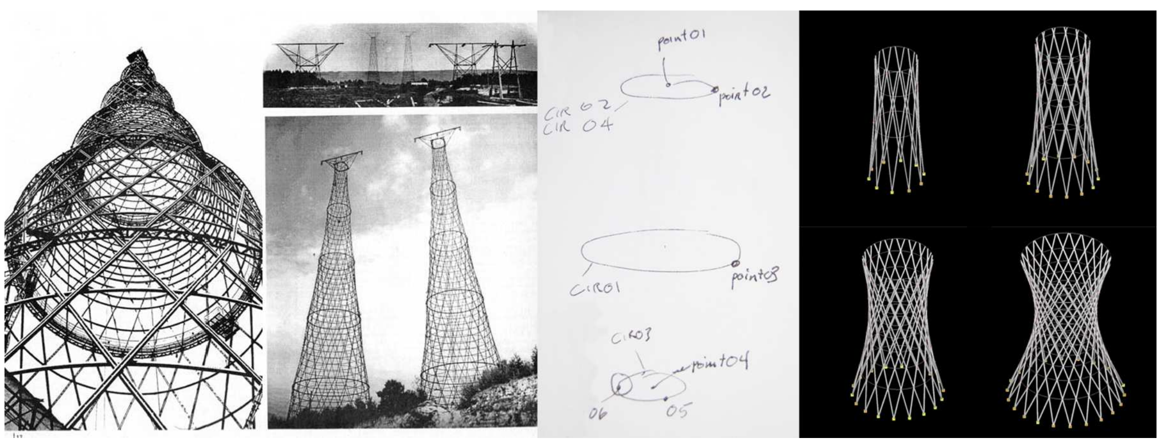

Figure 10.8: PV in ACADIA 2008: Hyperbolic Grid Tower.

Architecture educator PV arrived at the ACADIA 2008 workshop one day later than other participants. He only had two days to learn GC and work on his proposed project. He was teaching the a course called 'Architecture 324 - Structure II' in 2008 Fall semester. In his statement of interest, he compared trussed towers from the 19th century through to the 1950s, and proposed to build a parametric model of Vladimir Suchov's hyperbolic grid tower (left images in Figure 10.8).

- *Work Domain*: Being a structures course instructor, PV was very familiar with the algorithm behind the hyperbolic grid he planned to develop in GC. He also gained some C programming experience more than five years ago. In the 20-participant workshop, although he came a little bit late, he had four tutors to guide him through the basic GC functions. We demonstrated the digital repository of design patterns to him at the first stage to show the basic potentials of the software.

- *Control Tasks*: PV aimed to build the whole grid tower with an elegant script file. In the model he would be able to control the number of stacks, height of the tower, and density of the crossed grid. Considering his limited time in the workshop, he said that he would just try to finish one stack at the first stage.

- *Strategies*: PV accomplished the task through adopting the "Drill" sample code from the *Increment* pattern. Before realizing the usage of patterns and samples, he met some obstacles on the way. PV started to draw sketches on his notebook (middle images in Figure 10.8) before drafting models in GC. Noticing that the starting position of T parameter is very essential, he stopped me to ask if it was possible to rotate the $T = 0$'s position directly. After realizing that he had to create a new coordinated system to achieve that, he gave up the idea because he thought scripting would be a more elegant approach. He started to write some scripts as a function, but some of his coding could not be interpreted by GC. He opened our pattern repository and started to read *Increment* samples. "I can see a lot of commonalities between your samples and my model. More importantly for me, I need to know how you wrote the scripts." He downloaded the script file for the "Drill" sample and started to edit it. Later on, he was also attracted by the *Controller* pattern and planned to separate the controlling part from the structure itself.

- *Social Organization and Cooperation*: PV's first coding experience was not very smooth even though he had prior programming experience. On the second morning, he constantly asked for help from tutors to debug his code. At the same time, he also edited some scripts and made suggestions to other workshop participants.

- *Designer Competencies*: On the second day's afternoon, he finished his parametric tower stack with full control of radius size, height, grid density and grid rotation angle (right images). At the completion, he created a Flash animation for the tutor to demonstrate the infinite variations of this flexible stack in the ACADIA conference.

In the questionnaire (sent one month later), he wrote to me,

> "The script I wrote in the workshop was developed from the *Controller* and *Increment* scripts. If I get time over Christmas break I would like to expand that tower script and develop a GA optimization based on that too. I worked up a little trial to design bridge trusses generated with GC. The truss parameters are contained in an Excel file and the geometry output from GC is analyzed and members sized with STAAD-pro. Then a GA optimizes the trusses based on the weight determined by STAAD. The selection for the GA breeding was done by students in my class based on visual preference. There is a web site that was used for the breeding and where you can view the outcome (link provided). Also, foster.gct which is the script I wrote for this, can be downloaded there."

10.3.3.2 Work-around Activities

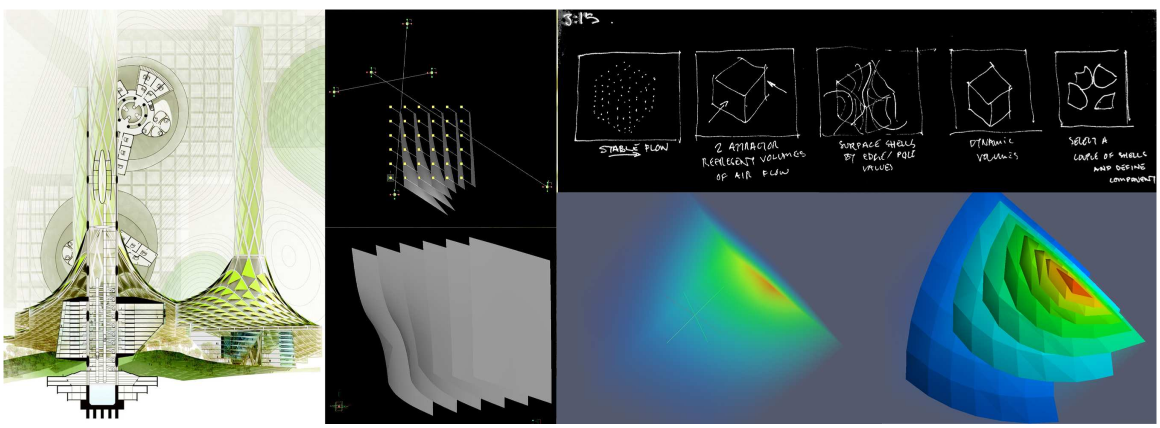

Figure 10.9: CB in SmartGeometry 2008: Green Desert Mine.

Being a lead architect, CB was invited to work on the Green Desert Mine project located between the Nile river basin and the eastern Sahara (left image in Figure 10.9). He planned to focus on "utilizing environmental parameters to drive the complex canopies and core solar towers forming the basis of the economic and urban tectonics in the hopes of achieving autonomous systems and rich forms of biodiversity." The main goal to allow GC to produce variations on the design. Although he started to teach himself GC several months ago, CB registered the pre-training workshop to learn more functions.

- *Work Domain*: After pre-training, CB worked in the Environmental group for four days. This was the largest group among all the six workshop groups, with 35 participants. Although there were five professional tutors circulating around to consult, it was still difficult to get instant help. Participants sat around six big tables. At the end of each day, every participant was required to report to the group on his or her progress. Although it was a nice collaborative setting, most participants were focused and seldom had time to chat with their neighbors.

- *Control Tasks*: CB originally planned to develop a model of a solar tower. He changed his mind, and decided to work on a more environmentally responsive part - a breathing canopy because “the tower should be built upon the result of the canopies.” Instead of working on the overall canopy structure, he selected to analyze one piece of it and evaluate it in Ecotect.

- *Strategies*: There was an Ecotect tutorial arranged in the first day of workshop. On the second day, CB started to work on one piece of the canopy. Illustrated in the middle two images in Figure 10.9, CB employed the *Reactor* pattern and created a set of BSplineSurfaces that were responsive to an external moving point. On the third day, CB decided to jump outside of the two-dimensional array and consider the canopy in a three dimensional box. In order to do that, he first sketched the possibilities (upper right image) of outside vectors' effects. Although he was able to sketch wind and direct forces, it was hard for him to develop those easily in GC. In the afternoon, he was able to develop a colored cube that showed the gradient change of forces with the help of tutor KD.

- *Social Organization and Cooperation*: The five professional tutors were very knowledgable in both GC and architectural design. However, getting their help required that you explain your project and current problem to them clearly every time they sat down in front of you. CB complained to me that, “I had explain my problem more than four times this morning, but nobody can tell me directly how to solve it!” CB was also constantly inspired when other participants presented their projects. However, few strategies can be used directly to solve his current problem.

- *Designer Competencies*: The final day was a frustrating day for CB. He has finished the responsive canopy on the second day and gradient vector cube on the third day,

> but did not know what to do next. It took a day for him to package all the outcome of the work with the original design proposal. But there was such a huge gap between them, CB found it hard to put them together and make "a complete story".

Cognitive Work Analysis revealed many details of what took place in the context of the workshop design practice. CWA analysis on PV and CB's projects reveals two distinctive cognitive processes. Although only having two days, PV had a very practical plan and achieved what he wanted by the end, albeit in the form of a very simple structure. CB's project was much more complicated. He made the choice of narrowing the task down and working on specific aspects. However, he didn't set up a clear goal for what was to be achieved. During the four days, he learned in part from the pattern repository, in part from Ecotect, and in part from the tutors. But he was constantly interrupted by new ideas and new directions. Also different tutors helped him with different strategies and coding styles. The workshop was a great experience for him to learn and explore. In these two cases, the help tutors were able to provide varied. Because of the smaller size of the ACADIA workshop, PV was able to get support in time from tutors. For CB, waiting for tutors wasted a lot of his time.

From the first round of evaluative discussion, I learned more about pattern use from the data, including perspectives of context, representation and cognitive work analysis: in the pattern structure, there are some components (such as *Name* and *Samples*) providing contextual information to users while some (*Related Patterns*) actually confused users. The structure of the pattern repository generally supports design practice and communication. Unsurprisingly, graphic-based representations are more intuitively supportive than language-based representations. But diagrams is less useful. Also as the result shown from the CWA analysis, different designers' plan and external environments lead to different outcomes and effects. These results suggest areas for the pattern repository project to make improvements.

In order to observe, understand, support, and augment users' design practice in parametric design system, I chose patterns as the approach to model and represent the designers' learning and working process. The initiative reason of the choice is that pattern provides an explicit and sophisticated structure. Through the literature review in Chapter Four, I understood the variety, distinctive characteristics, and method missing issues of pattern approach. The cognitive evaluation shows that different components in the pattern structure can demonstrate different usability values in the CAD learning and working processes. Especially, providing ample and easy-accessible examples and presenting them vividly turned out to be very useful. Although Alexander's 30-year-old formulation structure is rigid, but it can be robust, succinct, and communicative in use.

10.4 Evaluative Discussion: Being Supportive

The above evaluation discussion focuses on the cognitive construction of design patterns. Following is the second round of evaluative discussion focuses on whether or not design patterns can support the several key issues in design activity: parametric tool learning, building up expertise of scaling up, and design collaboration.

10.4.1 Patterns Support Tool Learning

The repository of design patterns was first introduced to designers as a learning object resource when we used it in the workshops. Before it was developed, GC pre-training workshops started by demonstrating menu functions one by one. Then the tutor would lead the group through one or two complicated examples. After such a tutorial, some students were able to understand the basic logic of the application, and start to work on their own projects. However, Most students were either left lost or limited by the functions used in the examples, since the length of pre-training workshops were usually short that limited to 1 or 2 days. How to teach a comprehensive (not only structurally but also logically) software in such a short time frame is always a challenge for GC workshop tutors.

From the SmartGeometry 2008 workshop, my senior supervisor Dr. Woodbury started to lead GC tutorials with patterns. We only teach four examples: *Punch Holes in a Solid*, *Variable Place Holder*, *Select and Report Points*, and *Four Curves by Scripts*. All of these examples are simple and easy to follow. Students can get a copy of these four scripts and follow the transactions step by step without supervision. More importantly, each of these examples links to two or more patterns. After getting to know how to do, students got the opportunity to learn why to use the strategy and how to use it in other contexts. We tried to leave enough free time for students to explore related patterns among the example demonstrations. Aiming to present students more potentials of the parametric tool, we tried not to stress them with specific functions. Through following given samples, many students understood the fundamental logics quickly.

10.4.1.1 Reflective Thinking

Reflective thinking is a careful, deliberate mode of thinking that helps us make sense of what we experience and what we know [165]. It was a great method to compose new knowledge. Students need a mental space to add new representation, modify the old ones and compare the two to learn. From participants' feedback, our design patterns were able to somehow achieve this goal:

- *Useful for self-teaching*: PV missed the first day pre-training. When he came to the workshop, other participants had started to work on their own projects. Tutor KD walked him through the basic functions of GC and PV started to run and test the four tutorial examples and pattern repository. During the morning, he asked tutors several questions: some about linkages between transactions, and others about how to implement a certain function. Then in the afternoon, he had basically caught up with the group and started to worked on his hyperbolic grid tower.

- *Helpful for understanding the implicit logic*: In the ACADIA workshop, JG was impressed by the example of *Variable Place Holder*. However, he was still quite confused why he must use "polygon01.Vertices[0]" instead of "point01" to define following structures. After asking the tutor once, he hesitated to ask other tutors to repeat the reason. So he opened the repository to read other samples in the *Place Holder* pattern. "I read the text and the samples carefully. Finally, I understand what you guys mean by 'proxy object'. It is basically a transmitter to transfer complex structures, right? In order to do that, you need make the transmitter as simple and as clear as possible. So any definition should be based on the transmitter." Although JG failed to understand the trick from the tutor's explanation, he was able to figure it out with the patterns' help.

- *Inspiring for problem solving*: In DS's sunshine shaping tower project (Figure 10.6), he failed to calculate the proper building elevation. During a frustrating break, he went back to read through the tutorial examples. In the *Punch Holes in a Solid* example, we had to unite all the cones into one solid to punch through the curved board. Although all the cones belonged to a set, in the model, they were still treated separately. This "Unite + Cut" strategy inspired DS: he united all possible sunshine rays and trimmed the tower as a whole. He was so excited with this trick that he recommended it to me as a new pattern idea.

- *Stimulating for new explorations*: MT's rigid foldable structure is quite complicated (in Figure 10.4). She said she was excited to see the *Transformer* pattern because she thought it was something directly applicable. However, after reading through all the sample scripts, MT did not find anything that could be employed in a straightforward fashion. She then settled down to read the pattern explanations and samples. "As you said here (pointing to the webpage), defining the rigid body and determining the transformation paths are the two key steps of transformation. I am working on the path part now. Your pattern is helpful to analyzing the problem."

Parametric designers do not always have the luxury of a professional tutor around to consult. Design patterns provide a resource and the space for them to read, think, digest, explore and communicate. Such a thinking process is reflective thinking.

10.4.1.2 Scaffolding

I tend to understand the notion of scaffolding at two levels: it refers to ways the software tool itself can support learners rather than only teachers or peers; and, there is a delicate negotiation between providing support and continuing to engage learners in the learning process.

The first level is very basic and the repository of design patterns can generally meet its requirements. GC uses the symbolic model with graph visualization for users to navigate his/her own transaction process freely. Apart from that, the modeling process is recorded by the linear transaction-based structure. Both of these two features provides a foundational structure for scaffolding. Patterns themselves can be viewed as a scaffolding strategy. By providing explicit examples and showing, stepwise, how these relate to more abstract ideas, the hope is that patterns can enable people to eventually construct such useful abstractions themselves. From the discussion of reflective thinking, I can find evidence that using design patterns to learn the GC software is possible and practical.

Currently, it is hard for the pattern repository to achieve the second level of scaffolding. Being a information storage, the repository aims to provide a platform to author, share, and represent design patterns. There is limited online interactivity provided by the system. It is able to record critics' annotations and add authors' contributions, but it does not provide a space for visitors to anchor their learning process. When the visitor's skills has been enhanced, the repository system will not hide basic patterns and add more advanced patterns by itself. Basically, it is an online information sharing repository instead of a personalized learning tool. Whether or not additional functionality should be developed in the repository will depend on how design patterns develop in the future.

10.4.2 Patterns Support Expertise of Scaling Up

One design goal for the design patterns was to improve expert work practices through reflecting the community's own problem solving strategies for reuse and adaptation. Our first observations revolve around the frequency and type of pattern use, both individually and between users. We identified three scenarios of pattern use in practice: *directness*, *unawareness* and *serendipity*.

10.4.2.1 Directness: Use Patterns to Solve Problems

Some participants were very comfortable with design pattern concepts. They either had a clear plan of how to use them in their projects or recalled some experience in using related concepts. For example, in our first interview, subject MT said:

> "I am trying to achieve geometric construct movement and different morphologies within the possibilities of constraints "What is this on your screen?") This is a single unit. By studying these single units, I am trying to propagate these deployable parts in the overall structure. What you have seen now is just one of the configurations, so the next step would be more complicated. I heard of your design patterns was in Delft in a lecture Professor Woodbury gave us. Then I started to look at your website...For this project, I think it would be many as I can use: the *Transformer* approach, the *Controller* approach and definitely the *Place Holder* approach."

Figure 10.4 shows images of MT's project: the upper images were the ones we mentioned during the first interview. The bottom images were her final outcomes with GC. From coding her scripts, it is clearly that she had made full use of several patterns. Although all the *Transformer* samples were much simpler than her project, she was able to learn the logics of separating rigid components and defining the transforming path to solve the problem. She mentioned that she would use the elegant foldable structure on a complex gym roof surface her team was currently designing.

During the SmartGeometry pre-training workshop, subject BB came up to us during a coffee break and recommended himself as a subject to us.

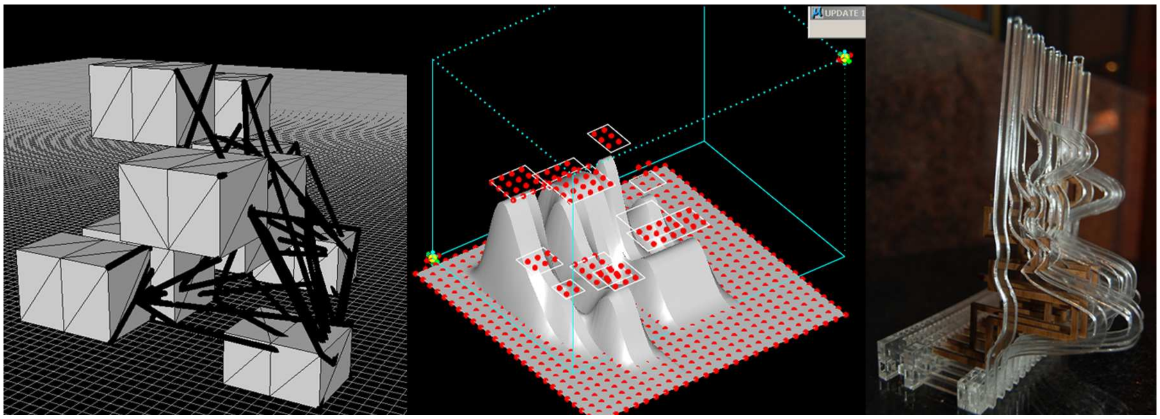

Figure 10.10: BB in SmartGeometry 2008: Flexible Skin to Wrap Blocks.

> "I had never heard of your patterns before. But today when I looked through the repository, wow, I really can point out several of them I had already used in my previous projects. When I was using them, I didn't realize that they were strategies, I mean, patterns. ... I am going to at least use the *Projection* pattern in the project tomorrow. It is about a skin, trying to wrapping up the structure behind it and I am also going to design proper components that can populate the skin that can do some sort of solar tracking. It is a *Place Holder*, right?"

BB mainly used the *Projection* pattern in his project. He met some problems during the implementation. Projection aims to produce a transformation of an object in another geometric context. There are three parts in the projection structure: source object, projection method and receiver. In BB's project, the skin was the result of projection instead of the receiver. BB was confused by this part and spent a whole day to fix its logic. Finally, he was able to generate a skin to wrap up the imported structure (left image in Figure 10.10) provided by his school design team.

We note that in both these cases, the designers were familiar with the concept of explicit strategies for their individual tasks. The interpretation and samples in the patterns demonstrated the possibilities of strategies. By comparing their own projects with the pattern samples, they were able to figure out the commonalities and adopt the solution to their complex problems.

10.4.2.2 Unawareness: Use the Patterns without Noticing Them

Subject MK became aware of our design patterns by searching Google for GC files when he started to learn GC a couple of months ago. During the interviews, he stated several times that he would not use patterns:

> "No, I do not think I am going to use them (design patterns. See? I am programming to get what I want. It would be more direct and controllable than your approaches. I looked at your pattern examples to check how you wrote those scripts."

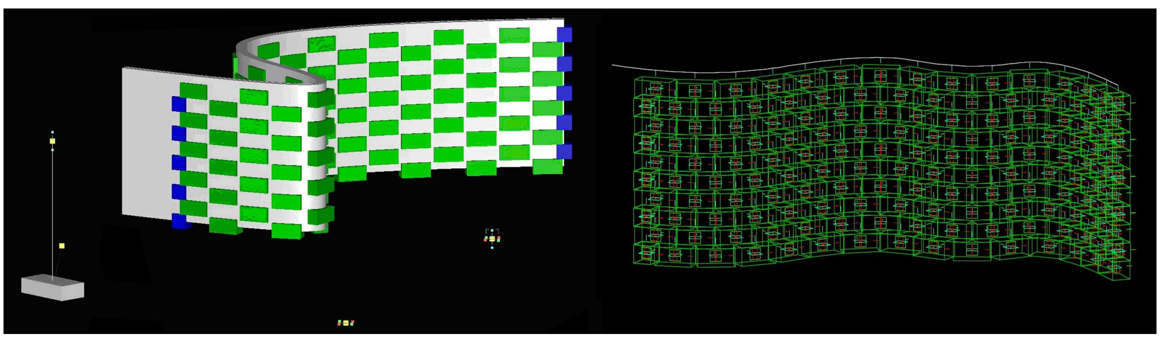

Figure 10.11: MK in SmartGeometry 2008: Possible Controller in a Curved Brick Wall.

However, from coding his data, we found that there were at least two patterns used in the implementation. For example, as illustrated in Figure 10.11, he developed a separate controlling component beside the main model to adjust the shape and height of curved brick wall and size of bricks. Such a strategy was introduced by the very first pattern *Controller* in our repository. Isolating manipulations to a simple place away from the complex detail of a model means that the designer can change the model more easily. Using a logic for control that is different from the way the model is defined means that the designer can use the most appropriate interaction metaphor. Maybe MK didn't use the sample scripts directly, but the logic he used happened to be exactly the same.

The fundamental principle of pattern *Transformer* is to keep the shape of the rigid body no matter the orientation and location is. Defining each object's initial coordinate system and calculating the relations among these coordinate systems are the essential steps. MK calculated the coordinate system for each brick carefully so that the size would not be stretched when the curve changes (right image in Figure 10.11). He also used the same logic as the *Transformer* pattern without awareness.

Pattern is a mental construct to transfer strategy ideas. It is great that some of its sample files can be used by designers directly. But the essential goal is to demonstrate the strategy and help users to scale up. In MK's case, I believe that he understood and adopted the same strategy in this problem solving process without noticing that he did so.

10.4.2.3 Serendipity

"I find that a great part of the information I have was acquired by looking up something and finding something else on the way[190]." This definition of *serendipity* accurately describes our discovery of multimodal pattern of use during collaboration. Most participants started to read pattern descriptions in the pre-training workshop and tried our samples following the step-by-step demonstration. We assumed that going through the tutorials would be the main way for them to learn and use. During the interviews, we realized that the conditions were much richer.

Direct Poaching: MH and SR from UN studio were not our subjects. They were widely acknowledged as having done a great job in their collaborative project and in demonstrating multiple interpretations of their design ideas. MH told us that he scanned through the samples in our site, downloaded one sample file (left upper image in Figure 10.12) in the *Transformer* pattern and used it directly as an inclusion in their hexagon structure (left bottom images) to make it slide to open.

When we looked at the representations of this project (Figure 10.12), we did not notice this direct adoption at all. Maybe Michel de Certeau's concept *poaching* [67] is a proper description of this form of use.

Inquiry for Further Interpretation: In some cases, the text explanation and sample demonstrations were still insufficient. For example, in the second workshop, senior Tutor SB came to talk to us:

> "Hi, we met this problem in my group...I looked through your site and thought *Recursion* would be the solution. But we don't quite understand

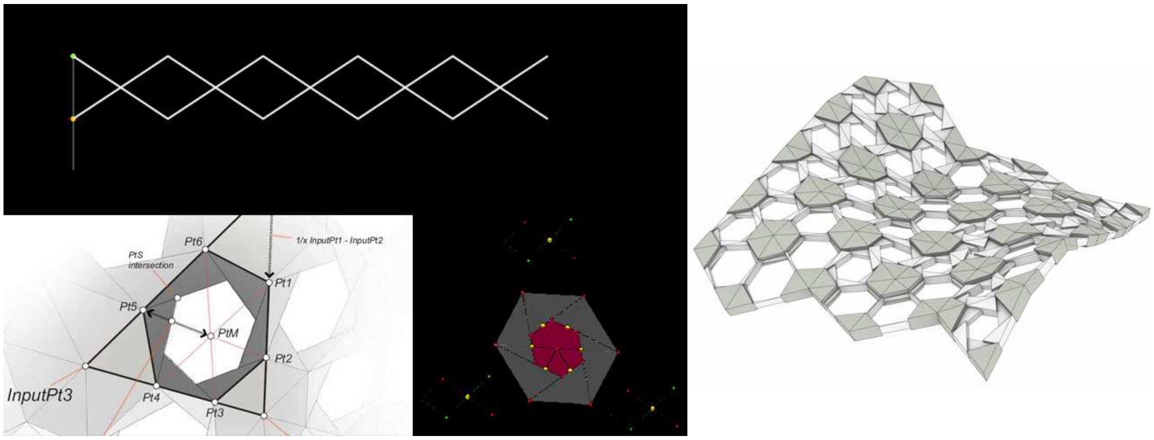

Figure 10.12: MH and SR in SmartGeometry 2008: UN Studio Breathing Surface Project.

> the function. Can you go through the scripts with us? ... Hmm, it seems that that is not what we want. Anything other examples in your site you can recommend?”

Extension Leads to Scaling: Apart from suggesting new patterns ideas, participants and tutors were also interested in improving our solutions and recommending new sample files in existing patterns. This kind of discussion went on beyond the limit of the workshop. For example, here is one email sent by subject DS one month after the SmartGeometry workshop:

> “I am just writing to say I have found a way to find the maximum and minimum x and y values of a line. It can improve your Goal Seeker pattern . . . The whole idea is that you change the graphs to get different solutions for the extrusion and the max and min extents will update. If you know of an easier way to do this I would appreciate your input.”

Subject TH also packaged some GC files he created before and sent them to me by email with his questionnaire answers after the ACADIA workshop. He believed that he had used some pattern-related strategies in those works and wanted us to filter out some samples to enrich the repository.

In different scenarios, our patterns, as small capsules of strategies with samples of small problems, were accepted by designers to solve their complicated projects. Even in some cases, the designers were not aware of such acceptance. We still feel satisfied that the patterns help designers to understand and communicate design problem in depth.

10.4.3 Patterns Support Collaborative Design

The design and construction industries are in the midst of a fundamental transformation towards so-called "digital practice" [24]. This is driven by a confluence of factors: parametric modeling, digital fabrication, high bandwidth communication, and the globalization of practice. A feature of this transformation is that it is being aided by an unprecedented collaboration across firms and schools. There is a crucial need to share information, learning, and techniques at an explicit level. Parametric CAD systems support this process of collaborative design in several ways:

Share the process among collaborators: In traditional CAD systems, the modeling outcome or 2D drawings are the only artifacts for communication. The information of the process - how they were designed - has been lost. It is easy to copy the items, but very hard to duplicate the process of making something different but with the same logic. Parametric design systems record the creation process, retaining the design logic so that collaborators can interpret the originator's work. As a result, other designers can reinforce the sense of design in the group.

Communicate with other professionals: Outside the design group, architects need to collaborate with clients, engineers, fabricators, and contractors. In many cases, architects have to adjust the design slightly here and there to meet others' needs. Editing parameters is much more direct and secure than editing the values of an existing artifact.

Facilitate interdisciplinary design: Experts in fields such as computer science, mathematics and engineering are usually peripheral to the architecture domain but increasingly need to be part of the design conversation [24]. Parametric modeling makes use of explicit computational and mathematical expressions, making it easier for these experts to contribute their knowledge and help architects to understand and implement complex geometry in their design projects.

While recruiting the subjects, we did not purposely search for designers who worked in teams. However, the phenomena described in section *Contextual Aspects of Design Patterns* are also useful evidence to show how patterns and the pattern repository were used by designers during collaboration and communication.

The extension of collaboration also took other forms. For example:

Collaborate with Design Firms: Participants TS and DR were working on a system that limit amount (27) of hexagon shapes can spawn infinite kinds of net layout structure (Figure 10.13). During the workshop, they had got the spawning principles done, but they needed somebody who can script in GC to really solve their design problem. A collaborative design relationship has evolved between their team and ours.

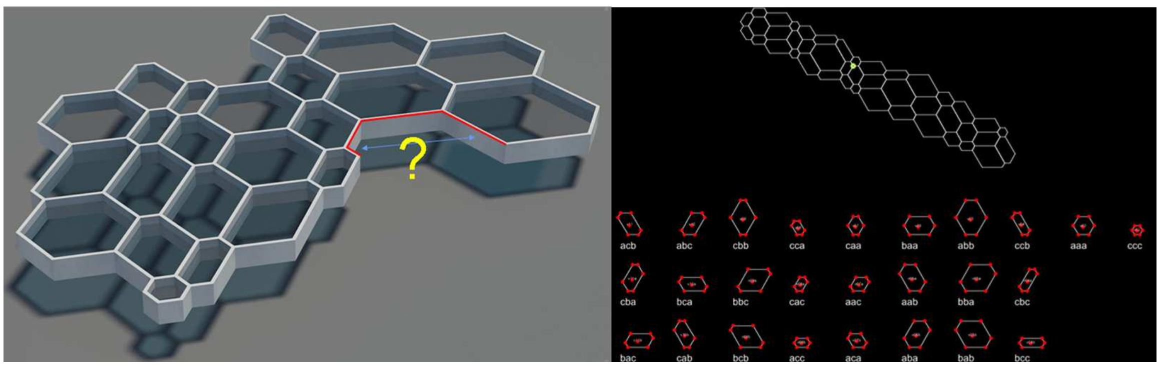

Figure 10.13: TS and DR in SmartGeometry 2008: Hexagon Spawning Project.

Collaboration with University Institutor After the ACADIA workshop, university educator PV used our repository directly to teach a undergraduate course (Arch534, course project is a parametric bridge design) in 2009 Spring semester. In his questionnaire answers, he sent the link of his students' design outcomes and suggested that we adopt some of them as sample patterns.

Parametric modeling is transforming professional practice. But it is evident that this transformative technology imposes new needs for a higher level of support for its communities of practice. I conjecture that design patterns are a good device around which to structure such support.

10.5 Evaluative Discussion: Tetrad of Media Effects of Parametric CAD Systems

In Chapter Three, I remapped the tetrad diagram of McLunhan's laws of media effects (Figure 3.1) into a two-dimensional coordinate system, with each law occupying one quarter of the space (Figure 3.2). I also reviewed the projects of physically based modeling and spreadsheet modeling to analyze their tetrad media effects with my coordinate system (Figure 3.3, Figure 3.4). In the domain of parametric CAD systems, my research did not try to survey and investigate other new parametric modeling tools. Instead, I focused on understanding the design aspects of Bentley's GenerativeComponents and conducted a series of studies to use patterns to model and support the learning and working process. Through these studies, I gained access to real-life design practice. According to the data, below I discuss parametric CAD system's media effects. I am aware of the problem of "taking a part for the whole" so this discussion is just limited to my current understanding in this domain.

Prospective - Positive What does it enhance?

- **Design Collaboration**: The systems firstly help design team to manage product data, share design information, automate workflows, and improve collaboration between designers. Secondly, the collaboration inside the design group has been extended towards the external professionals. In the previous section, I described how parametric CAD systems help collaborators share the creation process, communicate with other professionals through on-going edits, and facilitate interdisciplinary design between designers and scientists.
- **Design Space Exploration**: In a real design environment, participants want to explore different alternatives and decide on the best choice. *Design space exploration* is an approach that helps designers consider many alternatives at once and recall related work previously done in their organization [250]. It also helps designers make, test, remember and adapt designs to new situations. Traditional CAD systems only show one instance of a design. Until now, paper-based sketching was the usual exploratory method. In a parametric CAD model, it is straightforward to review different variations through adjusting the parameters. A built-in recording tool can record key steps and stages for further analysis. For example, Figure 10.14 shows a project in SmartGeometry workshop. The

designer can generate fourteen alternatives of stadium bowl design and compare their plans, elections and perspective views directly.

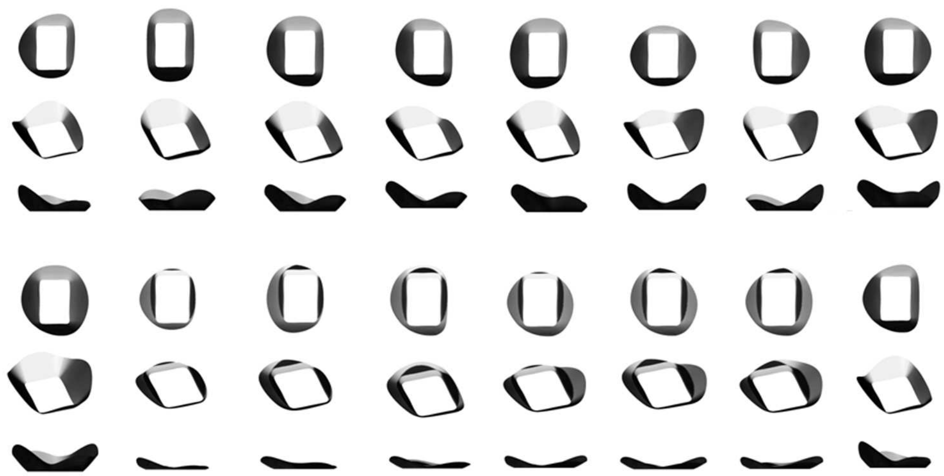

Figure 10.14: Demo of Using GC to Explore Alternatives of Stadium Bowl Design.

- **Design Iteration**: Parametric CAD systems not only support comparing and optimizing alternatives, but also allow designers to easily subject designs to the same conditions that they will experience in the real world [212]. They enhance the quality of design while reducing the cost for live prototypes and testing. For example, Autodesk Ecotect is an building analysis program that allows designers to apply necessary tools for an energy efficiency calculation [76]. Now GC is able to talk to Ecotect via a client-server model to do energy perform calculations [69]. SolidWorks provides a computational fluid dynamics (CFD) analysis application. The designer can simulate liquid and gas flow in real world conditions, run "what if" scenarios, and quickly analyze the effects of fluid flow, heat transfer, and related forces on immersed or surrounding components. With the help of such quick simulations and evaluations, design iterations can be significantly accelerated.

Prospective - Negative What does it flip into when pushed to extremes?

- **GUI tools or Programming**: It is far too often the case that people must perform repetitive tasks through the graphic user interface (GUI) that could be completed much more quickly and correctly with an algorithm. Designers interact with parametric CAD systems at three levels: modeling (using existing functions provided by GUI), scripting (writing short scripts in GUI), and programming (writing scripts outside GUI). Modeling can only build comparable simple structures. To work with complex geometries, designers have to include some scripting in their parametric models. Some complicated projects might need full programming outside of the parametric CAD system interface. To think like a designer and work like a computer scientist is big challenges for many designers.

End-user programming tools promise to support people in expressing and using algorithms from within computing tools such as spreadsheets, word processing tools, image systems and computer-aided design systems. However, current, useful end-user programming systems have been hard to achieve. How to teach, encourage, and support designers to write algorithms remains as a problem for parametric CAD system developers.

- **Uncommunicative Parametric Neighborhoods**: Nowadays, architects emerge to apply parametric design techniques frequently in their creation. Parametric design lets architects and engineers pursue designs and achieve results that were "virtually unthinkable before" [85]. However, the over-emphasis on the individual geometric structure can cause the ignorance of visual communication between the building itself and its neighborhood context. In some cases, neighboring parametrically designed buildings look too independent in the same district. Recently, some researchers started to concern this issue and suggested to bring neighborhood parametric planning into studio design courses to draw the attention in design fields [74].

Retrospective - Negative What does it make obsolete?

- **Unadaptable Library Components**: In conventional architectural CAD software, the systems usually provide 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 predefined by CAD software vendors with limited adaptability - a designer cannot add a structural element if there are no structural elements in the library to add. Nowadays, a lot of building projects are targeted to be sustainable and components are expected to be responsive to their environments. Thus, designers have to take more control over the properties of components instead of using fixed items.

- **Drafters**: Word processing, spreadsheets, and database software changed the role of "secretary" in the late 20th century. According to Strassman, "the value-added secretary of the future will have to have mastery over the electronic medium. Secretaries should focus on electronics-aided management of recorded knowledge as their primary value to the organization [217, p.16-17]". I believe that a similar case will emerge in the computer-aided design domains. Traditionally, it is the drafter's task to prepare, duplicate, modify, and finish technical drawings and plans. Parametric design systems accelerate the design iteration and in a lot of cases can generate the drawings directly from parametric models. In the near future, the drafters will face the reconstruction of their roles in the design and production cycle.

Retrospective - Positive What does it retrieve?

- **Sketch Book**: It is an architects' tradition to carry a sketch book around. Today, young designers tend to carry a laptop around because computers can store much more information. By definition, a *sketch* is a roughed out design. It is quick, inexpensive, plentiful, disposable, with clear vocabulary and appropriate degree of refinement. [42]. It is used to suggest and explore rather than confirm deign ideas. In contrast, a *prototype* or more detailed model aids idea evaluation and confirmation.

  Parametric CAD systems enable designers to draft sketches in a quick and inexpensive way. For example, shown by Figure 7.4, ZD's first stage of horizontally sliding arc panels was actually a sketch model of his second stage of sequenced shingles with control arms. As another example, in Figure 7.11, MD and SP firstly used a sketch model to understand the system: the roof surface is responsive to the four red blocks below. Then, they analyzed and developed the responsive roof surface in details. Design is naturally an iterative process. Parametric CAD systems support such iterations through record the transaction history and allow designer to undo and redo selectively. As a result, they made "digital sketching" possible.

- **Building Geometry**: The origin of geometry lies in the concerns of everyday life. Greeek mathematician Eucllid wrote the most influential geometry textbook "Elements of Geometry" 2300 years ago. Illustrated by Tones Brunes [231], geometry had been broadly and penetrably applied on ancient architecture in ancient Egypt, Pythagoras, early Christian Era and the Middle Ages. Gorgeous examples of ancient temples, palaces, and building sites now act as the main tourist attractions around the world. Industrialization and modern construction technology accelerated the speed of building fabrication and also fundamentally changed the design goal. Modernists more commonly ascribed the changed apprehension of space not to scientific concepts, but rather to technology [37]. Architects' loyalty and interests in geometry started to fade during the Modernism.

  Parametric design systems emphasize on the explicit interpretation of geometric structures during the design process. In order to drive the innovative software and create more impressive works, architects urgently need to rebuild up the solid knowledge of building geometry to catch up with the new trend.

10.6 Summary

This chapter works in concert with Chapter Two, adopting the cognitive frameworks to evaluate and discuss how patterns were used as tools in real world design practice. There are two rounds of discussion: one focuses on the cognitive aspects of the design practice with patterns, and the other emphasizes whether or not patterns are supportive in three perspectives. These two discussions are closely related and also overlapped in some ways. Overall, they show us a thick and honest description of how patterns were used in two parametric architecture design workshops.

Chapter 11

Conclusion

11.1 Review of the Journey

This research has been motivated by an interest in cognitive and human-computer interaction issues, surrounding the support of designers' learning, working and collaboration. It defined a research hypotheses focus on the use of patterns to model designers' learning and working process in parametric modeling applications, and the application of such patterns to support designer action.

The literature review comprises three parts: cognitive frameworks of design activity and design process (Chapter Two), existing parametric design systems and technologies (Chapter Three), and pattern-based research (Chapter Four).

- The review of cognitive frameworks not only provides the foundational knowledge of cognitive theories but also outlines criteria that must be considered when constructing the research design and evaluation studies.
- Parametric modeling software has introduced into practice computational mechanisms and interfaces for representing variation in design. Compared with the CAD domain, such concepts have been used in other areas (such as physically based modeling and spreadsheet modeling) for many years. By using parametric modeling in CAD systems, it is possible to develop models that generate, record, and represent design alternatives, but it is very difficult to understand the range of possibilities entailed by such models.
- The literature on design patterns shows that patterns express design work at a tactical

> level, above simple editing and below overall conception. A pattern is a particular mental construct. We use it to explain regularity in human practice, and employ it to make better learning material to support that practice.

After scoping the research methodology, I conducted a series of studies to search, gather, elicit, author, communicate, and evaluate patterns. We observe in designers' practice with parametric CAD systems that patterns are not explicit in the practice, but pattern-like phenomena and strategies are seen in many instances. We also see many conversations about such things. To analyze the data, I searched for idioms of use that suggest pattern-like thinking. Pattern ideas emerge as packages of examples and discussions around similar applications in different problems. Our efforts at discerning and authoring patterns aim to provide designers a platform that they can scan through and find the most related strategies with abundant example scripts to reference, copy and modify. The design of both pattern structure and pattern repository is to serve such goals.

Data analysis shows that some components in patterns convey contextual information to users while some did not fully satisfy our expectation. The language-based and graphic-based representations serve different aspects of communication. Around the use of patterns in projects, we observe both goal-directed and work-around activities. In general, design patterns can support learning, expertise in scaling up, and collaborative design processes. However, users were not fully satisfied with such limited support. Suggestions on the new content and structure are rich in the collected data. Currently, DesignPatterns.ca is neither a sophisticated digital repository nor a complete content authoring tool. It combines both basic functions and aims to provide simple and sufficient support to its users. We chose the DSDM agile method and communicated with our users regularly and frequently during the entire development. DesignPatterns.ca supports the current authoring and communication around parametric design patterns. Users expect us to add more scaffolding and communicative functions.

11.2 Knowns and Unknowns of this Research

From this PhD research:

We *do know* that the current group of thirteen parametric design patterns can augment designer' practice when they are using Bentley's GenerativeComponents™. There are two main types of practice in parametric CAD systems: one that constructs the models from low-level primitives and the other that involves connecting pre-defined model components with parameters and algorithms. GC adopts the former while others such as Maxon Cinema 4D [139] and Rhino Grasshopper [65] belong to the latter type. We *do not know* if the concepts of current GC patterns can directly transfer to the other type of parametric CAD applications. If not, adjustment at some level should be made to make these patterns benefit users of other applications.

We *do know* that it requires a long time frame to see complete pattern effects. Reflective thinking helps users to access and employ stored knowledge, make inferences about new information, and determine implications [165]; it is how people actually learn. The short time frame available in workshop studies only informs us of early signals of use. We *do not know* how parametric design patterns engage in prolonged design practice. If we want to investigate patterns' effects in such a direction, what should the proper methods and criteria to inspect and measure that be?

We *do know* that during data analysis of current short time frame studies, *activity theory* [118] helped to search for persistent structure across design activities, and *distributed cognition* [104] focused on the common transformation and coordination between individuals and artifacts. According to current data, we did see outlier cases that some users interpret and adopt patterns with unexpected approaches. We *do not know* whether or not more outlier cases will take place in prolonged design studies. If yes, is *situated action models* [218, 219] a mature and suitable cognitive framework to investigate emergent and contingent facts of how designers use patterns?

We *do know* that there is a loop in which parametric design strategies cycle from design practice to theory and then from theory to design practice. This research focuses on augmenting, contributing, observing, enhancing, and evaluating such a loop with the metaphor of design patterns. We *do not know* if this loop can run independently. There is one piece in the loop we have not been able to investigate further - how designers' contributions can feed back to the patterns, and how a pattern and its samples can be polished and enriched by designers' active involvement and communication.

We *do know* that most designers using parametric modeling are amateur programmers. In the studies, they accepted patterns because their preferred style of work is to find, copy and modify. That might also be the reason why the samples are the most welcomed component in pattern structure. We *do not know* what the next step to enhance the "copy and modify" end user programming style should be. Perhaps we should provide a bigger resource of useful scripts. Maybe the existing samples should be integrated in the system design and make "recommendations" that are not interruptive to designers' thought flow. Or maybe "copy and modify" can form a new programming style supported by parametric design systems.

At the current stage of research, I continue to receive feedback, questions, and advice from my designer friends. A fact of this research is that I learned much more from my research participants than I could contribute to them. This work is dedicated to them.

Appendix A

Appendices

A.1 An Email Questionnaire to Check on Subsequent Progress

Dear participant AA:

How are you! It has been two weeks after our ACADIA GenerativeComponents workshop. I guess it might be a proper time to check your recent progress with the software and continue our discussion on your current parametric design problems. I prepare a questionnaire to gather your feedback. However, please do not be restricted by the list. Only select and answer the questions you have something to say, or tell me your current experience and impression with GC and patterns. If possible, some related images and model scripts could be attached.

1. During the past week, did you use GenerativeComponents? If yes, how many hours have you been working on it?

2. During the past week, did you revisit the DesignPatterns repository? If yes, did you focus on several specific patterns or scan through all of them?

3. Can you describe the project you are currently working on, especially the problems that you need GenerativeComponents to solve? If possible, related sketches could be attached.

4. Do you think you have used some patterns in the current project? If yes, please list them and explain the context.

5. In the previous projects, do you think any of the patterns can improve your work? If yes, could you please list these patterns?
6. What are the current unsolvable problems? Do you think any pattern in the repository is potential related but not directly helpful?
7. To improve existing patterns, do you have any new samples to suggest? If possible, could you please contribute some model scripts?
8. Do you have any new pattern idea to suggest? If possible, can you illustrate the pattern idea through words, paper-sketch or model scripts?
9. From your observation, what are the problems and mistakes (if there is any) in the existing pattern repository? Is there any suggestion you want to make to improve the content development?
10. Have you demonstrated GenerativeComponents to your colleagues? Have you introduced the DesignPatterns repository to your colleagues as well?
11. If you need to teach somebody GenerativeComponents, will you use pattern repository as part of your instruction materials?
12. Have you ever used other parametric modeling systems? Do you think any of the patterns are transferable in these systems? If yes, what is the system and which pattern(s)?

Again, thank you very much for participating in the workshop study and this follow-up questionnaire interview. Please be assured that any and all data will be handled confidentially.

Bibliography

[1] R. Abraham and M. Erwig. Ucheck: A spreadsheet type checker for end users. *Journal of Visual Languages and Computing*, 18(1):71–95, 2007.

[2] J. Adams, J. Flatto, and L. Gardner. Combining hands-on, spreadsheet and discrete event simulation to teach supply chain management. In M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, editors, *2005 Winter Simulation Conference*, Orlando, FL, 2005. Winter Simulation Conference.

[3] S. Ahmed, K. M. Wallace, and L. T. Blessing. Understanding the differences between how novice and experienced designers approach design tasks. *Research in Engineering Design*, 14:1–11, 2003.

[4] R. Aish. Bentley's generativecomponents: a design tool for exploratory architecture. 2003.

[5] R. Aish and R. Woodbury. Multi-level interaction in parametric design. In *Smart Graphics*, pages 151–162, Frauenwoerth Cloister, Germany, 2005. Lecture Notes in Computer Science.

[6] O. Akin. *Psychology of Architecture Design.* Pion Ltd., Cambridge, 1989.

[7] L. B. Alberti. *Ten Books on Architecture.* (1404-1472) Translated into Italian by Cosimo Bartoli and into English James Leoni, edited by Joseph Rykwert. Transatlantic Arts., New York, 1966.

[8] C. Alexander. *Notes on the Synthesis of Form.* Harvard University Press, Cambridge, 1964.

[9] C. Alexander. *The Oregon Experiment.* Oxford University Press, Berkeley, Calif., 1975.

[10] C. Alexander. *The Timeless Way of Building.* Oxford University Press, New York, 1979.

[11] C. Alexander, H. Davis, J. Marinez, and D. Corner. *The Production of Houses.* Oxford University Press, New York, 1985.

[12] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language Which Generates Multi-service Centers*. Center for Environmental Structure, Berkeley, Calif., 1968.

[13] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, New York, 1977.

[14] C. Alexopoulos and A. F. Seila. Output data analysis. In J. Banks, editor, *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. John Wiley, New York, 1998.

[15] R. Anderl and R. Mendgen. Parametric design and its impact on solid modeling applications. In *the Third ACM Symposium on Solid Modeling and Applications*, pages 1–12, Salt Lake City, Utah, United States, 1995. ACM.

[16] J. Anderson. *The Architecture of Cognition*. Harvard University Press, Cambridge, MA, 1983.

[17] B. Archer. *The Need for Design Education*. Royal College of Art, 1973.

[18] ATLAS.ti. Atlas.ti - the qda software. 2007(Dec. 18th), 2007.

[19] E. R. Babbie. *Survey Research MetSurvey Research Methodshods*. Wadsworth, Belmont, CA, 2nd. edition, 1990.

[20] N. I. Badler, K. Manoochehri, and G. Walter. Articulated figure positioning by multiple constraints. *CG&A*, 7(6):28–38, 1987.

[21] N. I. Badler, C. B. Pillips, and B. L. Webber. *Simulating Humans: Computer Graphics Animation and Control*. Oxford University Press, New York, 1993.

[22] L. Bartram. *Enhancing Information Visualization with Motion*. PhD thesis, School of Computing Science, Simon Fraser University, 2001.

[23] R. Barzel. *Physically-based Modeling for Computer Graphics: a Structured Approach*. Academic Press., Boston, USA, 1992.

[24] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development. http://www.agilemanifesto.org(Jan 24th), 2001.

[25] J. Bergin, J. Eckstein, M. L. Manns, and E. Wallingford. Patterns for gaining different perspectives: A part of the pedagogical patterns project pattern language. (17), 2001.

[26] T. Berners-Lee. What the semantic web can represent. http://www.w3.org/DesignIssues/RDFnot.html(Dec. 18th), 1998.

[27] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-centered Systems*. Morgan Kaufmann Publishers, San Francisco, Calif., 1998.

[28] B. G. Bezirtzis, M. Lawis, and C. Christeson. Interactive evolution for industrial design. In *the 6th ACM SIGCHI conference on Creativity and Cognition*, pages 183–192, Washington DC. USA, 2007. ACM.

[29] B. Boehm, A. Abts, S. Brown, B. Chulani, E. Clark, R. Horowitz, D. Madachy, and B. Steece. *Software Cost Estimation with COCOMO II*. Prentice Hall PTR, Upper Saddle River, 2000.

[30] J. Borchers. *A pattern approach to interaction design*. Books24x7.com, Norwood, Mass, 2001.

[31] A. Borning. The programming language aspects of thinglab, a constraint-oriented simulation laboratory. *ACM Transactions on Programming Languages and Systems*, 3(4):353–387, 1981.

[32] M. Botsch and L. Kobbelt. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634, 2004.

[33] R. E. Boyatzis. *Transforming Qualitative Information: Thematic Analysis and Code Development*. Sage Publications, Thousand Oaks, CA, 1998.

[34] R. Bridson, R. Feskiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *International Conference on Computer Graphics and Interactive Techniques 2005*, pages 1–10, Los Angeles, California, 2005. ACM.

[35] W. Broll, I. Lindt, J. Ohlenburg, M. Wittkamper, C. Yuan, T. Novotny, A. F. g. Schiecky, C. Mottramy, and A. Strothmann. Arthur: A collaborative augmented environment for architectural design and urban planning. *Journal of Virtual Reality and Broadcasting*, 1(1):1–10, 2004.

[36] R. Brooks. Comparative task analysis: an alternative direction for human-computer interaction science. pages 50–59, 1991.

[37] V. Burgin. *Geometry and Abjection*. Routledge, London and New York, 1990.

[38] M. M. Burnett, M. J. Baker, C. Bohus, P. Carison, S. Yang, and P. van Zee. Scaling up visual programming languages. *IEEE Computer*, 28(3):45–54, 1995.

[39] M. M. Burnett and H. J. Gottfried. Graphical definitions: Expanding spreadsheet languages through direct manipulation and gestures. *ACM Transactions on Computer-Human Interaction*, 5(1):1–33, 1998.

[40] M. M. Burnett, G. Lixin, L. Dupis, and A. Sheretov. A methodology for testing spreadsheets. *ACM Transactions on Software Engineering and Methodology,*, 10(1):110–147, 2001.

[41] R. Butler. Is this spreadsheet a tax evader? how h. m. customs and excise test spreadsheet applications. In *33rd Hawaii International Conference on System Sciences*, pages 1–6, Hawaii, USA, 2000. IEEE Computer Society.

[42] W. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann, San Francisco, CA, 2007.

[43] CADWORK. Cadwork: 3d cad/cam for timber/log construction. 2009(May 8th.), 2009.

[44] P. E. Carlson. Updating and broadening the use of single subject designs in reading. *Reading Psychology*, 6(3-4):251–265, 1985.

[45] CATIA. Catia - virtual design for product excellence. http://www.3ds.com/products/catia/welcome/(June 26th.), 2009.

[46] J. Chai and J. K. Hodgins. Constraint-based motion optimization using a statistical dynamic model. *ACM Transactions on Graphics*, 26(3):Article No.8, 2007.

[47] C. Chen and E. C. Prakash. Physically based facial expression synthesizer with performance analysis and gpu-aided simulation. In *2006 International Conference on Game Research and Development*, pages 171–176, Perth, Australia, 2006. Murdoch University.

[48] W.-K. Chen, editor. *The VLSI Handbook*. CRC Press., Boca Raton, FL, 2000.

[49] Y. V. Chen, Z. C. Qian, and R. F. Woodbury. Developing a simple repository to support authoring learning objects. In *Intelligent Interactive Learning Object Repositories (I2LOR) Conference*, Montreal, Canada, 2007.

[50] M. T. H. Chi, P. Feltovich, and R. Glaser. Categorization and representation of physics problems by experts and novices. *Cognitive Science*, 5:121–152, 1981.

[51] L. Chittaro and D. Corvaglia. 3d virtual clothing: from garment design to web3d visualization and simulation. In *The Eighth International Conference on 3D Web Technology*, pages 73–85, Saint Malo, France, 2003. ACM.

[52] B. Choe, M. G. Choi, and H.-S. Ko. Simulating complex hair with robust collison handling. In *SIGGRAPH 2005 / Eurographics Symposium on Computer Animation*, pages 153–160, Los Angeles, California, 2005. ACM.

[53] C.-H. Chu, M.-C. Song, and V. C. Luo. Computer aided parametric design for 3d tire mold production. *Computers in Industry*, 57(1):11–25, 2006.

[54] A. Clark. *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing*. MIT Press, Cambidge, Mass, 1991.

[55] M. Clermont. Heuristics for the automatic identification of irregularities in spreadsheets. *First Workshop on End User Software Engineering (WEUSE I)*, 30(4):1–6, 2005.

[56] S. Connelly, J. Burmeister, A. MacDonald, and A. Hussey. Extending and evaluating a pattern language for safety-critical user interfaces. In *Sixth Australian Workshop on Safety Critical Systems and Software*, pages 39–49, Brisbane, Australia, 2001. Australian Computer Society, Inc.

[57] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill Book Company, Cambridge, Mass, 2001.

[58] J. W. Creswell. *Quasi-experimentation: Design and Analysis Issues for Field Settings*. Rand McNally College Pub. Co.,, Chicago, 1979.

[59] J. W. Creswell. *Qualitative Inquiry and Research Design: Choosing among Five Traditions*. Sage Publications,, Thousand Oaks, Calif., 1998.

[60] J. W. Creswell. *Research Design: Qualitative, Quantitative, and Mixed Method Approaches*. Sage Publications,, Thousand Oaks, Calif., 2nd edition, 2003.

[61] L. J. Cronbach and P. Suppes. *Research for Tomorrow's Schools: Disciplined Inquiry for Education*. Macmillan, New York, 1969.

[62] N. Cross. Expertise in design: an overview. *Design Studies*, 25(5):427–441, 2004.

[63] M. Crotty. *The Foundations of Social Research: Meaning and Perspective in the Research Process*. Sage, London ; Thousand Oaks, Calif., 1998.

[64] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Conference on Human Factors in Computing Systems*, pages 175–182, Vienna, Austria, 2004. ACM.

[65] S. Davidson. Grasshopper: Generative modeling for rhino. http://www.grasshopper3d.com/(June 26th.), 2009.

[66] E. A. Davis. Scaffolding students' knowledge integration: Prompts for reflection in kie. *International Journal of Science Education*, 22(8):819–837, 2000.

[67] M. De Certeau. *The Practice of Everyday Life*. University of California Press., Berkeley, 1984.

[68] A. Dearden, J. Finlay, L. Allgar, and B. McManus. Evaluating pattern languages in participatory design. In *CHI' 02 Extended Abstracts on Human Factors in Computing Systems*, pages 664–665, Minneapolis, Minnesota, USA, 2002. ACM.

[69] K. DeBiswas. Gc ecotect link. http://www.ecotect.com/(July 26th.), 2009.

[70] S. L. Delp and J. P. Loan. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine*, 25(1):21–34, 1995.

[71] J. Deng, E. Kemp, and E. G. Todd. Focussing on a standard pattern form: the development and evaluation of muip. In *7th ACM SIGCHI New Zealand Chapter's International Conference on Computer-human Interaction*, pages 83–90, Christchurch, New Zealand, 2006. ACM.

[72] K. M. DeWalt and B. R. DeWalt. *Participant Observation: a Guide for Fieldworkers.* AltaMira Press, Walnut Creek, CA, 2002.

[73] C. DiGiano, L. Yarnall, C. Patton, J. Roschelle, D. Tatar, and M. Manley. Collaboration design patterns: Conceptual tools for planning for the wireless classroom. pages 39–47, the IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE02), 2002. IEEE Computer Society.

[74] A. Dong, A. Vande Moere, and J. S. Gero. Parametric designing in architecture. In *Computer-Aided Architectural Design Futures (CAADFutures)*, pages 237–250, Sydney, Australia, 2007.

[75] N. Ducheneau and R. Moore. Let me get my alt: Digital identiti(es) in multiplayer games. In *CSCW 2004 Workshop on Representation of Digital Identities*, Chicago IL., 2004. ACM.

[76] Ecotect. Ecotect homepage. www.mit.edu/kkdb/newhome/generative/index.html(July 26th.), 2009.

[77] L. Ede. *Work in Progress: a Guide to Academic Writing and Revising.* St. Martin's Press, New York, 1998.

[78] K. A. Ericsson. Expertise. 1999.

[79] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *SIGGRAPH 2001*, pages 251–260, Los Angeles, CA., 2001. ACM SIGGRAPH.

[80] G. Fischer. Beyond "couch potatoes": from consumers to designers and active contributors. 7(12), Oct.15 2007 2002.

[81] G. Fishman. *Monte Carlo Concepts, Algorithms and Applications.* Springer, New York, 1996.

[82] J. D. Foley, A. Van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, Mass., 2nd edition, 1990.

[83] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software.* Addison-Wesley, Reading, Mass., 1995.

[84] C. Geertz. Thick description: Toward an interpretive theory of culture. In *The Interpretation of Cultures*, pages 3–30. Basic Books, New York, USA, 1973.

[85] GenerativeComponents. Microstation: A source of innovation, a cad platform for integration. 2007(December 8th.), 2007.

[86] J. S. Gero. Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4):26–36, 1990.

[87] S. Giertz, B. Diekkruger, and G. Steup. Physically based modelling of hydrological processes in a tropical headwater catchment in benin - process representation and multi-criteria validation. *Hydrology and Earth System Sciences*, (2):595–651, 2006.

[88] R. Giordano and B. David. Participant stakeholder evaluation as a design process. In *ACM Conference on Universal Usability (CUU 2000)*, pages 53–60, Arlington, VA, USA, 2000.

[89] I. Graham. *A Pattern Language for Web Usability*. Addison-Wesley, London; Boston, 2003.

[90] A. Granlund, D. Lafreniere, and D. Carr. A pattern-supported approach to the user interface design process. In *HCI International 2001 9th International Conference on Human-Computer Interaction,*, pages 1–5, New Orleans, USA, 2001.

[91] T. A. Grossman. Spreadsheet engineering: A research framework. In *European Spreadsheet Risks Interest Group Symposium*, Cardiff, Wales, 2002.

[92] T. A. Grossman and O. Ozluk. A paradigm for spreadsheet engineering methodologies. In *European Spreadsheet Risks Interest Group Symposium 2004*, pages 23–24, Klagenfurt, Austria, 2004.

[93] E. G. Guba. Criteria for assessing the trustworthiness of naturalistic inquiries. *Educational Communication and Technology Journal*, 29:75–92, 1981.

[94] E. G. Guba and Y. S. Lincoln. *Fourth Generation Evaluation*. Sage Publications, Newbury Park, Calif., 1989.

[95] E. G. Guba and Y. S. Lincoln. Competing paradugns in qualitative research. In N. K. Denzin and Y. S. Lincoln, editors, *Handbook of Qualitative Research*, page 643. Sage Publications, Thousand Oaks, 1994.

[96] L. A. Guion. Triangulation: Establishing the validity of qualitative studies. Technical report, University of Florida, 2002.

[97] S. Hadap, M.-P. Cani, M. Lin, T.-Y. Kim, F. Bertails, S. Marschner, K. Ward, and Z. Kacic-Alesic. Strands and hair: Modeling, animation, and rendering. 2007.

[98] S. Hadap and N. Magnenat-Thakmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3):329–338, 2001.

[99] J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 2(16):30–37, 2001.

[100] J. K. Hodgins, W. L. Wooten, D. C. Brogan, and J. F. O'Brien. Animating human athletics. In *the 22nd International Conference on Computer Graphics and Interactive Techniques*, pages 71–78, Los Angeles, 1995. ACM.

[101] R. Hoss and C. Hoffman. Does dream color reflect emotion? a long term journaling study. In *The International Association for the Study of Dreams PsiberDreaming Conference*, Online, 2004. ADS.

[102] S. Hudson. User interface specification using an enhanced spreadsheet model. *ACM Transactions on Graphics*, 13:209–239, 1994.

[103] C. Hughes and J. Moshell. Action graphics: A spreadsheet-based language for animated simulation. In T. Ichikawa, E. Jungert, and R. Korfhage, editors, *Visual Languages and Applications*, pages 203–235. Plenum Publishing, New York, 1990.

[104] E. Hutchins. *Cognition in the Wild*. MIT Press, Cambridge, Mass., 1995.

[105] E. Hutchins. How a cockpit remembers its speeds. *Cognitive Science*, 19(2):265–288, 1995.

[106] G. Ifrah. *The Universal History of Numbers: from Prehistory to the Invention of the Computer*. John Wiley and Sons, Inc., New York, 2000.

[107] T. Isakowitz, S. Schocken, and H. C. Lucas. Toward a logical/physical theory of spreadsheet modeling. *ACM Transactions on Information Systems*, 13(1):1–37, 1995.

[108] P. Jackson. *Introduction to Expert Systems*. Addison-Wesley, Harlow, England, 3rd ed. edition, 1998.

[109] H. W. Jensen. Digital face cloning. In *SIGGRAPH 2003 Conference on Sketches and Applications*, San Diego, 2003. ACM.

[110] V. John-Steiner. *Creative Collaboration*. Oxford University Press, New York, 2000.

[111] A. Kelley, M. Malin, and G. Nielson. Terrain simulation using a model of stream erosion. In *SIGGRAPH 88*, pages 263–268, Atlanta, Georgia, 1988. Elsevier North-Holland, Inc.

[112] S. a. Kjeld. Ordering systems: Coordinative practices in architectural design and planning. pages 274–283, 2003.

[113] M. Klein, H. Sayama, P. Faratin, and Y. Bar-Yam. The dynamics of collaborative design: Insights from complex systems and negotiation research. *Concurrent Engineering*, 11(3):201–209, 2003.

[114] E. L. Kohler. *A Dictionary for Accountants.* Prentice-Hall, Inc., New York, 1952.

[115] G. N. Koini, S. S. Sarakinos, and I. K. Nikolos. Parametric design of turbomachinery blades. In *the International Conference on Computer as a Tool*, pages 539–542, Sava Center, Belgrade, 2005. IEEE Society.

[116] F. Kolatan. Responsive architecture through parametric design: How parametric systems introduce diversity, adaptability, and responsiveness into the architectural design process. http://www.wolframscience.com/conference/2006/presentations/kolatan.html(June 26th. 2009), 2006.

[117] A. Kumiega and B. Van Vliet. Spreadsheet good practice: Is there any such thing? In *EUSPRIG 2004*, Klagenfurt, Austria, 2004. July 15-16.

[118] K. Kuutti. Activity theory and its application to information systems research and development. In H.-E. Nissen, editor, *Information Systems Research*, pages 529–549. Elsevier Science Publishers, Amsterdam, 1991.

[119] K. Laker and A. Shenker. Exlpoiting the power of oracle using microsoft excel. Technical report, Oracle Corporation, December 2004.

[120] J. Larkin. Display-based problem solving. In D. Klahr and K. Kotovsky, editors, *Complex Information Processing: The Impact of Herbert Simon*, pages 319–341. Lawrence Erlbaum Associates, Hillsdale, N.J., 1989.

[121] J. H. Larkin and H. A. Simon. Why a diagram is (sometimes) worth 10,000 words. *Cognitive Science*, 11:65–99, 1987.

[122] J. Lave. *Cognition in Practice: Mind, Mathematics and Culture in Everyday Life.* Cambridge University Press, Cambridge, 1988.

[123] M. LeCompt and J. P. Goetz. Problems of reliability and validity in ethnographic research. *Review of Educational Research*, 52:31–60, 1982.

[124] T. Leinonen, O. Virtanen, K. Hakkarainen, and G. Kligyte. Collaborative discovering of key ideas in knowledge building. In *Computer Support for Collaborative Learning Conference (CSCL 2002)*, Boulder, Colorado, 2002.

[125] A. Leont'ev. The problem of activity in psychology. *Soviet Psychology*, 13(2):4–33, 1974.

[126] M. Lepper, M. Woolverton, D. Mumme, and J. Gurtner. Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. Lajoie and S. Derry, editors, *Computers as Cognitive Tools*, pages 75–105. Lawrence Erlaum Associates Inc., Hillsdale, N.J., 1993.

[127] M. Levoy. Spreadsheet for images. *Computer Graphics*, 28, 1994.

[128] Y. S. Lincoln and E. G. Guba. *Naturalistic Inquiry*. Sage Publications, Beverly Hills, California, 1985.

[129] Y. S. Lincoln and E. G. Guba. But is it rigorous? trustworthiness and authenticity in naturalistic evaluation. In D. Williams, editor, *Naturalistic Evaluation*, page 416. Jossey-Bass, San Francisco, 1986.

[130] A. Lindblad and G. Turkiyyah. A physically-based framework for real-time haptic cutting and interaction with 3d continuum models. In *the 2007 ACM symposium on Solid and Physical Modeling SPM '07*, pages 421 – 429, Beijing, China, 2007. ACM.

[131] B. Lindelof, B. Forslind, M. Hedblad, and U. Kaveus. Human hair form. *Archives of Dermatology*, 124(9):1359–1363, 1988.

[132] C. K. Liu, A. Hertzmann, and Z. Popovic. Composition of complex optimal multi-character motions. In *2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 215–222, Vienna, Austria, 2006. Eurographics Association.

[133] S. Mader. *Wikipatterns: a Practical Guide to Improving Productivity and Collaboration in Your Organization*. Wiley Pub., Indianapolis, IN, 2008.

[134] N. Maiden, M. Cisse, H. Perez, and D. Manuel. *CREWS Validation Frames: Patterns for Validating Systems Requirements*. Centre for Human Computer Interface Design, London, 1998.

[135] C. Marshall. *Designing Qualitative Research*. Sage Publications, Thousand Oaks, CA, 4th edition edition, 2006.

[136] D. Martin and I. Sommerville. Patterns of cooperative interaction: Linking ethnomethodology and design. *ACM Transactions on Computer-Human Interaction*, 11(1):59–89, 2004.

[137] R. Mattessich. Budgeting models and system simulation. *The Accounting Review*, 36(3):384–397, 1961.

[138] R. Mattessich. Accounting and analytical methods. *Journal of the American Statistical Association*, 61(313):292–293, 1966.

[139] Maxcon. Maxcon cinema 4d homepage. http://www.maxon.net/products/cinema-4d.html(June 26th.), 2009.

[140] R. E. Mayer. *Multimedia Learning*. Cambridge University Press, New York, 2001.

[141] J. McDaid. Halting, sphexishness, and analysis terminable and interminable. *ACM SIGWEB Newsletter*, 8(1):33–40, 1999.

[142] E. McLean, L. Kappelman, and J. Thompson. Converging end-user and corporate computing. *Communications of the ACM*, 36(12):79–92, 1993.

[143] M. McLuhan and E. McLuhan. *Laws of Media: the New Science*. University of Toronto Press, Toronto, 1988.

[144] Z. Melek and J. Keyser. Driving object deformations from internal physical processes. In *ACM Symposium on Solid and Physical Modeling*, pages 51–59, Beijing, China, 2007. ACM.

[145] D. M. Mertens. *Research Methods in Education and Psychology: Integrating Diversity with Quantitative and Qualitative Approaches*. Sage, Thousand Oaks, CA, 1998.

[146] M. Meyer. How to apply the anthropological technique of participant observation to knowledge acquisition for expert systems. *IEEE Transactions on Systems, Man and Cybernetics*, 22(5):983–991, 1992.

[147] G. Miller and R. Dingwall. *Context and Method in Qualitative Research*. Sage Publications, London, 1997.

[148] W. J. Mitchell, R. S. Liggett, and T. Kvan. *The Art of Computer Graphics Programming: A Structured Introduction for Architects and Designers*. Van Nostrand Reinhold, New York, 1987.

[149] R. Mittermeir and M. Clermont. Finding high-level structures in spreadsheets. In *Proceedings Ninth Working Conference on Reverse Engineering WCRE 2002*, Richmond, Vancouver, 2002. IEEE Computer Society.

[150] G. E. Monahan. *Management Decision Making: Spreadsheet Modeling, Analysis, and Application*. Cambridge University Press., Cambridge, New York, 2000.

[151] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *SIGGRAPH 88*, pages 289–298, Atlanta, USA, 1988. ACM.

[152] J. M. Morse. Strategies for sampling. In J. M. Morse, editor, *Qualitative Nursing Research: A Contemporary Dialogue*. Sage, Newbury Park, CA, 1991.

[153] J. M. Morse, M. Barrett, M. Mayan, K. Olson, and J. Spiers. Verification strategies for establishing reliability and validity in qualitative research. *International Journal of Qualitative Methods*, 1(2):1–19, 2002.

[154] C. E. Moustakas. *Phenomenological Research Methods*. Sage, Thousand Oaks, Califonia, 1994.

[155] M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *SIGGRAPH 2002*, pages 49–54, San Antonio, USA, 2002. ACM.

[156] J. P. Munson and P. Dewan. A flexible object merging framework. In *ACM conference on Computer Supported Cooperative Work 2004*, pages 231 – 242, Chapel Hill, North Carolina, 1994. ACM Press/.

[157] MXURBAN. Bentley mxurban: Road design for cities and suburbs. 2007(December 8th.), 2004.

[158] N. Naikar. Beyond interface design: Further applications of cognitive work analysis. *International Journal of Industrial Ergonomics*, 36(5):423–438, 2006.

[159] B. A. Nardi. *A Small Matter of Programming: Perspectives on End User Computing*. The MIT Press., Cambridge, MA., 1993.

[160] B. A. Nardi. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. The MIT Press., Cambridge, Massachusetts, 1996.

[161] B. A. Nardi and J. R. Miller. Twinkling lights and nested loops: Distributed problem solving and spreadsheet development. *International Journal of Man-Machine Studies*, 34(2):161–184, 1991.

[162] A. Newell and H. Simon. *Human Problem Solving*. Prentice Hall, Englewood Cliffs, N.J., 1972.

[163] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, New York, U.S.A, 1988.

[164] D. A. Norman. Cognitive artefacts. In J. Carroll, editor, *Designing Interaction: Psychology at the Human Computer Interface*. Cambridge University Press., Cambridge, 1991.

[165] D. A. Norman. *Things that Make Us Smart*. Addison-Wesley Pub. Co., Reading, Mass, U.S.A, 1st edition edition, 1993.

[166] D. A. Norman. *The Design for Future Things*. Basic Books, New York, 2007.

[167] P. O'Beirne. Spreadsheet mistakes - news stories. 2007(December 12th), 2007.

[168] J. C. Owen. Algebraic solution for geometry from dimensional constraints. In *the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*, pages 397–407, Austin, Texas, United States, 1991. ACM Press.

[169] D. K. Pai. Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum*, 21(3):347–352, 2002.

[170] D. K. Pai, S. Sueda, and Q. Wei. Fast physically based musculoskeletal simulation. *ACM SIGGRAPH 2005 Sketches*, Article No.25, 2005.

[171] A. Palladio. *The Four Books on Architecture.* (1508-1580) Translated by Robert Tavernor and Richard Schofield. MIT Press., Cambridge, Mass, 1997.

[172] R. Panko. What we know about spreadsheet errors. *Journal of End User Computing: Special issue on Scaling up End User Development*, 10(2):15–21, 1998.

[173] F. I. Parker. Computer generated animation of faces. In J. A. Meads, editor, *SIGGRAPH 1*, pages 451–457, Boston, Mass, USA, 1972. ACM.

[174] S. M. Pistsch. *The Effective Use of Three Dimensional Visualization Modeling in the Routine Development Control of Urban Environments.* PhD thesis, Adelaide University, Adelaide, Australia, 2001.

[175] J. Platt and A. Barr. Constraint methods for flexible models. In *SIGGRAPH 88*, pages 279–288, Atlanta, Georgia, 1988. Elsevier North-Holland, Inc.

[176] J. M. Pullan. *the History of the Abacus.* Hutchinson, London, 1968.

[177] Z. C. Qian. Design patterns: Augmenting user intention in parametric design system. In *Creativity and Cognition 2007 Graduate Symposium*, Washington DC, USA, 2007.

[178] Z. C. Qian, Y. V. Chen, and R. F. Woodbury. Near: Collaborative filtering and visualizing information relations in a multimedia repository. In *International Symposium on Smart Graphics 2006*, pages 236–241, Vancouver, Canada, 2006.

[179] Z. C. Qian, Y. V. Chen, and R. F. Woodbury. Participant observation can discover design patterns in parametric modeling. In *ACADIA 2007: Expanding Bodies*, Halifax, Canada, 2007.

[180] Z. C. Qian, Y. V. Chen, and R. F. Woodbury. Developing a simple repository to support authoring learning objects. *Journal of Advanced Media and Communication*, 2(2):154–173, 2008.

[181] V. Quint and I. Vatton. Structured templates for authoring semantically rich documents. In *International Workshop on Semantically Aware Document Processing and Indexing (SADPI-07)*, pages 41–48, Montpellier, France, 2007. ACM.

[182] C. Quintana, J. Eng, A. Carra, H.-K. Wu, and E. Soloway. Evaluating pattern languages in participatory design. In *SIGCHI conference on Human factors in Computing Systems*, pages 473–480, Pittsburgh, Pennsylvania, USA, 1999. ACM.

[183] C. Quintana, B. J. Reiser, E. A. Davis, J. Krajcik, E. Fretz, R. G. Duncan, E. Kyza, D. Edelson, and E. Soloway. A scaffolding design framework for software to support science inquiry. *the Journal of the Learning Sciences*, 13(3):337–386, 2004.

[184] J. Rasmussen. Merging paradigms: Decision making, managment, and control control. In R. Flin, E. Salas, M. Strub, and L. Marting, editors, *Decision Making under Stress: Emerging Paradigms and Applications*, pages 67–85. Ashgate, Aldershot, England, 1997.

[185] J. Rasmussen, A. M. Pejtersen, and L. Goodstein. *Cognitive Systems Engineering*. Wiley, New York, 1994.

[186] M. E. Raven and A. Flanders. Using contextual inquiry to learn about your audiences. *ACM SIGDOC Asterisk Journal of Computer Documentation*, 20(1):1–13, 1996.

[187] J. Reichwein, G. Rothermel, and M. M. Burnett. Slicing spreadsheets: An integrated methodology for spreadsheet testing and debugging. In *2nd Conference on Domain-specific Languages*, volume 2, pages 25–38, Berkeley, CA, 2000. ACM.

[188] B. J. Reiser. Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *The Journal of the Learning Sciences*, 13(3):273–304, 2004.

[189] Revit. Revit architecture. http://usa.autodesk.com/adsk/servlet/index?id=3781831&siteID=123112(June 26th.), 2009.

[190] R. M. Roberts. *Serendipity: accidental discoveries in science*. Wiley, New York, 1989.

[191] F. Ronteltap, P. Goodyear, and S. Bartoluzzi. A pattern language as an instrument in designing for productive learning conversations. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*, Lugano, Switzerland, 2004.

[192] M. Rubin. *Cosserat Theories: Shells, Rods and Points*. Kluwer Academic Publishers, Dordrecht; Boston, 2000.

[193] J. Ruskin. *The Seven Lamps of Architecture*. First published in 1849. Electric Book Co., London, 2001.

[194] J. R. Ruthruff, M. M. Burnett, and G. Rothermel. An empirical study of fault localization for end-user programmers. In *27th International Conference on Software Engineering*, pages 352 – 361, St. Louis, USA, 2005. ACM.

[195] J. Sajaniemi. Modeling spreadsheet audit: A rigorous approach to automatic visualization. *Journal of Visual Languages and Computing*, 11(1):49–82, 2000.

[196] P. Sanguinetti and S. Abdelmohsen. On the strategic integration of sketching and parametric modeling in conceptual design. In *Conference of Association for Computer-Aided Design in Architecture (ACADIA)*, pages 242–249, Halifax, Canada, 2007. Riverside Architectural Press.

[197] M. Scaife and Y. Rogers. External cognition: How do graphical representations work? *International Journal Human-Computer Studies*, 45:185–312, 1996.

[198] M. Scaife and Y. Rogers. External cognition, innovative technologies, and effective learning. In P. Gardenfors and P. Johansson, editors, *Cognition, Education and Communication Technology*, pages 181–202. Erlbaum Associates Publishers, Mahwah, N.J., 2005.

[199] D. J. Schiano, B. A. Nardi, M. Gumbrecht, and L. Swartz. Blogging by the rest of us. In *Extended Abstracts in Conference on Human Factors in Computing Systems*, pages 1143–1146, Vienna, Austria, 2004. ACM.

[200] K. Schmidt and I. Wagner. Ordering systems: Coordinative practices and artefacts in architectural design and planning. *Computer Supported Cooperative Work*, 13:349–408, 2004.

[201] D. A. Schon. *Educating the Reflective Practitioner: toward a New Design for Teaching and Learning in the Professions.* Jossey-Bass, San Francisco, Calif., 1st ed. edition, 1987.

[202] D. Schuler. Liberating voices! a pattern language for communication revolution. 2007(http://trout.cpsr.org/program/sphere/patterns/), Apr. 2nd, 2007 2002.

[203] D. Schuler. A pattern language for living communication. In *Proceedings of Participatory Design Conference 2002*, Malmo, Sweden, 2002.

[204] J. Seidel. Method and madness in the application of computer technology to qualitative data analysis. In N. Fielding and R. M. Lee, editors, *Using Computers in Qualitative Research*, pages 107–116. Sage Publications, Newbury Park, 1991.

[205] A. F. Seila. Spreadsheet simulation. In *38th Conference on Winter Simulation*, pages 11–18, Monterey, CA, 2006. Winter Simulation Conference.

[206] J. J. Shah and M. Mantyla. *Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications.* Wiley, New York, 1995.

[207] V. Shapiro and D. L. Vossler. What is a parametric family of solids? In *the Third ACM Symposium on Solid Modeling and Applications*, pages 43–54, Salt Lake City, Utah, United States, 1995. ACM.

[208] H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: beyond Human-Computer Interaction.* J. Wiley & Sons, New York, 2002.

[209] E. Sifasis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics*, 24(3):417–425, 2005.

[210] K. Sims. Evolving virtual creatures. In *SIGGRAPH 1994*, pages 43–50, Orlando, Florida, 1994. ACM.

[211] J. O. Smith III. *Physical Audio Signal Processing for Virtual Musical Instruments Audio Effects*. Stanford University, Stanford, California, 2007.

[212] SolidWorks. Solidworks homepage. http://www.solidworks.com/(June 26th.), 2009.

[213] J. P. Spradley. *The Ethnographic Interview*. Holt, Rinehart and Winston, NEw York, 1979.

[214] J. P. Spradley. *Participant observation*. Harcourt Brace Jovanovich College Publishers, Orlando, FL, 1980.

[215] SpreadsheetCalculators. Free spreadsheet calculators to help reach your goals. www.gips.com.au/freecalcs.html(April. 18th), 2009.

[216] J. Stapleton. *Dynamic Systems Development Method: The Method in Practice*. Addison Wesley Longman Ltd, Harlow, Essex, 1997.

[217] P. Strassmann. Knowledge management: Oppotunity for the secretary of the future. *Secretary*, 47(6):14–17.

[218] L. A. Suchman. *Plans and Situated Actions: the Problem of Human-Machine Communication*. Cambridge University Press., Cambridge, New York, 1987.

[219] L. A. Suchman. *Human-Machine Reconfigurations: Plans and Situated Actions*. Cambridge University Press., Cambridge, New York, 2007.

[220] L. A. Suchman and R. H. Trigg. Understanding practice: Video as a medium for reflection and design. In J. Greenbaum and M. Kyng, editors, *Design at Work: Cooperative Design of Computer Systems*. Lawrence Erlbaum, Hillsdale, N.J., 1991.

[221] D. Z. Sui and M. F. Goodchild. A tetradic analysis of gis and society using mcluhan's law of the media. *the Canadian Geographer*, 47, 2003.

[222] G. Sussman and G. Steele. Constraints - a language for expressing almost-hierarchical descriptions. *Artificial Intelligence*, 14(1):1–39, 1980.

[223] I. E. Sutherland. *Sketchpad, A Man-Machine Graphical Communication System*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1963.

[224] L. A. Suzuki, A. K. Ahluwalia, Muninder K.and Arora, and J. S. Mattis. the pond you fish in determines the fish you catch: Exploring strategies for qualitative data collection. *the Counseling Psychologist*, 35(2):295–327, 2007.

[225] I. Tabak. Synergy: A complement to emerging patterns of distributed scaffolding. *the Journal of the Learning Sciences*, 13(3):305–325, 2004.

[226] J. Teran, S. Blemkerm, V. N. T. Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 68–74, San Diego, California, 2003. Eurographics Association.

[227] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics*, 21(4):205–214, 1987.

[228] D. Terzopoulos and H. Qin. Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, 12(2):103–136, 1994.

[229] D. Terzopoulos and K. Waters. Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation*, 1(2):76–80, 1990.

[230] J. Tidwell. *Designing Interfaces.* O'Reilly,, Beijing; Sebastpol, 1st edition edition, 2006.

[231] B. Tons. *The Secrets of Ancient Geometry and Its Use.* Sacred Science Institute, EMR Labs, Cascade, CO, 2006.

[232] K. van den Doel. Physically based models for liquid sounds. *ACM Transactions on Applied Perception*, 2(4):534–546, 2005.

[233] A. Vera and H. Simon. Situated action: a symbolic interpretation. *Cognitive Science*, 17:7–48, 1993.

[234] A. Verroust, F. Schonek, and D. Roller. Rule-oriented method for parameterized computer-aided design. *Computer-Aided Design*, 24(10):531–540, 1992.

[235] K. J. Vicente. Cognitive work analysis: Implications for microworld research on human-machine interaction. *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, 2(4):3432–3436, 1995.

[236] K. J. Vicente. *Cognitive Work Analysis: toward Safe, Productive, and Healthy Computer-based Work.* Lawrence Erlbaum Associates, Mahwah, N.J. ; London, 1999.

[237] C. Wagner, M. A. Schill, and R. Manner. Collision detection and tissue modeling in a vr-simulator for eye surgery. In *Workshop on Virtual Environments 2002*, Barcelona, Spain, 2002. Eurographics Association.

[238] D. Week. *A Hermeneutic Approach to the Practice of Architecture in a Foreign Culture.* PhD thesis, University of Sydney, 2000.

[239] D. Week. *the Culture Driven Workplace: Using Your Company's Knowledge to Design the Office.* RAIA Publication, Australia, 2002.

[240] J. Weil. the synthesis of cloth objects. In *SIGGRAPH 86*, pages 49–54, Dallas, Texas, 1986. Elsevier North-Holland, Inc.

[241] C. D. Wickens, J. Lee, Y. Liu, and S. G. Becker. *an Introduction to Human Factors Engineering*. Pearson Prentice Hall, Upper Saddle River, N.J., 2nd edition, 2004.

[242] J. Wilhelms. Using dynamic analysis for realistic animation of articulated bodies. *CG&A*, 7(6):12–27, 1987.

[243] L. Williams. Performance-driven facial animation. In *SIGGRAPH 1990*, pages 235–242, Dallas, TX, USA, 1990. ACM.

[244] T. Winn and P. Calder. A pattern language for pattern language structure. In *2002 Conference on Pattern Languages of Programs*, pages 45–58, Melbourne, Australia, 2003. Australian Computer Society, Inc.

[245] H. F. Wolcott. On seeking-and-rejecting-validity in qualitative research. In E. Eisner and A. Peshkin, editors, *Qualitative Inquiry in Education: The Continuing Debate*, pages 121–152. Teachers College Press, New York, 1990.

[246] J. Wong and J. Hong. Making mashups with marmite: towards end-user programming for the web. In *SIGCHI conference on Human factors in computing systems*, pages 1435 – 1444, San Jose, California, USA, 2007. ACM.

[247] J. Wood, S. Meek, and S. Jacobson. Quantitations of human shoulder anatomy for prosthetic arm control–i. surface modelling. *Journal of Biomechanics*, 22(3):273–92, 1989.

[248] R. F. Woodbury. Design patterns for parametric modeling. http://www.designpatterns.ca(Dec. 18th), 2007.

[249] R. F. Woodbury. Patterns for parametric design. In *Smart Geometry 2009 Alumni Summit*, number March 31st, San Francisco, Califonia., 2009.

[250] R. F. Woodbury and A. L. Burrow. Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 20(2):63–82, 2006.

[251] W. Wu and P. Heng. An improved scheme of an interactive finite element model for 3d soft-tissue cutting and deformation. *Visual Computer*, 21:707–716, 2005.

[252] XESS. Xess: The advanced x windows spreadsheet system. (April 12th.), 2009.