VISUAL SENSITIVITY ANALYSIS
OF
PARAMETRIC DESIGN MODELS:
IMPROVING AGILITY IN DESIGN

by

Nahal Hassanzadeh Salmasi
B.Sc. Civil Engineering, Tabriz University, 1998

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

In the School
of
Interactive Arts and Technology

© Nahal Hassanzadeh Salmasi, 2010
SIMON FRASER UNIVERSITY
Spring 2010

All rights reserved. However, in accordance with the Copyright Act of Canada, this work may be reproduced, without authorization, under the conditions for Fair Dealing. Therefore, limited reproduction of this work for the purposes of private study, research, criticism, review and news reporting is likely to be in accordance with the law, particularly if cited appropriately.
 APPROVAL

Name: Nahal Hassanzadeh Salmasi
Degree: Master of Science
Title of Thesis: Visual Sensitivity Analysis of Parametric Design Models: Improving Agility in Design

Examinining Committee:
Chair:

Dr. Marek Hatala
Associate Professor, School of Interactive Arts and Technology

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

Dr. Halil Erhan
Co-Supervisor
Assistant Professor, School of Interactive Arts and Technology

Dr. John Dill
External Examiner
Professor Emeritus, School of Interactive Arts and Technology

Date Defended/Approved: April 7, 2010
Declaration of Partial Copyright Licence

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection (currently available to the public at the “Institutional Repository” link of the SFU Library website <www.lib.sfu.ca> at: <http://ir.lib.sfu.ca/handle/1892/112>) and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author’s written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

While licensing SFU to permit the above uses, the author retains copyright in the thesis, project or extended essays, including the right to change the work for subsequent purposes, including editing and publishing the work in whole or in part, and licensing other parties, as the author may desire.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada
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, collaborator or research assistant 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.

Simon Fraser University Library
Simon Fraser University
Burnaby, BC, Canada

Last update: Spring 2010
ABSTRACT

The ability of generative and parametric computer-aided design systems to generate models rapidly enables designers to explore the downstream impacts of changes to key design parameters. However, the increasing complexity of these models challenges the abilities of the human-visual perception system, and creates challenges to their effective utilization for sensitivity analysis. In this prototyping study, we propose a method for visual sensitivity analysis that aims to make the effects of change within a parametric model measurable and apparent for designers, thereby improving the potential of these tools for design analysis and improve agility in design process. The approach aims to improve visually analysing the sensitivity of a design model to planned parametric changes. The method adapts the Model-View-Controller paradigm from software engineering to decouple customizable control and visualization features in the design model, while providing interfaces between them through parametric associations. We present findings from our case studies in addition to the results of a user study demonstrating the applicability and limitations of the proposed method.

Keywords: parametric modelling; sensitivity analysis; visual analytics; design

Subject Terms: parametric modelling; architecture; sensitivity analysis; computer-aided design
Dedicated to my dearest Ali & Rejina
ACKNOWLEDGEMENTS

I would like to record my sincere appreciation and gratitude to my supervisors Dr. Halil Erhan and Dr. Robert Woodbury for their exceptional support and guidance from the very early stage of this research.

Collective and individual acknowledgments are also owed to my colleagues at SIAT whose presence somehow perpetually refreshed, helpful and memorable. Many thanks go in particular to Davis Marques, Dr. Cheryl Qian, Roham Sheikholeslami, Maryam Maleki, and Melody Siadaty.

I also want to thank my parents for their never-ending love and encouragement throughout my entire life.

Finally, it is my pleasure to thank my husband Dr. Ali Roufegarinejad and my beloved daughter Rejina, for giving me moral support throughout my studies.
# TABLE OF CONTENTS

Approval ........................................................................................................................................ ii  
Abstract ........................................................................................................................................ iii  
Acknowledgements .................................................................................................................... v  
Table of Contents ........................................................................................................................... vi  
List of Figures ............................................................................................................................... ix  
List of Tables ................................................................................................................................... xiv  
Glossary ......................................................................................................................................... xv  

1: Introduction ................................................................................................................................ 2  
1.1 Parametric Design Modeling and Challenges ......................................................................... 2  
1.2 Using Vision to Support Decision-Making in the Design Process .......................................... 3  
1.3 Research Goals ..................................................................................................................... 4  
1.4 Research Outline ................................................................................................................... 5  

2: Background ................................................................................................................................. 7  
2.1 Parametric Modelling and Design Exploration ....................................................................... 7  
2.2 Sensitivity Analysis ............................................................................................................... 10  
2.2.1 Design Geometries and Parametric Modelling .................................................................. 10  
2.2.2 CAD-Based Sensitivity Analysis ....................................................................................... 17  
2.3 Visual Cognition .............................................................................................................. 20  
2.3.1 Limitations in Visual Cognition in Relation to Visual Sensitivity Analysis ....................... 21  
2.3.2 Visual Analytics: Improving Interaction during Sensitivity Analysis .............................. 22  
2.4 Research Objectives ............................................................................................................. 23  

3: A Method for Visual Analysis of Design ................................................................................ 27  
3.1 Visual Sensitivity Analysis of Parametric Design Models (ViSA) ....................................... 27  
3.1.1 Structural Composition for Change Analysis ................................................................. 27  
3.1.2 Process of Change Analysis ............................................................................................ 29  
3.2 Demonstration of Structuring and Using Design Models using ViSA .................................. 31  
3.2.1 Example of Control Features ....................................................................................... 31  
3.2.2 An Example Design Model and proposed ViSA Structure Implementation .................... 36  
3.3 ViSA and Design Model Structure: Designer’s Perspective .................................................. 38  
3.4 Selection and Initiation of Visualization ............................................................................... 40  
3.5 Analysis of Design Model Using Control and Visualization Features .............................. 42  
3.6 Summary ............................................................................................................................... 44  

4: Case Studies .............................................................................................................................. 47  
4.1 Bezier Curve ......................................................................................................................... 47
LIST OF FIGURES

Figure 2-1 ViSA fields of study ........................................................................................................7

Figure 2-2 Marina Bay Sands by Mashe Safdie and Associates, Source: http://www.msafdie.com. Accessed on: 15 April 2010..................................................8

Figure 2-3 HOK & Buro Happold Lansdowne Road stadium, Dublin (Left) TVS design, Dubai Towers, Dubai (Right), Source: www.bentley.com. Accessed on: 14 November 2009 ..............................................................................................................8

Figure 2-4 Dostyk Project, Almaty; Source: www.bentley.com. Accessed on: 14 November 2009 ............................................................................................................11

Figure 2-5 Sensitivity analysis methods ..........................................................................................14

Figure 2-6 Examples of solar incidence angle analysis for a tower project. “An analysis was performed for each facet of the tower model. Normal vector of each facet was compared to the direction to the sun in different times. As the angles grow, energy performance improves. This is shown by color changes from red to blue spectrum. Source: www.aecbytes.com/viewpoint/2007/issue_32.html; Accessed on: 18 October 2009. ....................................................................................19

Figure 2-7 The navigation control loop; Source: Ware, 2004 ..........................................................22

Figure 3-1 Structural model decouples model from view and controllers. Reference model refers to the original design model that its sensitivity to some of its input(s) is of interest. Target model refers to a clone of the reference model that changes are applied. Change in input(s) is controlled through control feature. Visualization feature calculates differences between reference and target models, and visualizes them to the user for analysis........................................28

Figure 3-2 Line visualization that shows the displacement vectors of the B-Spline curve. Change in perturbed point to the target model is controlled by control feature. Control feature comprises the line and the perturbed point. The sensitivity of reference model to the applied change is demonstrated by vectors in visualization feature. ...........................................................................................................29

Figure 3-3 Process model for conducting visual Sensitivity Analysis from users’ perspectives. .........................................................................................................................30

Figure 3-4 A Length control feature. This slider controls $\delta \in [0, 1]$.........................................................31

Figure 3-5 Length control feature. A slider controls length in this controller..................................32

Figure 3-6 An Angle control feature. ..............................................................................................32

Figure 3-7 Angle-Length control feature. Radius of circle $C$, $r = \delta \in [0,1]$. The $T$-value of perturbed point is controlled through values on the circle $C$.................................33
Figure 3-8 An Angle-Length control feature is linked to the line model. The black line is the reference model and blue line is the target model with its direction and its length controlled by Angle-Length control. ................................................................. 34

Figure 3-9 Azimuth-Altitude control feature........................................................................................................ 35

Figure 3-10 A line model with applied Azimuth-Altitude control feature................................................................. 35

Figure 3-11 A B-Spline surface example. .................................................................................................................. 36

Figure 3-12 The control feature comprises the circle $C$ and the perturbed point TP$_5$, centered on P$_5$. .............................................................................................................. 37

Figure 3-13 In visualization feature, the light brown colour indicates the increase in the area of the meshed polygons where the dark brown designates the decrease in the area of the meshed polygons due to the applied changes. ...................................... 37

Figure 3-14 Preparation phase activity diagram. The sequences of activities of this diagram shows the process of defining control feature, associating control feature to reference model and cloning the reference model as a target model. .......... 39

Figure 3-15 Visualization Features selection and definition activity diagram. ...................................................... 41

Figure 3-16 Analysis phase activity diagram. This diagram shows the process of sensitivity analysis for humans (not a system, perspective)................................................................. 43

Figure 4-1 Process of creating Bezier curve............................................................................................................. 48

Figure 4-2 The control polygon and all intermediate points labelled to correspond with the systolic array (Woodbury, 2005a, Woodbury, 2005b)............................................................................................................. 48

Figure 4-3 A 4 order Bezier curve creating by 4 poles P$_{00}$ – P$_{01}$ and applied Angle-Length control feature. ............................................................................................................. 49

Figure 4-4 The effect of changing parameters $T$-value and delta of perturbed point; TP$_{02}$ is fixed and TP$_{01}$ is altered ............................................................................................................. 50

Figure 4-5 The effect of changing parameters $T$-value and delta of perturbed point; TP$_{01}$ is fixed and TP$_{02}$ is altered ............................................................................................................. 51

Figure 4-6 The effect of changing parameters $T$-value and delta of control point (P$_{00}$). ............................... 52

Figure 4-7 Isometric view of a free form roof following a curved centre-line.............................................................. 53

Figure 4-8 Control points (P$_{00}$, P$_{03}$) and perturbed point (TP$_{00}$)........................................................................ 54

Figure 4-9 Control feature: Length control sliders.............................................................................................................. 54

Figure 4-10 Visualization features; two of the $X$, $Y$ and $Z$ parameters are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change.............................................................................................. 55

Figure 4-11 Visualization features; one of the $X$, $Y$ and $Z$ parameters are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change................................................................. 56

Figure 4-12 Visualization features; $X$, $Y$ and $Z$ are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change. .............................................................................................. 57

Figure 4-13 Shadow subdivisions (Jiang and Ward, 1992)............................................................................................. 58
Figure 4-14 A post casting shadow on the ground. While the light source location is changed, the position of shadow end points and shadow directions change. In this example, the designers are interested in the displacement of end points as displacement vectors ................................................................. 59

Figure 4-15 Azimuth-Altitude control feature controls the position of light source ................. 60

Figure 4-16 Change in shadow lines ..................................................................................... 61

Figure 4-17 Visualizing sensitivity of shadow lines during displacement of source light.  
(left) The angle between target and reference shadow lines; (up right) Shadow line displacements on a point; (down right) Shadow line displacements on a unit line ..................................................................................... 61

Figure 4-18 Visualizing sensitivity of shadow lines during displacement of source light.  
(left) The angle between target and reference shadow lines; (up right) Shadow line displacements on a point; (down right) Shadow line displacements on a unit line ..................................................................................... 62

Figure 4-19 Two intersecting surfaces ................................................................................... 63

Figure 4-20 Two intersecting surfaces; Control points of surface01 [P1-P9] and control points of surface02 [P10-P13]. Two Curvature-Distance controllers added to P5 and P4 ....................................................................................... 64

Figure 4-21 Intersection curve. Orange curve shows the reference intersection curve and blue curve shows the target intersection curve .................................................................................. 64

Figure 4-22 (a) Intersecting curve: orange curve shows the reference and blue curve shows the target curves of intersection; (b) The length of Surface02 isocurves to change in T-value or delta of TP4 and TP5 visualizing by bar chart .................................... 65

Figure 4-23 Interactor and References points; Their relation affects Result (Woodbury, 2007) ................................................................................................................................................. 66

Figure 4-24 Replication of result lines .................................................................................... 67

Figure 4-25 Generated roof one end of the result lines .......................................................... 67

Figure 4-26 Length control feature: Slider ................................................................................ 68

Figure 4-27 Displacement of the points on the surface during moving the delta controller slider ............................................................................................................................................. 69

Figure 4-28 Report the number of points on the reference and target surfaces that meet the specific requirements ............................................................................................................................................. 69

Figure 4-29 Quadrangular Mesh net built on the uv space of the B-Spline surface ............... 70

Figure 4-30 The Angle-Length control feature applied to P03, P04 and P06 to create the perturb points TP03, TP04 and TP06 .................................................................................................................. 71

Figure 4-31 Delta circles ............................................................................................................. 72

Figure 4-32 The effect of changing parameters T-value and delta of perturb point. TP04 and TP06 are constant and TP03 is altered; ......................................................................................... 72

Figure 4-33 The effect of changing parameters T-value and delta of perturb point. TP03 and TP06 are constant and TP04 is altered .............................................................................................................. 73
Figure 4-34 The effect of changing parameters $T$-value and $delta$ of perturb point. $TP_{03}$ and $TP_{04}$ are constant and $TP_{06}$ is altered.................................................................73

Figure 5-1 Process followed in the user study.................................................................76

Figure 5-2 Study environment..........................................................................................79

Figure 5-3 The model studying sensitivity of the angle-length ratio to unit change on $P_{01}$........85

Figure 5-4 (a) The line model provided in Set-I; (b) Same line model with visualization and control features integrated in Set-II.................................................................86

Figure 5-5 Bezier curve with four poles, $P_{01}$-$P_{04}$.........................................................86

Figure 5-6 Modified model with a Angle-Length control feature attached to point $P_{02}$........87

Figure 5-7 The sensitivity of changing point $P_{02}$ on the segment points.........................87

Figure 5-8 Torsion in the B-Spline curve changed as $P_{02}$ and $P_{03}$ locations changed..............88

Figure 5-9 (a) Coordinate systems along the curve; (b) Torsion in a B-Spline curve as two control points are changed.................................................................88

Figure 5-10(a) A given initial state of the curve (b and c) Torsion visualization after changes.................................................................89

Figure 5-11(a) Model mapping folded-planes on UV space; (b) folded-planes feature..............90

Figure 5-12 (a) B-Spline surface with nine control points; (b) The surface is divided by mapping points on the UV space to create a mesh structure.........................................90

Figure 5-13 (a) Paper-folding model with control features; (b) Visualization of edge changes.........................................................................................................................91

Figure 5-14 Quadrangular-meshed surface ..........................................................................92

Figure 5-15 Quadrangular-meshed surface with Angle-Length controllers and circular indicators.........................................................................................................................92

Figure 5-16 Benjamin Schneider-TU-Wein; Quad Meshes in Architecture; Source: www.bentley.com. Accessed on: 14 November 2009..................................................93

Figure 5-17 Data collection tools, method components, and issues..........................................94

Figure 5-18 BS’s approach for the Ratio angle/length model of Set-I ........................................96

Figure 5-19 WD’s approach in controlling ‘unit change’..........................................................97

Figure 5-20(a) ZN’s and (b) ST’s solutions for the Line model................................................98

Figure 5-21 WD’s solution for Bezier curve example..............................................................99

Figure 5-22 NN’s solution for the Bezier curve and the variable array showing displacement.........................................................................................................................99

Figure 5-23 ST’s solution for the Bezier curve using ‘vector field’ to calculate displacement.........................................................................................................................100

Figure 5-24 ST’s solution using a ‘target model’ to apply changes without changing torsion.........................................................................................................................101

Figure 5-25 ST’s solution for Paper-folding example............................................................101

Figure 5-26 The level of overall ease of (a) utilizing and (b) implementing ViSA method ......105
LIST OF TABLES

Table 5-1 Expertise domains and average experience in years ................................................................. 81
Table 5-2 Areas where parametric modelling tools used ........................................................................... 83
Table 5-3 Participants’ skills in using mentioned parametric tools ............................................................... 84
Table 5-4 Participants’ aliases and their familiarity with Parametric Modelling (PM) and Sensitivity Analysis (SA) .......................................................................................................................... 95
# GLOSSARY

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CF</td>
<td>Control Feature</td>
</tr>
<tr>
<td>GC</td>
<td>GenerativeComponents</td>
</tr>
<tr>
<td>MVC</td>
<td>Model-View-Controller</td>
</tr>
<tr>
<td>PM</td>
<td>Parametric Modelling</td>
</tr>
<tr>
<td>RM</td>
<td>Reference Model</td>
</tr>
<tr>
<td>SA</td>
<td>Sensitivity Analysis</td>
</tr>
<tr>
<td>TM</td>
<td>Target Model</td>
</tr>
<tr>
<td>VA</td>
<td>Visual Analytics</td>
</tr>
<tr>
<td>VF</td>
<td>Visualization Feature</td>
</tr>
<tr>
<td>ViSA</td>
<td>Visual Sensitivity Analysis (Method)</td>
</tr>
</tbody>
</table>
Chapter 1
1: INTRODUCTION

1.1 Parametric Design Modeling and Challenges

Parametric CAD systems provide powerful modelling functions that are more flexible than their predecessors. They can reduce, if not eliminate, much of the effort required to regenerate design models during design exploration or when changes to design are desired. The system can quickly generate variations of the model by responding to parametric changes through propagating them to the associated elements in the model. Models can incorporate parametric dependencies on data such as design requirements that are not even part of the design solution but can be used to derive the design. These systems provide multi-level interaction with design models: in addition to direct-manipulation, designers can build and change the models using transaction sequences, scripting, and custom features.

Although parametric systems have distinct potentials, they pose a number of challenges in the practice such as complexity of computational geometry, human-system interaction methods, and increasing scale in design models (Kasik et al., 2005). Today’s competitive design world leads designers to enhance their design ability (Eastman, et al., 2004); and use of parametric systems is an important advantage for designers. A related bottleneck caused by these challenges is observed during design that entails complex model building, analysis and evaluation. Complexity increases rapidly as the interdependencies between design elements grow. This may cause designers to not always be aware of the impact of changes introduced on one part of a design model to
other parts of the model. Designers typically adjust a model, observe the effects of the change by switching between different views, and then assess the consequences of that change before selecting the next action. This process is intense, tool-specific, and hinders particularly analysis-based decision-making.

At the task level, making changes to a model and observing the effects of those changes is hard to perform. Interaction with a design model in these systems is not tailored to aid the designer in perceiving the global or local effects of change made to the model. Comparison of design variations is discrete and manual: it requires the designers to manipulate models frequently while relying heavily on their memory to make comparison-based assessments. Functions provided in parametric CAD system can support change-control and visualization of change-effects to a limited degree and mainly through and on the original model. This is an inefficient approach for exploration as the original model is lost in the process that makes comparing current to previous states almost impossible.

1.2 Using Vision to Support Decision-Making in the Design Process

Designers continually make decisions that are dependent upon and coupled to other decisions. Among these decisions are parametric changes and assessing their effects on the design. A decision to alter a parameter results in a change to the model, which in turn may lead to other decisions in changing other parameters or returning to an earlier state. Current parametric CAD systems fail to facilitate this process. There are multiple solutions, one of which is increasing already complex functionality of these systems (the tool level); another one is to handle design-related changes in the human computer
interface (the model level), making the functions transparent. Both have justifiable reasons.

In this study, we choose to work on developing a method that aims at enhancing the design modelling and analysis as design progresses; and not at the tool level but at the model level. This is mainly because, every design model is unique and standard functions particularly for analysis can be limited in meeting unique analysis needs. Flexibility and custom control and visualization features are needed.

1.3 **Research Goals**

This research aims to investigate how parametric modelling CAD systems might be used more effectively to assess and make decisions about changes to design models. We believe that the weaknesses in parametric CAD systems and decision-making processes described can be overcome at the model-building phase by incorporating an iterative modelling-analysis-evaluation approach. In this study, we attempted to develop and test a prototype method that aims at alleviating these weaknesses. The method supports the control and visualization of change effects on design models for aiding analysis. It combines visual analytics and sensitivity analysis techniques in order to formalize the process of change control on parametric design models and their geometric dependencies. We contend that such a method can improve agility in parametric modelling processes and aid designers in assessing and making decisions about proposed changes before committing to them.

The method outlines the process, structure, representations and heuristics that are needed to support the decision-making on parametric changes. The method also addresses
how to manage model complexity by considering scale and level of details. Scale refers to the number of geometric elements and the dependencies between. Level of detail focuses on the detail level at which the designer interacts with the model. Both are important constraints when building and analyzing design. An example is to control and visualize the magnitude of a change in different resolutions. A significant change in one scale might be ignored at another.

1.4 Research Outline

This thesis is organized into six chapters. Chapter 2 reviews the status of published research on parametric modelling, CAD-based sensitivity-analysis and visual analytics; and enumerates specific research objectives. In Chapter 3, we describe a proposed prototyping method called ViSA, for Visual Sensitivity Analysis of parametric design models. The case studies presented in Chapter 4 introduces how the method can be applied on some basic parametric models, such as Bezier curves, B-Spline surfaces or other problem-specific design geometry. These basic parametric models are the fundamental elements for more complex design and are widely used in different application domains. Chapter 5 describes the qualitative user study we conducted to evaluate the effectiveness of the ViSA method. The thesis concludes with a summary of findings, and recommendations for future research.
Chapter 2
2: BACKGROUND

This thesis aims at studying sensitivity analysis of parametric design models. In this chapter we broadly review the literature of parametric CAD systems and sensitivity analysis methods and discuss their advantages and drawbacks. We were keen to look for supplementary methods that enhance parametric CAD systems abilities in interactive change-control process. Thus, we additionally looked for related literature on human-perception and interactive systems to tackle issues that were raised during change-control process in parametric CAD-based sensitivity analysis (Figure 2-1).

Figure 2-1 ViSA fields of study

2.1 Parametric Modelling and Design Exploration

Research on design exploration has gained attention in recent years following the emergence of advanced parametric and constraint-based computational design systems (Hernandez, 2006; Woodbury and Burrow, 2006a; Eastman, et al., 2004). Parametric design systems that support design generation are becoming a “source of inspiration” for designers (Kolatan, 2006) and are considered as tools for variable design representations.
(Woodbury, et al., 2006). These systems support creativity (Shneiderman, 2007) by enabling designers in generating, managing, and organizing highly complex design models, particularly when the “beauty” and “efficiency” of the model is also desirable (Kolatan, 2006). Figure 2-2 and Figure 2-3 show some of the most recent build-environments designed by the help of parametric systems. The forms created in the examples are not conventional and technical solutions require using complex geometry solvers.

Figure 2-2 Marina Bay Sands by Mashe Safdie and Associates, Source: http://www.msafdie.com. Accessed on: 15 April 2010

Figure 2-3 HOK & Buro Happold Lansdowne Road stadium, Dublin (Left) TVS design, Dubai Towers, Dubai (Right), Source: www.bentley.com. Accessed on: 14 November 2009
Katz (2007) identifies the difference between non-parametric and parametric systems based on how they treat design rules and geometry. Parametric modelling systems are explicit in terms of the rules and constraints between components, while they are implicit in terms of the resultant model geometry. Traditional non-parametric CAD systems on the other hand make geometry explicit while keeping relationships between components implicit. In other words, non-parametric CAD tools do not keep track of rules and constraints between components, forcing the designer to do so, manually (Katz, 2007). In non-parametric CAD systems, the designer must re-establish the design every time a change is made to the input variables.

Parametric systems have behaviours that distinguish them from conventional design systems. Kolatan (2006) lists these characteristics as “diversity”, “adaptability”, and “responsiveness”. The behaviour of models in such systems is analogous to the ‘rubber-band effect’: if the band is stretched from two ends, all the points on the band will respond to change, and as long as it is stretched the band will keep its form. Similarly, the geometric parameters on the model either trigger change, or are modified to follow other dependency changes such that the model in its entirety is kept as a coherent structure by not losing its defined characteristics (Aish, 2003). In other words, this system allows a designer to create a range of possible sketches of a single design model without the need to set up the models again from zero (Hernandez, 2006).

Similarly, parametric modeling systems provide designers with the opportunity to model, generate and modify the dependence and variations of design solutions rapidly (Aish and Woodbury, 2005); and to integrate phases of a design process. They provide
the necessary tools for the designer to create, evaluate, and modify the dependence and variations of design solutions (Qian, 2007).

In parametric design systems such as *GenerativeComponents*\(^1\) (Aish, 2003), the system is not only responsive but also adaptive to applied changes in real time. In other words, this system allows the designer to create a range of possible sketches of a single design model without the need to recreate the model again from zero (Hernandez, 2006).

### 2.2.1 Design Geometries and Parametric Modelling

Designers make decisions using different types of information at each phase of the design process. In the early design phase, decisions are often based on higher-level, abstract information as design requirements (Erhan, 2005). During solution generation, lower-level and spatial information is structured and used. Spatial information is represented in visual forms such as CAD models that describe and communicate design geometries. These are also means of exploring design alternatives.

Depending on the systems used to define design models, searches for a satisfactory solution can be costly and time consuming. It has become more common to use complex and irregular design geometries for creating objects in different design domains, such as architecture, aerospace, automotive, industrial product design, etc. (Steele, 2007). There are various motivations for using such geometries; sometimes aesthetic and sometimes functional justifications can be stated (Eastman, et al., 2008). As the complexity of design solutions increases, the cost and time required to evaluate them becomes more pronounced. Designers may need to make different versions of the model

\(^1\) *GenerativeComponents* is a registered trademark of Bentley Systems, Incorporated.
several times, potentially from scratch, in order to compare one solution to another. Regardless of the motivation, designing objects with complex geometries requires computational support such as parametric modeling systems. Dostyk Project includes free-form towers that use these systems (Figure 2-4).

![Figure 2-4 Dostyk Project, Almaty; Source: www.bentley.com. Accessed on: 14 November 2009](image)

Parametric systems can help by defining design geometries in terms of parameters and parametric associations such that they eliminate or reduce the need for reproduction of the design models when a model changes (Sacks, et al., 2004). In parametric design systems, objects are ‘intelligent’. The ‘design intent’ captured in parametric associations helps designers explore, revisit earlier decisions and improve the relationships in every stage of design process (Aish and Woodbury, 2005). Design models may help the designer to reveal and to compare the different design models in order to devise an improved design solution (Woodbury and Burrow, 2006b).

Once a model is defined, variations in design can be quickly generated by changing parametric values; and if properly defined, models can react to changes by propagating change to the associated elements in the design geometry. Although
parametric models are one of the obvious solutions, the systems for creating parametric models are not currently equipped with necessary tools to directly support design decision making. For example, comparison of design variations is modal and requires switching between views numerous times to observe and assess the effect of changes, which consequently imposes extra cognitive load on the designer. Given the complexity of design geometries, controlling the switch-observe-assess cycle hinders design exploration.

The complexity of design model representation in a switch-observe-assess loop should be reduced to set of less complex representations, which also results in reduction of the cognitive load (Bates-brkljac and Counsell, 2007). Designers hence are in need of methods that enhance their perception in predicting the effects of alterations in the design model without changing the reference design model itself (Bates-Brkljac, 2007). This suggests that we need to add sensitivity analysis as a visual overlay to parametric modelling systems.

2.2 Sensitivity Analysis

Sensitivity analysis is the process of measuring the effect of changing input variable values on the output of a design model (Ascough, et al., 2005; Saltelli, et al., 2000; Saltelli, et al., 1999; Saltelli, 2002). The term sensitivity in this study refers to the measure of change in one or more values of a parametric design model when any change is applied to its input parameters (Frey and Patil, 2002). Sensitivity analysis methods have been applied in a variety of different domains, e.g. in economics, engineering, physics, sociology, and medical decision-making (Frey and Patil, 2002; Frey, et al., 2004).
Parametric design models are typically complex. Small changes to the input parameters of a design model may result in hidden or unanticipated changes to other components of the design that may not be obvious at the outset. Sensitivity analysis in the context of parametric design may help the designer “to perceive the behaviour of the design model, the coherence between parameters in the design model as well as interaction between them” (Saltelli, et al., 1999). It may also help them make design decisions. Generally, these decisions are not only “the most important” task for designers, but also the most “difficult” ones (Arsham, 2003).

Sensitivity analysis can be performed during any stage of design development (Ascough, et al., 2005) and can identify (Saltelli, et al., 2000):

- Significant parameters that contribute the most output variability.
- Insignificant parameters that can be held constant or eliminated during evaluation of a design model,
- Whether or not there is any interaction between parameters of a design model; and if so, which group of parameters interacts?

Moreover, sensitivity analysis plays a main role in evaluating alternative design solutions (Fraedrich and Goldberg, 2000; Kleijnen and Sargent, 2000).

2.2.1 Sensitivity Analysis Methods

Sensitivity analysis methods are studied from two different but related perspectives: (a) based on application purpose and scope and (b) based on methodology used (Figure 2-5). The first perspective classifies the purpose as screening, local and global. These methods are distinguished with the scope in which they focus on, i.e. applying change at either one input at a time or applying change to a number of inputs.
They can methodologically adapt mathematical, statistical and graphical methods (Daradkeh, et al., 2008; Saltelli, 2002; Ascough, et al., 2005).

![Sensitivity Analysis: Goals and Objectives](image)

**Figure 2-5 Sensitivity analysis methods**

Sensitivity analysis methods are commonly selected with respect to different factors such as “model characteristics, the computational time needed to evaluate the model, available resources and the objective of sensitivity analysis” (Daradkeh, et al., 2008). Although, it is beyond the scope of this thesis to enter into a detailed discussion of these methods, we briefly describe them in the following section with the intention of introducing a new approach to sensitivity analysis.

### 2.2.1.1 Screening, Local and Global methods

Screening is a useful qualitative sensitivity analysis method that is generally used in early stages of modelling when a large set of data must be examined. It is used when the number of important independent parameters—that have the potential in influencing the results of the analysis—is large. Generally, in this type of sensitivity analysis, the analysis starts with an introductory screening check to recognize the most influencing parameters and hence to isolate the most non-influencing parameters in the analysis from further processing (Williams, 1963). This will save the process from computationally
expensive analysis that unnecessarily uses computing resources. In other words, utilizing this method helps to find the most sensitive inputs in the model, those that have the most significant effects on output of the model (Ascough, et al., 2005; Wagner, 2007). The qualitative nature of this method is its disadvantage since this method cannot quantitatively list the significant inputs and rank them based on their influence on the model. This qualitative method has not been utilized in this study.

The local sensitivity analysis intends to show the amount of effect that a relatively small change has on other parts of a model. Local sensitivity of a model is calculated based on first-order partial derivatives of outputs with respect to small changes in the inputs. In a design, this means studying the rate of change in the model output by varying input variables one at a time. Selection of these perturbed inputs depends on the purpose of sensitivity analysis (Frey and Patil, 2002; Kioutsioukis, et al., 2004; Gustafson and Wasserman, 1995).

Unlike local models that are commonly carried out by individually varying only one of the model inputs at a time, input parameters in the global models can be selected either individually or in a group with other input parameters. Global sensitivity is measured for the entire range of each input variable, while all the variables are varied, simultaneously (Homma and Saltelli, 1996; Sudret, 2008; Saltelli, 2002).

2.2.1.2 Mathematical, Statistical and Graphical methods

Sensitivity analysis generally uses mathematical methods in analysing the local sensitivity of output (dependent variable) to the ranges of individually varied inputs (independent variables). Mathematical methods analyze the sensitivity of model output for a few values of an input in its possible range or for a small perturbation. In other
words, this method calculates partial derivatives of model output based on the model input in a mathematical model. Although this method is a helpful means of screening the significant inputs in the models, it does not address the variance in the output due to the variance in the inputs.

In mathematical sensitivity analysis, a model is expressed through a mathematical equation; hence, modification of the model is a complex process. This method can only be used when the partial derivatives exist. In addition, this model is not preferable when the relation between the variables of the mathematical model is complex or when they related indirectly. This causes the mathematical sensitivity analysis to be usually an expensive and time-consuming process (Ababei, et al., 2007; Frey, et al., 2004; Morgan and Henrion, 1990; Brun, et al., 2001; Liebrock, 2005).

Statistical methods in sensitivity analysis are used to identify the effects of real-time interaction among several inputs in the model. One or more inputs can be varied at a time in the statistical method. Although, this method is widely used in engineering design it has some issues regarding applying it to the complex design models due to “computational and organizational” problems (Yin and Chen, 2008). In order to carry out the statistical sensitivity analysis (SSA), there is a need to simultaneously use statistical distributions assigned to the inputs as well as evaluating the impact of input variance on distribution of the output (Cullen and Frey, 1999; Andersson, et al., 2000; Frey and Patil, 2002).

In addition to the methods mentioned, there is a graphical sensitivity analysis method. This method demonstrates the sensitivity of the model using graphs, charts or surfaces due to a change in the model input. This method can be used individually or as a
supplementary method on top of the statistical and mathematical methods to visualize the
effects of variation in model inputs to the model output. A great advantage of utilizing
graphical methods for sensitivity analysis is its generality of application to a wide range
of complex models (Frey, et al., 2004; Frey and Patil, 2002).

However, in order to interactively control the effects of input on the design
model, the approach to sensitivity analysis in this study differs from the conventional use
of mathematical or statistical methods. It is, rather, a visual sensitivity method where the
sensitivity of the model is controlled and displayed by means of interactive
visualizations.

We observe that parametric CAD systems can support change control on design
models to a certain degree and this capability is not directly accessible. That is consistent
with other research findings (Hardee, et al., 1999). Although CAD systems functionality
is for geometric modeling, they are weak in controlling change on these models. To
augment the precision in change-control cycle, the functionality of parametric CAD
systems should be extended such that the control and visualization can interact with the
model while decoupling from each other.

2.2.2 CAD-Based Sensitivity Analysis

In CAD-based parametric design, the parameterization incorporates geometric
control points that are also used to create the design model as well as other input
variables deriving design (Hardee, et al., 1999). In sensitivity analysis, control point to
apply change is selected if it controls the part of geometry that designer intended to
change (Braibant and Fleury, 1984; Chang and Choi, 1992). Analysis generally computes
the vector field resulting from a change on any of the control points or variables. The vector field determines the significance of the change, its location, magnitude, and direction.

In earlier CAD-based sensitivity analysis, CAD tools represented only the surface geometry. Analysis included several steps. First, the sensitivity vector fields of points on the surface are calculated and then sensitivity vectors of interior points computed mathematically (Choi and Chang, 1994).

Visualization is an additional step usually performed outside of the model. However, new systems can integrate routines such that, when changes occur they can perform complex sensitivity calculations and visualize changes hence integrate this step to the analysis process. This feature enables designers to interactively apply changes and observe change effects as part of the design process.

A case study by Katz (2007) shows how this can be achieved using different parameterization techniques and processes. This case study calculates the sensitivity of a building’s skin to the change in solar incidence angle. Parameters are defined in the program such that they iteratively control diagrid\(^2\) proportions with regard to several aspects such as structural performance, area inside the building and desired aesthetical features (Figure 2-6).

---

\(^2\) A diagrid is a type of structural design and building construction that relies on a diagonal grid of structural elements on a façade to support the floors.
Figure 2-6 Examples of solar incidence angle analysis for a tower project. “An analysis was performed for each facet of the tower model. Normal vector of each facet was compared to the direction to the sun in different times. As the angles grow, energy performance improves. This is shown by color changes from red to blue spectrum. Source: www.aecbytes.com/viewpoint/2007/issue_32.html; Accessed on: 18 October 2009.

In the systems such as GC, SolidWorks\textsuperscript{3}, Solid Edge\textsuperscript{4} and CATIA,\textsuperscript{5} designers are able to move control points and visualize alternatives in a real-time basis. However, implementing a desired sensitivity analysis scheme is not trivial. For example, although some CAD applications allow what-if scenarios to be embedded in the model, the original model is changed during the course of analysis and therefore the designer is unable to track the change effects. Additionally, although they provide powerful modelling functions, they are weak in supporting, facilitating and visualizing sensitivity

\textsuperscript{3} SolidWorks is a 3D mechanical CAD program by Dassault Systèmes SolidWorks Corp. (Vélizy, France). Source: Wikipedia.org

\textsuperscript{4} Solid Edge is a 3D CAD parametric feature solid modelling software. It provides solid modelling, assembly modelling and drafting functionality for mechanical engineers. Source: Wikipedia.org

\textsuperscript{5} CATIA (Computer Aided Three-dimensional Interactive Application) is a multi-platform CAD/CAM/CAE commercial software suite developed by the French company Dassault Systemes. Source: Wikipedia.org
analysis directly. The bottlenecks are principally the result of system functionality, complexity and uniqueness of designs, and the limitation of human cognitive and perception systems. These bottlenecks include:

- No direct function for controlling change and change history.
- Altered original design model during analysis hence loss of reference data.
- Increasing complexity of local design decisions.
- Scale and resolution management task and limited screen space.
- Frequent shift in locus of attention on model and system (including UI).
- Cognitive bottlenecks in visual-spatial memory and visual perception.

This list is not exhaustive and among these, the most challenging are those related to human cognitive and perceptual abilities. Although we cannot directly change them, our thesis is that a parametric system can amplify some of these abilities.

2.3 Visual Cognition

Visualization extends the designer’s cognitive abilities (Bertoline, et al., 1996; Chandrasekaran, et al., 2004). It provides representations for concepts, and reveals relationships between concepts as spatial structures (Dietrich, et al., 2005). Vision, as a receptor, transfers acquired information about design model to memory. When humans visually receive data as input, that data is mapped and processed in the brain. As soon as any change in the design model happens, vision again sends the new information to the brain. Interaction adds a heavy load to the human cognition and visual perception system when the intensity and frequency of the input-process-react cycle increases (Ware, 2004). For designers, complexity in the CAD system and design model can quickly cause this to
happen, thereby hindering comparison-based decision making and sensitivity analysis during design.

2.3.1 Limitations in Visual Cognition in Relation to Visual Sensitivity Analysis

Visual sensitivity analysis of a design model requires constant attention to the model while changing and observing the effects of a change. Analysis also entails frequent switches of attention from one representation to another, and navigation of the model following the change-observe-manipulate cycle. We observe types of visually intense activities that are similar to those cited in human cognition research on visual sensitivity analysis. The literature on human cognition reveals a number of limitations of the human-visual system that are relevant to our work. First, change blindness, which Rensink (2005) describes as, “the inability to notice changes that occur in clear view of the observer, even when these changes are large and the observer knows they will occur.” The main goal in visual sensitivity analysis is to gain insight about the behaviour of a model under change for agile analysis. However, due to this limitation it is highly probable that as a consequence of change blindness a designer may be unable to perceive global or local change effects. Second, visual-spatial working memory is limited (Suwa, et al., 2000). When external visualization of a model is not capable of representing the difference between pre- and post-change information clearly, designers make use of their internal cognitive representations. However, these are not accurate and, more importantly, are subject to the limitations of visual-spatial working memory. In visual sensitivity analysis, change occurs dynamically and may or may not result in a directly observable outcome due to the magnitude or location of the change or the magnification of the view. Observing a dynamic change, as opposed to ‘intermittent’ changes (Ware,
2004), is restricted by the visual working memory’s capacity of tracking 4 to 5 objects at a time (Pylyshyn and Storm, 1988; Burkell and Pylyshyn, 1997). This impedes the recognition of important changes in a large model (Rensink, 2005). Designers need to maintain perceptual continuity in the location and time of the elements being changed.

Because parametric models are typically complex, designers frequently manipulate the view, rotating, panning, filtering, and zooming the view camera to analyze where and what changed. At the task level, this manipulation of the view creates a challenge for the visual system in that it involves a rapid shift in the locus of attention and intensive visual scan and search to gain insight about the model’s behaviour (Simons and Mitroff, 2001; Geisler and Chou, 1995; Walther, 2006).

### 2.3.2 Visual Analytics: Improving Interaction during Sensitivity Analysis

In order to tackle the challenges mentioned for CAD-based visual sensitivity analysis, we propose to utilize visual analytics techniques. Visual analytics provides methods that leverage human visual cognition by using “human-centred interactive” systems to support the process of decision-making (Thomas and Cook, 2005).

![Figure 2-7 The navigation control loop; Source: Ware, 2004](image-url)
Visual analytics methods define the process of task performance by visualization means. For example, during the interaction between (design) visualization and the user, three main loops occur (Ware, 2004). At the “low-level loop”, the designer interacts with objects by selecting and moving them using “eye-hand” coordination. At the “intermediate level loop, the exploration, navigation, and view refinement”, take place. The “higher-level loop” includes problem solving, where the observer forms hypotheses about (design) data and refines them through augmented visualization processes. These cycles are revised and replaced accordingly as new data are added (Figure 2-7). Visual information augments human cognition and supports understanding of the model, to enhance the comprehension of the complex relation between model components and the effects of change on them (Tufte and Weise Moeller, 1997; Card, et al, 1999).

Tasks related to visual sensitivity analysis will benefit from the visual design principle of applying multiple views of a design model (“windows” in Ware’s terms) rather than zooming and view manipulation to reduce visual complexity (Ware, 2004). We propose an approach that utilizes multiple visualizations for parametric controls and views generated from the parametric model, to be interacted with while analyzing sensitivity. Our approach draws on visual analytics to provide guidance to the development of these visualizations with the aim of improving task performance and comprehension of sensitivity.

2.4 Research Objectives

We present a prototyping study to formalize the process of change-control on parametric design models, and with a particular focus on geometric dependencies. The
study combines *visual analytics* and *sensitivity analysis* approaches for achieving the following objectives:

- Augment the designer’s control over changing parameters while maintaining the parametric design model’s integrity.
- Provide interactive visualization tools that support focused analysis of different aspects of the design model under consideration.
- Provide continuous feedback to support the change-analyze cycle and enhance design cognition, visual search, and decision-making.

As noted earlier, the approach to sensitivity analysis used in this study differs from the conventional use of mathematical or statistical methods. Rather, it falls under the umbrella of visual or graphical sensitivity methods, such that the sensitivity of the model is controlled and displayed visually to the designer through interactive representations that may be of the same or different type than the model itself. The designer interacts with the visualization and controls parametric variations of the model simultaneously.

We propose a set of reusable and extendible perspectives to a design model to control and visualize changes decoupled from the model. We believe that these can enhance the designer’s perception of input-output dependencies during sensitivity analysis. The goal of using various change control and visualization perspectives is to alleviate the limitations posed by the designer’s visio-spatial memory (Ware, 2004), including change blindness (Simons, 1996; Rensink, 2005), visual attention in terms of both locus and span (Itti, et al., 1998; Wolfe, 2000), and visual search (Geisler and Chou, 1995). We envision that these perspectives can reduce redundant navigation and zooming of the model and allow designers to focus on the analysis task rather than work around
application-specific view control features (Ware, 2008). In the following chapter, we
describe the proposed method in detail through a set of case studies that examine the
approach at different levels of complexity.
Chapter 3
In this section, we introduce the prototyping method we developed to improve design decision-making during sensitivity analysis of design and by utilizing visualization and control support. The goal of the method is to enhance input control of parameter on the design model when applying parametric changes and enable analysis of the effect of changes focusing on a particular aspect of the design. The method addresses how designers can control the design during sensitivity analysis and improve the representation of the design model under change. We first introduce the structural organization of design models to which we apply the method followed by our scheme for capturing design model preparation-change-evaluation activities. We then describe in a set of activity diagrams to how the activities in each of these phases can be performed.

3.1 Visual Sensitivity Analysis of Parametric Design Models (ViSA)

3.1.1 Structural Composition for Change Analysis

The ViSA method we propose adapts a structural model inspired from the Model-View-Controller (MVC) framework in software design (Reenskaug, 1979; Burbeck, 1992). The intent of this approach is to increase cohesion and decoupling of visualizations and controllers from the model such that the model contains only design-relevant information while leaving control and visualization independent (Figure 3-1).
Figure 3-1 Structural model decouples model from view and controllers. Reference model refers to the original design model that its sensitivity to some of its input(s) is of interest. Target model refers to a clone of the reference model that changes are applied. Change in input(s) is controlled through control feature. Visualization feature calculates differences between reference and target models, and visualizes them to the user for analysis.

This enables the controllers and the change-effect visualization to be modular and exchangeable. A controller is a geometric feature or graphical component that is not a part of the design model per se and may directly link with different parameters in the model (Figure 3.2). Controllers provide means to controlling precision, range, direction etc. Visualization features like controllers are geometric features or graphical components; their role here is to show changes by comparing the reference and target models, to filter and format change data as desired, and to make the effect of change visually determinable (Figure 3-2).

The term ‘reference model’ here refers to the original design model in which sensitivity of some of its input(s) is of our main interests; and ‘target model’ refers to a clone of the reference model to which changes are applied. ViSA comprises three distinct and iterative phases: model-controller preparation; visualization selection and setup; and analysis (Figure 3-2). In the first phase, a control feature from the design model where change is of interest is linked with the visualization feature. In order to keep the reference model intact during change and sensitivity analysis, we create a target model.
These concepts are demonstrated below on a simple Bezier curve model. The reference model is the black Bezier curve, which is created by four control point (P1 - P4). The control feature is linked to P3. The control feature in this example is a slider, which controls the position of the perturbed point. The target Bezier curve that is shown here as grey, is created by reference control points in which P3 is replaced by perturbed point TP3. The displacements of points on the reference Bezier curve are shown by a bar chart as a visualization feature.

Figure 3-2 Line visualization that shows the displacement vectors of the B-Spline curve. Change in perturbed point to the target model is controlled by control feature. Control feature comprises the line and the perturbed point. The sensitivity of reference model to the applied change is demonstrated by vectors in visualization feature.

3.1.2 Process of Change Analysis

Change in input(s) to the target design model is controlled through a control feature. Applying change on a design model through using control features simplifies the interaction task and increases the precision of change-control and change-analysis processes. The target model is created using the same input (e.g. geometric control
points) as those of the reference model with the exception that their corresponding perturbed points in the control feature have replaced the original control points.

In the second phase, visualization features are associated with controllers and models. Visualization features visually show the effects of applied change on the model. Visualization features can be used individually—as separate models—or attached to the design model. The selection of visualization features depends on the designers’ intent in representing the effects of change on the design model.

![Figure 3-3 Process model for conducting visual Sensitivity Analysis from users’ perspectives.](image)

In the third phase, the analysis of a model is conducted through a change-analyze cycle comprising four distinct tasks: introduction of a change to the design model through the control feature, switching focus between interaction and visualization, observing, and assessing change effects.

Through manipulating the controller, new changes are applied to the model. Simultaneously, the effects of these changes appear in the visualization features. Switching focus between control and visualization features enables the designers to observe and consequently assess the effects of the applied changes on the design model in a real time basis.
3.2 Demonstration of Structuring and Using Design Models using ViSA

3.2.1 Example of Control Features

Before we introduce a comprehensive example as to how to apply ViSA on analysis of a design model, we would like to describe the role of control features by examples. This is important as controllers are either designed and built by the designer or selected from a pre-defined list depending on the type and magnitude of change needed. In a sense, part of a parametric model becomes an interface to the model to provide designers means to control the type of change and precision on input parameters. In order to make the concept of ‘controller’ clear we describe four examples below: Length controller, Angle controller, Angle-Length controller, and Azimuth-Altitude controllers. These are selected for their simplicity; different controllers can be created depending on the model, and the part change of interest.

3.2.1.1 Length Controller

In the Length control feature, a slider controls the length of the linked object either directly or by introducing a coefficient. For example, the slider controller shown below controls unit displacement in domain of \( \delta \in [0, 1] \) of the length property of an object (Figure 3-4). The slider value changes the length.

![Figure 3-4 A Length control feature. This slider controls \( \delta \in [0, 1] \).](image)
In Figure 3-5, the change in the length of the black line is controlled by the Length control slider.

3.2.1.2 Angle Controller

An Angle control feature consists of a circle $C$ and two points: the reference control point $P$ and the perturbed point $TP$, in which the perturbed point $TP$ is orbiting around the reference control point, $P$ (Figure 3-6).
The reference control point is the center of circle $C$. The perturbed point in this control feature, controls $T$ parameter along the circle $C$ in the domain $T \in [0, 1]$ and it refers to the proportional distance along the curve between the start and end points on curve. The Angle control feature controls the direction of applied change.

3.2.1.3 Angle-Length Controller

An Angle-Length control feature consists of a circle $C$ and three points: the reference control point $P$, the perturbed point $TP$ and length controller point on circle $PL$ (Figure 3-7).

![Figure 3-7 Angle-Length control feature. Radius of circle $C, r = \delta \in [0,1]$. The $T$-value of perturbed point is controlled through values on the circle $C$.](image)

The reference control point is the centre of circle $C$ and perturbed point is orbiting on this circle around the reference control point. Length controller point in this control feature controls the radius of circle $C$. The perturbed point is used to control $T$ parameter along the circle in the domain $T \in [0, 1]$ and it refers to the proportional distance along the curve between the starting point and end point. The radius of circle $C$ is in the domain $R \in [0, 1]$. It controls the unit displacement of the perturbed point. Figure 3-8 shows an
example of Angle-Length control feature. This controller is linked to a line model to control the angle and the length of the target model.

![Diagram of Angle-Length control feature](image)

**Figure 3-8** An Angle-Length control feature is linked to the line model. The black line is the reference model and blue line is the target model with its direction and its length controlled by Angle-Length control.

### 3.2.1.4 Azimuth-Altitude controller

An Azimuth-Altitude control feature comprises two concentric circles lying in perpendicular planes (Woodbury, 2007) (Figure 3-9). The horizontal circle controls azimuth and the vertical circle controls altitude. Both circles have the same center point and radius. A control line shows the output direction. The line connects the center point of the circles to an altitude control point located on the vertical circle. To create the altitude control point on the Azimuth-Altitude control feature and generate the control line, first we create an azimuth control point on the horizontal circle. Next, we create the vertical circle with center and radius are the same as the horizontal circle and which goes
though the azimuth control point. Finally, we create the altitude control point on the vertical circle and build a line from the center to this point.

![Figure 3-9 Azimuth-Altitude control feature](image1.png)

![Figure 3-10 A line model with applied Azimuth-Altitude control feature.](image2.png)

The direction of this line is changed by moving the altitude control point on the vertical circle or by moving the azimuth control point on the horizontal circle. For example, in Figure 3-10 by linking the Azimuth-Altitude control feature to the reference model—the black line—direction of the target line—the blue line—is changed by
moving the altitude control point on the vertical circle or by moving the azimuth control point on the horizontal circle.

### 3.2.2 An Example Design Model and proposed ViSA Structure Implementation

In order to demonstrate the concepts introduced in the method, we use a B-Spline surface defined by nine control points ($P_1 – P_9$) as a reference model (Figure 3-11).

![Figure 3-11 A B-Spline surface example.](image)

The goal in this exercise is to analyze how the area of the meshes on this surface change as $P_5$ moves on a plane and in a set range. In order to meet this input control criteria, an Angle-Length control feature is composed of $TP_5$, $P_5$ and a circle orbiting $P_5$. The circle limits the movements of $TP_5$ to be on a fixed plane and on the circle (Figure 3-12).
This is followed by creating a target model as a clone of the reference B-Spline model using all control points but $P_5$, which $TP_5$ replaces. A script implements how to calculate the differences between target and reference model and how to visualize the change of the model. The effect of a change is visualized using color-coding to show increases or decreases in the area of the meshes (Figure 3-13).
3.3 ViSA and Design Model Structure: Designer’s Perspective

Below, we propose an activity model from the designer’s perspective that shows the tasks and products of activities. This is a high-level description as to how the proposed ViSA method can be implemented during sensitivity analysis. The description shows an activity diagram that includes actions taken as well as the input to and output from each activity. When needed, we refer to an example Bezier curve implementing ViSA for analysis that is demonstrated earlier (Figure 3-2).

In the model preparation phase, the designer clones the reference model as the target model as the first step of model structuring (Figure 3-14). A set of control features is associated with the target and reference model. The purpose of this replication is to keep the reference model intact while applying and comparing the changes on the target model. The control feature is used to control parameters in the target model for which sensitivity of the model to their change is of interest. For example, in Figure 3-2, $P_3$ is defined as part of the original Bezier curve (reference model) and reused as reference points for creating the control feature.
Figure 3-14 Preparation phase activity diagram. The sequences of activities of this diagram shows the process of defining control feature, associating control feature to reference model and cloning the reference model as a target model.
Reference models are kept unchanged as benchmark models to compare them computationally to their clones (target models) on which changes are performed. A target model as a ‘clone’ is includes all objects in the reference model except those of input for sensitivity analysis. Input objects are associated with objects in the control features to create the target model and manipulate it in a real time basis. In Figure 3-2, the perturbed point TP3 is an example of this type of object. Designers can edit the visibility of objects on the model to decrease the amount of information. For example, the clone Bezier curve shown by gray color or for a better control and visualization clarity its visibility can be turned on and off when needed.

3.4 Selection and Initiation of Visualization

In the second phase a set of visualization features is associated with both reference and target models (Figure 3-15). Their role is to continuously calculate the changes and visually inform the designer of the sensitivity of the model to the changes. The structure may include one or many visualization features associated with the design models.

Phase II starts with defining the purpose of visualization—why and what to visualize. It is followed by determining the visualization type and strategy. Since visualization features (objects) are independent modules from the models they can be stored in a repository. If there is a predefined visualization feature, it can be selected and connected to reference and target model for visualization. Otherwise, designer can create a visualization feature from scratch on the model-view or by writing scripts. If an existing visualization feature is used, it can be further enhanced before connecting to the design models.
The responsibility of the visualization features is to reveal changes, visually and interactively. Unlike control features, a visualization feature is associated with both reference and target models to receive input from both those models and to calculate the resulting visualization based on the differences in reference and target models. Visualization features should be strategically selected and assigned to focus on particular
locations for where change is of interest. In Figure 3-2, the changes of displacement of point on Bezier curve are shown by a bar chart. Through interacting with model and changing the place of perturbed point, change of displacement of points on Bezier curve is continuously calculated and reported.

3.5 Analysis of Design Model Using Control and Visualization Features

The visual sensitivity analysis process proposed in this study is iterative. Following the choice of parameters of study, the locations of the anticipated changes, and the appropriate visualization means, the designer starts the analysis process. While the control features provide ‘precision control’ on input, the visualization features dynamically and in real-time calculate changes in the target model relative to the reference model, and visualize these changes as specified in the visualization features internal structure.

The process can start by focusing on either control or visualization features. As needed, the designers can introduce new control or visualization features; or alternatively can refine parametric associations. These require revisiting the ‘model preparation’ and ‘selection and initialization of visualization’ phases. The change-observe-assess cycle continues until the designer gains insight about the effects of the applied change on the design model.

The process completes when the designer is satisfied, otherwise the cycle continues as illustrated in Figure 3-16 until the designer is satisfied with the perception achieved. In this case, either the whole cycle starts from the beginning and the analysis repeats until expected level of insight is achieved or the change-observe cycle repeats and
the analysis continues from the beginning of this cycle. The second scenario involves applying changes by designer on the intended parameter(s) and subsequently observing-assessing the changes that appear on the visualization feature(s), continues until needed insight is gained.

Figure 3-16 Analysis phase activity diagram. This diagram shows the process of sensitivity analysis for humans (not a system, perspective).
In Figure 3-2, the designer can analyze the sensitivity of Bezier curve through applying change using control feature, and observe and assess the effects of the change on model by means of visualization feature.

3.6 Summary

The ViSA method incorporates what-if scenarios to sensitivity-analysis during design exploration by providing control and visualization on parametric changes. The MVC framework provides a clean separation between design, control and visualization features while integrating them in a modular fashion. ViSA suggests three main phases: model preparation, initiation of visualization, and analysis phases. In the process, control features and target model are created as an extension to, but decoupled from, a reference model—the design model itself. This leaves the reference model unchanged and available for any computational comparison during change-control and change-analysis phase. On one hand, control features enable precision control on inputs on the parameters as part of the target model while on the other hand, visualization features calculates the differences between the target and reference models, and in particular where the change is expected. The effects of change on the design are visually reported.

ViSA suggests using various visualization and control features. These features can be selected from a precompiled set of features or can be defined by the designers. They can be defined geometrically or using routines that can generate them. The change-observe-assess cycle in analysis phase continues until insight gained about how the design model reacts to parametric changes on specific part of the model. In the next chapter, we experiment with how ViSA by working on case studies including different
design geometries at varying levels of complexity. The case studies demonstrate proposed structural and procedural aspects of the method discussed in this chapter.
Chapter 4
4: CASE STUDIES

We have conducted several case studies to evaluate and verify the applicability of the ViSA method. Some of the models tested in these case studies are Bezier curve, B-Spline surface, surface overlap and other problem-specific design models. These examples are selected as they are the foundation of widely used complex parametric models. The following sections present a selection of the conducted case studies.

4.1 Bezier Curve

In the process of design creation, curves are a fundamental elements. Like straight lines, they are extensively used not only in architectural design, but also in engineering, multimedia, animations, and computer art (Salomon, 2006). Generally, designers demonstrate curves by identifying a set of control points, which are an abstract form of the curves. Then an algorithm from these control points computes the curve itself.

One of the simple curve forms is the Bezier curve. In this case study, a cubic form of the Bezier curve which follows de Casteljau’s algorithm\(^6\) with four higher order control points is studied. For a given parametric value, each of the lines between points is used to create a parametric point. Each successive pair of points is used to specify points along the defining line. Then new parametric point is placed on each line (Figure 4-1).

---

\(^6\) “In the mathematical subfield of numerical analysis, the de Casteljau's algorithm, named after its inventor Paul de Casteljau, is a recursive method to evaluate polynomials in Bernstein form or Bézier curves. The de Casteljau's algorithm can also be used to split a single Bézier curve into two Bézier curves at an arbitrary parameter value.” Source: www.Wikipedia.org
Each intermediate point has a $T$-value along the assigned line. The $T$ linearly interpolates that point. The $T$-value represents the parametric distance of the point along the line, $T \in [0, 1]$. For each point, $P_j^i$, $i$ is the level in the systolic array and $j$ refers to the index of the point (Figure 4-2).

In Figure 4-3, we demonstrate a cubic Bezier curve created by the $GC$ parametric modeller from Bentley Systems, Inc. The curve is produced using four control points ($P_0^0$-$P_3^3$) utilizing de Casteljau’s algorithm.
4.1.1 Control Feature: Angle-Length Control Feature

An Angle-Length control feature is used to apply a unit change to $P_{01}$ and $P_{02}$ (Figure 4-3). The controller comprises of two points: the reference control point and the perturbed point. The reference control points, points $P_{01}$ and $P_{02}$, are replaced in the target model with their corresponding perturbed points. The perturbed points are used to control two parameters: delta and $T$. Delta refers to the unit distance between the reference control point and the perturbed point. In other word, $\delta \in [0, 1]$ is the radius of the circle. A slider controls the value of delta. Parameter $T$ is in the domain $T \in [0, 1]$ and it refers to the proportional distance along the curve between the start and end points of the curve. The sensitivity of model is evaluated by changing $\delta$ and $T$-values (Figure 4-3).

![Figure 4-3 A 4 order Bezier curve creating by 4 poles $P_{00} - P_{01}$ and applied Angle-Length control feature.](image)

4.1.2 Visualizing Sensitivity: On Model, Line, Circle, and Point

The sensitivity of the Bezier curve to displacement of the points of the Bezier curve is explored, herein. Different strategies for visualizing sensitivity are used to gain insight about the behaviour of the curve under change. The displacement vectors are demonstrated on a unit line, unit circle, on a point (hodograph) and on the model—i.e. the curve itself—(Figure 4-4 and Figure 4-5). The visualization of displacement vectors on a
unit circle, similar to that of on a unit line, shows the order and direction of vectors whereas the visualization of displacement vectors on a point (hodograph) shows the magnitude of vectors. Figure 4-5 and Figure 4-5 shows the sensitivity of the Bezier curve to changing parameters $T$-value and delta of TP$_{01}$ and TP$_{02}$.

<table>
<thead>
<tr>
<th>On Model</th>
<th>On a Unit Line</th>
<th>On a Unit Circle</th>
<th>On a Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta_{TP_{01}} = 1.0$</td>
<td>$T_{TP_{01}} = 0.625$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_{TP_{10}} = 1.0$</td>
<td>$T_{TP_{10}} = 0.375$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_{TP_{01}} = 1.0$</td>
<td>$T_{TP_{01}} = 0.125$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\delta_{TP_{02}} = 1.0$</td>
<td>$T_{TP_{02}} = 0.875$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4-4 The effect of changing parameters $T$-value and $\delta$ of perturbed point; TP$_{02}$ is fixed and TP$_{01}$ is altered

Sensitivity Issue: Points on curve displacement
Control Feature: Angle-Length controller
Control Parameter: TP$_{01}$ and TP$_{02}$
Visualization Feature: On model, on a unit line, on a unit circle and on a point (hodograph)

$\delta_{TP_{02}} = 1.0 \quad \delta_{TP_{01}} = 1.0$
$T_{TP_{02}} = 0.125 \quad T_{TP_{01}} = 0.625, 0.375, 0.125, 0.875$
Sensitivity Issue: Points on curve displacement
Control Feature: Angle-Length controller
Control Parameter: TP₀₁ and TP₀₂
Visualization Feature: On model, on a unit line, on a unit circle and on a point (hodograph)

\[
\begin{align*}
\delta_{TP₀₁} &= 1.0 \\
T_{TP₀₂} &= 0.500, 0.750, 0.000, 0.250
\end{align*}
\]

4.1.3 Results of Sensitivity Analysis of Bezier Curve

The analysis shown as a vector field reveals that the Bezier curve is highly sensitive to change on T-value of the perturbed point controlling P₀₁ and P₀₂ (Figure 4-4 and Figure 4-5). The direction of the vector field moves dramatically as the T-value changes.
Chapter 4. CASE STUDIES

Figure 4-6 The effect of changing parameters $T$-value and delta of control point ($P_{00}$)

Sensitivity Issue: Points on Curve displacement
Control Feature: Angle-Length controller
Control Parameter: $TP_{00}$
Visualization feature: On model, on a unit line, on a unit circle and on a point (hodograph)

$TP_{01}$ and $TP_{02}$ are constant and $TP_{00}$ is altered

$\delta_{TP_{00}} = 0.0, 1.0$  \hspace{1cm} $\delta_{TP_{01}} = 1.0$  \hspace{1cm} $\delta_{TP_{02}} = 1.0$

$T_{TP_{00}} = 0.000, 0.250, 0.000, 0.750, 0.500$  \hspace{1cm} $T_{TP_{01}} = 0.875$  \hspace{1cm} $T_{TP_{02}} = 0.125$
4.2 Free Form Roof

In this case study, we explore the sensitivity of a free-form roof that follows a spatially curved path in a landscape (Woodbury, et al., 2007). The B-Spline curve is employed as an abstracted form of landscape in this model. The 3 order B-Spline curve is created using four top-level control points \( P_{00} - P_{03} \) and a knot vector or \{0,0,1,1,1\}. Figure 4-7 shows the control points, curved landscape as well as the roof.

![Image of Free Form Roof](image)

**Figure 4-7 Isometric view of a free form roof following a curved centre-line**

4.2.1.1 Control Feature: Length Control Feature

While manipulating the \( x \), \( y \) and \( z \) parameters of the control points \( P_{00} - P_{03} \) various roof designs will be generated. In order to create the target design model, first, perturbed points need to be defined. The coordinate of a perturbed point is composed of the coordinate of corresponding reference control point plus parametric values. For instance, here point \( P_{00} \) is altered as \( TP_{00} \) (Figure 4-8) and the perturbed point coordinate is:

\[
\begin{align*}
\text{Point}_{TP_{00}.X} \text{Coordinate} &= \text{Point}_{P_{00}.X} \text{Coordinate} + x \\
\text{Point}_{TP_{00}.Y} \text{Coordinate} &= \text{Point}_{P_{00}.Y} \text{Coordinate} + y \\
\text{Point}_{TP_{00}.Z} \text{Coordinate} &= \text{Point}_{P_{00}.Z} \text{Coordinate} + z
\end{align*}
\]
The values of $x$, $y$ and $z$ are added to the coordinate of the reference control point to compute the coordinate of the perturbed point. In order to control the change at the control points sliders are used (Figure 4-9).

![Control feature: Length control sliders](image)

### 4.2.2 Visualizing Sensitivity: Color Coding, Vector Fields, XY-Plane Projection

By manipulating the mentioned perturbed points, the designer might want to evaluate different aspects of change effects on a design model, for example, displacement of the projected points on curve in $XY$ plane, change in area of roof panels or displacement of points on a roof. To visualize the sensitivity of the roof, we make use of different visualization strategies. Figure 4-10, Figure 4-11 and Figure 4-12 shows sensitivity vector fields as well as color-coded panels to visualize the sensitivity of the roof under the applied change.
Figure 4-10 Visualization features; two of the X, Y and Z parameters are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change.

Sensitivity Issue: Points on roof displacement, Displacement of projected points of surface on XY plane, Area

Control Parameters: TP00
Control Feature: Length Control Feature
Visualization Feature: Color-coding, on model, on XY-Plane
Sensitivity Issue: Points on roof displacement, Displacement of projected points of surface on XY plane, Area

Control Parameters: TP_{90}

Control Feature: Length Control Feature

Visualization Feature: Color-coding, on model, on XY-Plane

Figure 4-11 Visualization features; one of the X, Y and Z parameters are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change.
Figure 4-12 Visualization features; $X$, $Y$ and $Z$ are equal to 1.0 unit. Green color in color-coding visualization shows increase in area of polygons under the applied change.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Analog Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>$Y$</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>$Z$</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

Sensitivity Issue: Points on roof displacement, Displacement of projected points of surface on XY-plane, Area

Control Parameters: $TP_{00}$
Control Feature: Length Control Feature
Visualization Feature: Color-coding, on model, on XY-Plane

visualize the sensitivity of the roof under the applied change.

By moving the sliders, instantly the entire visualization feature is updated and the sensitivity of the roof to the applied change is computed and displayed by vector and color-coding strategies. For instance, any increase or decrease in amount of the area of each panel of the roof will be shown by the different colors. The aim is to perceive the effects of change at a glance. Vector fields aim to identify which parts of the roof panel displace more than others.

### 4.2.3 Sensitivity Analysis Results of Free-Form Roof

The analysis shows that moving the $TP_{00}$ along the $Y$ direction has less of an effect in area of each polygon in comparison to moving $TP_{00}$ along the $X$ or $Z$ directions (Figure 4-11). Additionally, by looking at the vector field we are able to find the direction of displacement of points on roof as well as the magnitude of change.
4.3 **Shadow Lines**

Shadows are an important issue in architectural design. Shadows occur where objects totally or partly occlude direct light from a light source. Jiang and Ward (1992) divided shadow boundaries to four subdivisions: shadow-making lines, shadow lines, occluding lines and hidden shadow lines (Figure 4-13). The study of shadow lines is main interest.

![Figure 4-13 Shadow subdivisions (Jiang and Ward, 1992)](image)

In this case-study we simulate a row of posts casting shadows on the ground and study the effect of change in light source on shadow lines (Woodbury, 2007). The posts are shown by parallel lines as an abstraction form of posts. The lines stand vertically on the $XY$-plane. The $XY$-plane simulates the landscape in this example and shadows are cast onto it.
In order to be more clear, firstly, we study shadow line of a one post while the light source changes the location. Figure 4-14 shows the post. The reference shadow line is shown in blue. Through displacement of light source, the target shadow line is created and shown in green. In this study, the displacement of endpoints of shadow lines as well as the angle between reference and target shadow line is in our interest.

Figure 4-14 A post casting shadow on the ground. While the light source location is changed, the position of shadow end points and shadow directions change. In this example, the designers are interested in the displacement of end points as displacement vectors.
4.3.1 Control Feature: Azimuth-Altitude Control Feature

As mentioned in Section 3.2.1.4, an Azimuth-Altitude control feature comprises two concentric circles lying on perpendicular planes. The horizontal circle controls azimuth and the vertical circle controls altitude. Both circles have the same center point and radius. A control line shows the output direction. The line connects the center point of the circles to an altitude control point located on the vertical circle. By linking the Azimuth-Altitude control feature to the reference model, the position of a light source will be changed through manipulating the location of the two control points (Figure 4-15).

![Azimuth-Altitude control feature](image)

**Figure 4-15 Azimuth-Altitude control feature controls the position of light source**

4.3.2 Visualizing Sensitivity: Visualization Features

The sensitivity of the shadow lines to changing the position of light source is visualized through using different strategies (Figure 4-16). In this example, displacement of reference shadow lines visualized on a unit line and a point. The angles between reference and target shadow lines during the applied change are reported by text,
instantly. The Azimuth-Altitude controller is independent from model and it is not part of model per se (Figure 4-17 and Figure 4-18).

Figure 4-16 Change in shadow lines.

Figure 4-17 Visualizing sensitivity of shadow lines during displacement of source light. (left) The angle between target and reference shadow lines; (up right) Shadow line displacements on a point; (down right) Shadow line displacements on a unit line
4.3.3 Sensitivity Analysis Results

Figure 4-17 and Figure 4-18 visualize how sensitive the shadow lines are to a change in direction of light source. Information such as changing the angle of the shadow lines and, their maximum displacement are interactively reported to designer. In other words, by comparing the features, the designer gains insight into which parameter alteration does and which does not have a significant effect on shadow lines.

4.4 Intersecting Surfaces

The intersecting surfaces case study shows the sensitivity of intersection between two surfaces to a change applied to either or both surfaces. In particular the sensitivity of the length of the left part of Surface02 is studied here (Figure 4-19). Of course, this
example is specific to two surface patches that intersect such that Surface01 divides Surface02 into two parts. Such surfaces intersect along a curve that is a function of \( U \) and \( V \) in Surface02 \((f(U, V))\). Note that \( f(U,V) \) divides Surface02 on space into two. We call these the right and left parts of Surface02.

Figure 4-19 Two intersecting surfaces

4.4.1 Control Feature: Angle-Length Control Feature

As can be seen in Figure 4-20, there are two intersecting surfaces that both are created by control points – \([P_1 - P_9]\) and \([P_{10} - P_{13}]\) –. When a change is applied to the control points (\(P_4\) and \(P_5\)), the shape of intersection and the size of the divided surface are changed (Figure 4-21).

In order to analyze the sensitivity of a surface to a unit change on delta and T-value of \(P_4\) and \(P_5\), the Angle-Length control feature and several visualization features are selected. \textit{Delta} refers to unit distance between reference and perturb point and \(T\) refers to proportional distance (\(T \in [0, 1]\)) along the curve—circle in this control feature—, respectively.
Figure 4-20 Two intersecting surfaces; Control points of surface01 [P₁-P₉] and control points of surface02 [P₁₀-P₁₃]. Two Curvature-Distance controllers added to P₅ and P₄.

Figure 4-21 Intersection curve. Orange curve shows the reference intersection curve and blue curve shows the target intersection curve.
4.4.2 Visualizing Sensitivity: Curve Graph and Bar Chart

The sensitivity of intersecting curve between two surfaces and the length of the left part of Surface02 isocurves to any change in the $T$-value or $\delta$ of TP$_4$ and TP$_5$ are visualized using graphs (Figure 4-22).

Figure 4-22 (a) Intersecting curve: orange curve shows the reference and blue curve shows the target curves of intersection; (b) The length of Surface02 isocurves to change in $T$-value or $\delta$ of TP$_4$ and TP$_5$ visualizing by bar chart

(1) $\delta_{TP4}=1.0$  $\delta_{TP5}=1.0$
$T_{TP4}=0.130$  $T_{TP5}=0.130$

(2) $\delta_{TP4}=1.0$  $\delta_{TP5}=1.0$
$T_{TP4}=0.675$  $T_{TP5}=0.130$

4.4.3 Sensitivity Analysis Results

There could be several references to the benefits of using this method in visualizing the sensitivity. With the use of this method, designers are able to interactively...
compare the resulting shape of intersection with its original one; hence, one can change the position of control points to result in a similar curve for the intersection of two surfaces. In addition, visualizing length of surface isocurves through bar charts with reference benchmarks on it helps the designer to get insight about the sensitivity of this model under applied change.

4.5 Reporter of Points on Surface

The following case study visualizes sensitivity of a roof surface that is created by a function (script) to the height of supporting lines. To create such a roof we make use of Woodbury’s lift example (2007). The roof surface is created on the endpoints of the (result) lines that are used as ‘lifts’ for the roof surface. The height of the line is a function of the distance between interactor point and the start point of the lines as reference (Figure 4-23).

![Figure 4-23 Interactor and References points; Their relation affects Result (Woodbury, 2007)](image)

Equation (4-1) calculates the heights. Please note that the equation is taken as is for demonstration purposes, its logical connection design is ignored in this case study.
After replicating the result line Figure 4-24, the surface is generated at the end points of these lines (Figure 4-25).

\[
\text{Height of lines} = \frac{5}{(\text{Distance (reference, interactor)} + 0.05)}
\]

Equation 4-1

---

**Figure 4-24** Replication of result lines

**Figure 4-25** Generated roof one end of the result lines.
4.5.1 Control Feature: Length Control Feature

In this case study, we analyze the behaviour of a designed surface while manipulating the result line creator function. As mentioned the reference result line is created based on Equation (4-1). We define a control variable \( \text{delta} \in [0, 1] \) and use it as a unit-distance coefficient in Equation (4-1) (Figure 4-26). The new Equation (4-2) changes the lengths of the result lines without changing the location of the reference and interactor points, hence, live the model intact.

\[
\text{Height of lines} = \frac{5}{(\text{Distance (reference, interactor)} + 0.05) \times \text{delta}} \quad \text{Equation 4-2}
\]

![Figure 4-26 Length control feature: Slider]

4.5.2 Visualizing Sensitivity: On Model Displacement Vectors and Text Reporter

As mentioned earlier, the slider in Length control feature controls generating the target model. The effects of change are visualized using text and vector fields. Figure 4-27 shows the displacement of the points on the surface while moving the \( \text{delta} \) controller slider. Figure 4-28 reports the number of points on the reference and target surfaces that meet the specific requirements, in this case study specific heights for points is being considered. By comparing two lines, which show the number of points on the reference and target surfaces, the designer realizes the effect of change on the design model.
Figure 4-27 Displacement of the points on the surface during moving the $\delta$ controller slider

$\delta = 0.7$

$\delta = 0.2$

RM: 23 points with the height of 1.8
TM: 6 points with the height of 1.8

RM: 23 points with the height of 1.8
TM: 17 points with the height of 1.8

Figure 4-28 Report the number of points on the reference and target surfaces that meet the specific requirements
4.6 **Elongation of Meshed Surface Components**

This case study focuses on how ViSA can be used with a design model incorporating a quadrangular mesh surface and controlling its behaviour under change. A B-Spline surface is constructed using six poles (points P01-P06) then the surface used to create quadrangular-meshed nets on its uv space. In this case study we test the sensitivity of the elongation of edges of the meshed net to the changes on the poles (Figure 4-29).

![Figure 4-29 Quadrangular Mesh net built on the uv space of the B-Spline surface.](image)

**4.6.1 Control Feature: Angle-Length Control Feature**

Three Angle-Length controllers are attached to the poles defining the B-Spline surface. Target Model as the clone of this surface replaces its poles with the perturbed points on the controllers. A target surface is implemented through applying individual deltas as well as $T$-values to change the position of the perturbed points (Figure 4-30). Different target models are generated by changing delta and $T$-value of perturbed points.
4.6.2 Visualizing Sensitivity: Circular Indicators (on Edges)

Circular-Indicator shows the change of length of a linear object using circle on the object changed. The center of the circle is located at the middle of each line. The radius of the circle is calculated differently. For this study, we used Equation 4-3

\[ r = k \left| \frac{l_0 - l_1}{l_0} \right| \]

Equation (4-3)

where \( l_0 \) refers to the reference length for the component and \( l_1 \) refers to the corresponding component length after change. \( k \) is an externally chosen scale factor. For example if \( k = 0.5 \), it scales the radius of circle to half. The goal is to enhance the change visualization. Red color in the circle shows an increase in length and blue color decreased length (Figure 4-31). Using Circular-Indicator strategy, the effects of change in elongation of components of quadrangular-meshed nets are visualized in Figure 4-34.
Figure 4-31 Delta circles

Figure 4-32 The effect of changing parameters $T$-value and $\delta$ of perturb point. $TP_{04}$ and $TP_{06}$ are constant and $TP_{03}$ is altered;

Sensitivity Issue: Elongation in Quadrangular-Meshed Components
Control Feature: Angle-Length Control Feature
Control Parameter: $TP_{03}$, $TP_{04}$ and $TP_{06}$
Visualization Feature: Circular-Indicators

$\delta_{TP_{03}} = 1.0$  $\delta_{TP_{04}} = 1.0$  $\delta_{TP_{06}} = 1.0$

$T_{TP_{03}} = \text{altered}$  $T_{TP_{04}} = 0.000$  $T_{TP_{06}} = 0.000$
Sensitivity Issue: Elongation in Quadrangular-Meshed Components
Control Feature: Angle-Length Control Feature
Control Parameter: TP03, TP04 and TP06
Visualization Feature: Circular-Indicators

\[ \delta_{TP03} = 1.0 \]
\[ T_{TP03} = 0.000 \]

\[ \delta_{TP04} = 1.0 \]
\[ T_{TP04} = \text{altered} \]

\[ \delta_{TP06} = 1.0 \]
\[ T_{TP06} = 0.000 \]

Figure 4-33 The effect of changing parameters \( T \)-value and \( \delta \) of perturb point. TP03 and TP06 are constant and TP04 is altered.

Sensitivity Issue: Elongation in Quadrangular-Meshed Components
Control Feature: Angle-Length Control Feature
Control Parameter: TP03, TP04 and TP06
Visualization Feature: Circular-Indicators

\[ \delta_{TP03} = 1.0 \]
\[ T_{TP03} = 0.000 \]

\[ \delta_{TP04} = 1.0 \]
\[ T_{TP04} = 0.000 \]

\[ \delta_{TP06} = 1.0 \]
\[ T_{TP06} = \text{altered} \]

Figure 4-34 The effect of changing parameters \( T \)-value and \( \delta \) of perturb point. TP03 and TP04 are constant and TP06 is altered.
4.6.3 Sensitivity Analysis Results

The sensitivity of quadrangular meshes to changes in \( \text{delta} \) and \( T \)-value of incorporated three Angle-Length controllers is examined in this case study. By looking to color coded circular indicators the place of increasing and decreasing the values are identified. Figure 4-32, Figure 4-33, and Figure 4-34 show the effect of changing parameters \( T \)-value and \( \text{delta} \) of perturb points \( \text{TP}_{03} \), \( \text{TP}_{04} \), and \( \text{TP}_{06} \). The visualization feature in this study helps to recognize which parts of the meshed model experiences more decrease or increase. With this method, designer can distinguish the change effects on model, instantly.

4.7 Summary

Implementing the ViSA method was examined in this chapter. We explained how ViSA method was implemented in parametric design models, in three steps: applying control feature(s), introducing visualization feature(s) and finally analyzing sensitivity of the design models to the applied changes. Although the selected cases were simple, they were fundamental in complex geometry designs. In these examples, we created the target models using object-copy and input-replication techniques. In order to be more precise in the structure of ViSA implementation, and to decrease the complexity of those that occurred by adding compound geometries to the design models, simple control features were selected and used in these case studies.
Chapter 5
5: EVALUATION OF VISA METHOD

5.1 Design of User Study

The main goal of the user study is to receive feedback from the users of parametric modelling tools to assess how the visual sensitivity-analysis method proposed augments the users’ control of change and change visualization, and if the proposed continuous change-analyze cycle has potential. In particular, the objectives are to reveal structural, procedural, and (perceived) usefulness evaluation of the method.
Chapter 5. EVALUATION OF ViSA METHOD

Structural evaluation shows that what users understand from the model and applied ViSA on it. Similarly, procedural evaluation show how users understand the implementation of a specific ViSA on a design model and finally, usefulness evaluation studies how ViSA answers questions of ViSA utility in the world. In order to achieve these objectives, we designed a three-part qualitative user study. An extensive quantitative study, which is needed for more conclusive results, would take longer, and more importantly would not provide the immediate response we need for further improving the method. Figure 5-1 shows the flow of evaluation activities with input and output artefacts.

5.1.1 Part I: Structural Evaluation

In Part-I, the participants were given four parametric models with increasing levels of complexity built in GC: a line, a Bezier curve, a B-Spline Curve with torsion, and feature mapping on a 3D surface (cf. Section 5.3). Participants were not required to construct the models. This is because, on one hand, we wish to focus on control and analysis rather than model building; and on the other hand, the models were to be quickly modified, when needed. There were two sets of four tasks on these models. In the first four tasks in Part I, the participants were asked to demonstrate their approach to analyzing the sensitivity of the models given. While these tasks were designed to reveal possible overlaps between our method and the participants’ methods, they also prepared the participants for the next set of tasks.

In the second set of tasks in Part-I, the participants were given the modified versions of the same models adapting ViSA we proposed. They included visualization
and control features directly accessible on the models. We intend to introduce the method to the participants and seek information as they interact with them.

5.1.2 Part II: Procedural Evaluation

In the second part of the user study, the participants were given a transaction file that builds a meshed-surface model, a clone of the model, and control and visualization features. The transaction file is the record of every action user performs in GC. The transaction file simulates the model building for visualization and control purposes as proposed by our method. Our goal was to receive feedback from the participants regarding the complexity of the procedural aspects. The participants were asked to ‘rerun’ the transaction file and observe how the ViSA method is implemented on the model through the transaction sequence (cf. Section 5.3.5).

5.1.3 Part III: Usefulness Evaluation

The participants were given a survey querying the usefulness of the method in relation to the tasks performed. In addition, the participants had an opportunity to report their post-task opinion about advantages and disadvantages of the method. They were encouraged to make comments about the issues they have observed during the study and report them to the researchers.

5.1.4 Process Details

Prior to the user study, we carried out a pre-test survey to collect background information about participants. The questions were addressing their active design domain, expertise, experience, knowledge about parametric design and sensitivity analysis. We explained to the participants their ethical rights and they completed an informed consent
form. The participants were given a handout introducing the study and tasks. Since we did not want to influence the participants with our ideas that may differ from their own, we encouraged them to ask questions and waited for them to initiate conversation about the tasks. We observed the participants as passively as possible, and only interrupted them when it seemed they needed help in understanding the instructions or controlling software features. We allocated 60-80 minutes for each participant. However, this varied from 30 minutes to 150 minutes. In order to encourage participation, we offered gift cards as a token of appreciation of their time.

5.1.5 Setting Details

The study environment included a desk to accommodate one participant and the research assistant, a computer, a camcorder, and notebooks. We conducted the experiment at different locations to encourage participation. However, at each location, we had the same setting as shown in Figure 5-2.

![Study environment](image)

Figure 5-2 Study environment

We provided a computer with GC installed. During the study a research assistant was next to the participant, continuously monitoring and helping during the process when
needed. A digital camcorder was used to record participants’ interaction with the given parametric models and to record participants’ communications throughout the session. The participants were provided with paper for their sketches and written comments.

5.1.6 Ethical Considerations

Prior the user study, we received the ethics approval from the Office of Research Ethics at Simon Fraser University in accordance with university policy R20.01.

During the course of the user study, we ensured the participants that their personal information would remain confidential. Participants attended the user-study individually and their information and input were not disclosed to each other. We kept the collected feedback in the office in a secure file and participants had full access to their own information.

5.2 Participants: Selection and Characteristics

Parametric modelling and sensitivity analysis are two methods applied in design domains such as architecture, product design, civil and mechanical engineering. We chose our participant base to reflect this diversity as much as possible in the given limited resources. We believe that this was essential to reveal applicability and deficiencies of the method from a broader perspective; the visual sensitivity analysis method we proposed is not domain-specific and is meant to be used in parametric modelling in general.

We approached designers who were particularly experienced in using ‘digital’ design modelling tools. In this user study, ten designers with different backgrounds volunteered: three females and seven males, ages between 27 and 40. We believe that the number was sufficient to gather enough data to study the method proposed and satisfy the
Chapter 5. *EVALUATION OF VISA METHOD*

The qualitative nature of this work. Our goal is not to perform an extensive ‘usability’ study, but to understand how the designers would receive the method and comment on our research goals.

5.2.1 **Expertise and Experience in Parametric Modelling**

The participants were given a list of design domains and asked to rank their experience in each. Three participants selected their expertise as civil, mechanical, and computer engineering. We had seven participants from architectural design. An interesting result emerged when participants had reported expertise in one primary domain, but also reported different levels and years of experience in working in other domains. The table below shows expertise levels and average years of experience of the participants.

<table>
<thead>
<tr>
<th>Expertise Domains</th>
<th>Novice</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Expert</th>
<th>Average Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>2</td>
<td>-</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>Product Design</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Engineering</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>-</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Visual/Graphics</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>Software</td>
<td>1</td>
<td>1</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>5</td>
</tr>
</tbody>
</table>

The expertise level was presented in five-level Likert scale: the participants were asked to mark “1” for novice and “5” for expert to indicate their expertise domains and level. They were also asked to note the number of years of experience in each design domains. If a participant did not select any expertise level or did not report years in
experience, they were excluded from the averages shown on the table. The maximum
experience was reported 17 years for architectural design and 12 years for engineering.

5.2.2 Parametric Modelling and Sensitivity-Analysis

The participants were asked to mark their expertise level in parametric modelling
on a five-level Likert scale where “1” indicating novice and “5” expert levels. Four
participants reported they were novices in parametric modelling and three ranked
themselves as experts. Similarly, we asked their experience in sensitivity analysis on a
five-level Likert scale where “1” corresponds to ‘unfamiliarity’ and “5” indicates ‘high
familiarity’. The concept of “sensitivity analysis” raised confusion initially among the
participants due to the technical complexity implied by the term. The majority of the
participants noted that they were unfamiliar with sensitivity analysis. Only one
participant indicated ‘high familiarity’ and two participants reported they were ‘some-
what familiar’. Although during the user study we observed the participants performed
‘operations’ that essentially are sensitivity analyses of the models from our perspective,
they hesitated to identify themselves as doing so. We attribute this hesitation to
designers’ feeling their use of the term could be different from the one we used. Hence,
their ‘labelling’ of these operations were not overlapping with ours.

5.2.3 Use Areas of Parametric Modelling

Parametric modelling is not only used for design but also for research, education,
and tool (software) development. In order to determine the relevancy of these areas to our
participants, we asked them to rank the purposes for which they use parametric modelling
(Table 5-1). Six participants reported they mostly use parametric modelling for research;
seven employ it in design modelling. One of the participants noted that the main purpose of using these tools is for design software development. The participants also reported that they use parametric tools for educational purposes such as teaching or learning how to model. One participant reported visualization as another purpose for using parametric models.

Table 5-2 Areas where parametric modelling tools used

<table>
<thead>
<tr>
<th></th>
<th>Least Relevant</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Most Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>2</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Education/Teaching</td>
<td>3</td>
<td>-</td>
<td>2</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Design</td>
<td>1</td>
<td>-</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Software/Tool Development</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>Other</td>
<td>1</td>
<td>-</td>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

5.2.4 Parametric Modelling Tools Used

We gave the participants a list of parametric design tools to understand their familiarity with these tools and ask their skill-level in using them. In addition, we encouraged them to include other parametric tools they use that are not in the list. The most known and used parametric modelling tools among the ten participants were GC followed by Revit\(^7\). Use of SolidWorks and CATIA was almost similar. The participants indicated Rhino\(^8\) (with Grasshopper) and Cinema4D\(^9\) as the two other tools they use for

---

\(^7\) “Autodesk Revit is Building Information Modelling software for Microsoft Windows, currently developed by Autodesk, which allows the user to design with parametric modelling and drafting elements. Building Information Modelling is a CAD paradigm that allows for intelligent, 3D and parametric object-based design.” Source: www.Wikipedia.org

\(^8\) “Rhinoceros (Rhino) is a stand-alone, commercial NURBS-based 3-D modelling tool, developed by Robert McNeel & Associates. The software is commonly used for industrial design, architecture, marine design, jewellery design, automotive design, CAD / CAM, rapid prototyping, reverse engineering as well as the multimedia and graphic design industries.” Source: www.Wikipedia.org

\(^9\) “Cinema 4D is a professional 3D modelling, animation and rendering software, developed by MAXON, widely used in the film, television, and video game industry. It is known for its fluid dynamics, particle simulation, and extensively customizable user interface.” Source: www.maxon.net
parametric modelling (Table 5-1). Although two participants reported they use
\textit{AutoCAD} \textsuperscript{10} for parametric modelling, due to its minimal direct capabilities for this
purpose, we de-emphasized it in our research. However, we note that there are parametric
modelling methods in \textit{AutoCAD} at the programming level.

<table>
<thead>
<tr>
<th>Table 5-3 Participants’ skills in using mentioned parametric tools</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Catia</td>
</tr>
<tr>
<td>SolidWorks</td>
</tr>
<tr>
<td>GenerativeComponents</td>
</tr>
<tr>
<td>Revit</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>

5.2.5 Summary

The ten participants selected for the user study represent a broad cross section of
the user population of design tools. Design domain expertise and computational tool use
were evenly distributed: we had both experts and novices in the group. The diversity in
design domain is also acceptable. Particularly, interdisciplinary design expertise among
the group was a desirable characteristic as they were able to look at ViSA from different
perspectives.

\textsuperscript{9} “CINEMA 4D or C4D is a commercial, cross-platform, high-end 3D computer graphics application,
produced by MAXON Computer GmbH of Friedrichsdorf, Germany. Noted for its flexible interface and
ease of use, it is capable of procedural and polygonal/subd modelling, animating, lighting, texturing and
rendering.” Source: www.Wikipedia.org

\textsuperscript{10} AutoCAD is a CAD software for 2D and 3D design and drafting, developed and sold by Autodesk, Inc.
5.3 Models Used in the Study

The models used during the user study were designed to demonstrate change control and change visualization for sensitivity analysis. The method presented in this study was new to the users, at least the part which how the models are configured and used for sensitivity analysis. Therefore, we chose ‘generic’ design-geometry problems that can be understood by designers from different domains, and that may appear in different contexts. In order to focus on the method rather than geometric complexity, we incorporated consistent control-features and visualization features across the models. In the following, we briefly describe the models that we prepared and used during Parts I and II of the user study.

5.3.1 Elongation-Angle Ratio Control in a Line

In this model, we created a line $L$ by using two control points, $P_{01}$ and $P_{02}$ (Figure 5-3). The task given to the users is to study the sensitivity of the angle-length ratio when a unit change is applied on $P_{01}$. The ratio is shown as $R = \alpha/l$ where $\alpha$ is the angle between $L$ and $Y$-direction and $l$ is the length of line $L$.

![Figure 5-3 The model studying sensitivity of the angle-length ratio to unit change on $P_{01}$.](image)

We asked participants to show how they would control changing the location of $P_{01}$ one-unit in any direction, and to demonstrate how they would study the effect of the applied change on the angle-length ratio dynamically (Figure 5-4a). The modified version
of this model incorporates elements proposed in ViSA. Figure 5-4b shows the same model with a unit Length-Angle control feature and bar-label visualization. The modified model was described to the participants before they performed the task.

![Figure 5-4](image)

**Figure 5-4** (a) The line model provided in Set-I; (b) Same line model with visualization and control features integrated in Set-II.

### 5.3.2 Displacement Control in a Bezier Curve

In the second model, we created a 2D Bezier curve using four poles, P₀₁-P₀₄. The curve was divided into ten parametrically equal segments using a series of points (Figure 5-5).

![Figure 5-5](image)

**Figure 5-5** Bezier curve with four poles, P₀₁-P₀₄

We asked the participants to apply a unit displacement on P₀₂ and to study the sensitivity of the displacements of the segment points on the curve in reaction to this change. As described previously, the participants were first asked to show or propose
their own solution to this task. The modified model uses the ViSA method components: a control feature for applying one unit change on \( P_02 \) as well as a set of visualization features for demonstrating the displacement of points on a Bezier curve (Figure 5-6). The visualization includes ‘on the model’ (Figure 5-7a), ‘on a line’ (Figure 5-7b), ‘on point’ and ‘on a circle’ features (see Section 4.1.3 for details).

Figure 5-6 Modified model with a Angle-Length control feature attached to point \( P_02 \)

The vector field shown on Figure 5-7b, magnifies the change shown on Figure 5-7a, to make the visualization independent from the view-scale of the model and to compare the direction changes.

Figure 5-7 The sensitivity of changing point \( P_{02} \) on the segment points

### 5.3.3 Torsion Control in a B-Spline Curve

This model tests how the participants would study the effect of non-planarity of control points. Torsion occurs when, as a point moves along a curve, the \( Z \)-axis of the
associated Frenet frame rotates about the frame’s tangent vector (Figure 5-8). We wanted to receive feedback from the users as to how ViSA method can facilitate control and visualization of ‘torsion occurrence’ in the curve. The same Bezier curve with four poles P₀₁-P₀₄ is adapted from the previous model.

![Figure 5-8 Torsion in the B-Spline curve changed as P₀₂ and P₀₃ locations changed.](image)

We asked the participant to relocate P₀₂ and P₀₃ one unit in 3D and to study the sensitivity in torsion occurrence and its location. In other words, they were expected to show if and where the torsion occurs. Following their solution proposed or implemented, the model adapting ViSA method was given. The control and visualization features of this model are shown in Figure 5-9b. In order to enable one unit change on points P₀₂ and P₀₃, we used the Angle-Length control feature (Figure 5-9a).

![Figure 5-9 (a) Coordinate systems along the curve; (b) Torsion in a B-Spline curve as two control points are changed.](image)
The changes in the Z-directions of points along the curve are calculated by creating an array of coordinate systems mapping to each point along the curve. Given the parametric properties of the points, the coordinate systems react to changes introduced on the control points \( \text{P}_0 \) and \( \text{P}_3 \). Figure 5-10b shows these coordinate systems. The change in torsion is visualized as a vector field. A given initial state of the curve in Figure 5-10a shows some minimal torsion in the beginning of the curve. When the control points are moved, the torsion change is visualized by significant direction change of the Z-direction vectors (Figure 5-10b and Figure 5-10c).

(a) (b) (c)

Figure 5-10(a) A given initial state of the curve (b and c) Torsion visualization after changes

5.3.4 Edge-Length Control in Paper-Folding Model

The first three models demonstrate how ViSA could be applied on individual elements of a design’s geometry. However, design models are complex in real life, and we present in the following case studies how the proposed method can address this complexity. In order to get feedback from the users as to the methods applicability in realistic design models, the fourth model introduces a 3D surface on which a feature of folded-planes in UV space is array-mapped (Figure 5-11a). The folded-plane consist of a base polygon and six other polygons that are parametrically associated with the base polygon (Figure 5-11b). The folded plane after defined as a new feature is arrayed on the
surface following the method described in ‘place holder design pattern’ (Woodbury, 2007). The resulting model is similar to space-frame structures covering large spaces like stadiums, museums, and train stations (Glymph, et al., 2004; Menges, 2006).

![Model mapping folded-planes on UV space; (b) folded-planes feature](image)

Figure 5-11(a) Model mapping folded-planes on UV space; (b) folded-planes feature

The model structure has three basic components: a B-Spline surface, UV mapping of points on the surface to create a meshed structure, and UV mapping of the folded-plane feature as array on the meshes. The B-Spline surface is constructed using nine control points, \( P_{01}-P_{09} \) (Figure 5-12a).

![B-Spline surface with nine control points; (b) The surface is divided by mapping points on the UV space to create a mesh structure.](image)

Figure 5-12 (a) B-Spline surface with nine control points; (b) The surface is divided by mapping points on the UV space to create a mesh structure.

The parametric model propagates any change applied on the poles to the mesh structure by changing the size of the polygons of the meshes and hence the folded-planes.
Any change on the surface makes the edge lengths of the polygons of the paper-folding module change. The task given is to control the effect of the unit displacement of the points $P_{02}$ and $P_{03}$, and to observe the sensitivity of the edge lengths of the folded-planes.

![Image](a) Figure 5-13 (a) Paper-folding model with control features; (b) Visualization of edge changes

This was a relatively a complex task requiring iteration on each folded-plane and on the edges of the polygons composed in it. Therefore, rather than expecting all participants to implement their solutions, we preferred to have them choose any approach in describing their solution methods. In the modified version of this model, the point $P_{04}$ is controlled by an angle-distance controller and $P_{05}$ with an Azimuth-Altitude controller (See Section 4.3.1). The change visualization was a histogram of edge lengths (Figure 5-13).

### 5.3.5 Elongation Control in the Components of a Meshed Surface

In Part II, the participants were asked to replay step-by-step a previously recorded set of transactions that builds a B-Spline surface with a meshed grid structure and observe
how the ViSA method is applied incrementally (Figure 5-14). The B-Spline surface is built using six control points $P_{01}$-$P_{06}$. A quadrangular meshed grid was constructed on the surface using lines. The modified version of the model incorporated three Angle-Length controllers on $P_{03}$, $P_{04}$ and $P_{06}$ and circular indicators attached to each edge as visualization features (See Section 4.6).

![Figure 5-14 Quadrangular-meshed surface](image)

**Figure 5-14 Quadrangular-meshed surface**

![Figure 5-15 Quadrangular-meshed surface with Angle-Length controllers and circular indicators.](image)

**Figure 5-15 Quadrangular-meshed surface with Angle-Length controllers and circular indicators.**

The elongation change is measured on each line and the diameter of the circles were used to show this change where length increases are shown in red, and decreases in
blue (Figure 5-15). Figure 5-16 shows a practical example of quadrangular-meshed structure.

![Figure 5-16](image.png)

Figure 5-16 Benjamin Schneider-TU-Wein; Quad Meshes in Architecture; Source: www.bentley.com. Accessed on: 14 November 2009

### 5.4 Research Tools and Data Collection

We collected data utilizing five primary tools: CAD models created and used by the participants, questionnaires, sketches and notes, video recording during the user testing, and notes taken by the research assistant. The data collected from these tools covered different aspects and was used for validation and verification purposes. For example, the recorded interaction with the system was useful if the participants’ solutions were implemented, otherwise their sketch-based descriptions had to suffice in understanding their intent. Participants were free to sketch their own ideas and write their comments while interacting with the given models, which helped us to verify if a participant’s response was consistent. The participants filled a questionnaire at the end. The video recordings were mainly used to resolve unclear issues that were raised by the participants and recorded with the other tools.
5.4.1 Data validation

In order to confirm the validity of our qualitative study, we focused on data triangulation. This method is listed among other five different types of triangulation (Guion, 2002). Data triangulation involves the use of a variety of data sources in a single research study. By combining and analyzing information gathered from the data sources together the triangulation was performed. The tools and topic mapping is shown in Figure 5-17.

![Figure 5-17 Data collection tools, method components, and issues](image)

As mentioned, we evaluate the ViSA method in three parts: structural, procedural and usefulness evaluations. Evaluation of each part utilized information from different tools such as recorded videos, written notes, sketches, models and recorded verbal comments. These information pieces were cross checked for consistency and relevancy. For example, during usefulness evaluation we checked the data gathered from participants’ sketches, their answers to the questionnaire and some notes. To validate the collected data, first we extracted the data from each individual participant’s sketches.
Then we checked the data against their answers to the questionnaire. Finally, we compared the gathered data with our notes.

In addition to triangulation applied on the data obtained from each participant, in our generalized findings and conclusions we applied triangulation of information provided by all participants. All ten participants completed the same set of tasks during the user study that contributed to the confidence in the user study results.

5.5 Findings: User Study

We summarize the results of ViSA evaluation under three categories: structural, procedural and usefulness (Figure 5-1). The summary includes details of the application of the method on models and how it can augments participants’ control on change and change visualization. The participants’ expertise and experience in design fields were incorporated when verifying the relevancy of the comments made.

<p>| Table 5-4 Participants’ aliases and their familiarity with Parametric Modelling (PM) and Sensitivity Analysis (SA) |
|---------------------------------|-------------------|-------------------|</p>
<table>
<thead>
<tr>
<th>Alias</th>
<th>Group of Architects</th>
<th>Familiar with PM</th>
<th>Familiar with SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>NN</td>
<td>✓</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>DR</td>
<td>✓</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>EN</td>
<td>✓</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>ZN</td>
<td>✓</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>TN</td>
<td>-</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alias</th>
<th>Group of Engineers</th>
<th>Familiar with PM</th>
<th>Familiar with SA</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS</td>
<td>-</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>GL</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>WD</td>
<td>✓</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

As mentioned earlier in Section 5.2, we had ten participants in this user-study. Three of these participants were (civil, mechanical, and computer) engineers and the
other seven participants were architects. We divided the participants based on their expertise in two main groups: the group of engineers and the group of architects. In addition, we assigned participants an arbitrary alias that we use to hide their real identities (Table 5-4). Table 5-4 also shows their familiarity with parametric modelling and sensitivity-analysis that are the two key focus points in this study.

5.5.1 Findings: Structural Evaluation

5.5.1.1 Participants’ Approach to Change-Control and Visualization

In the user study, participants used different approaches depending on their expertise domains when completing the Set I tasks of Part I. For example, BS and GL from the group of engineers chose mathematical solutions while architects chose sketches and visualization as their intended solution. Figure 5-18 shows BS’s mathematical approach. BS used a spreadsheet to calculate changes.

![Figure 5-18 BS’s approach for the Ratio angle/length model of Set-I](image)

BS attempted to use several applications. Initially, he worked with a common CAD application to draw a line and read its properties (length and angle) from the line.
dialog box. He entered the values to a spreadsheet manually, on which he applied other changes. He commented that the CAD tool he used is not suitable for the task given; therefore, he noted a need for additional tools with some scripting capabilities to calculate ‘algebraic deformation’. It took him 43 minutes to implement his approach while the average time for the participants was 30 minutes.

While most of the participants had some level of difficulties in understanding the concept of “unit change”, WD had adapted this concept quickly. WD had more parametric design experience than the other engineers. Although his solution on the model missed visualization of changes, he provided sketches as a means for controlling unit change (Figure 5-19).

![Figure 5-19 WD's approach in controlling 'unit change'](image)

GL’s proposed solution was based on a mathematical formula to calculate change. However, he was not able to implement the solution due to its complexity. NB and TN from the group of architects were unfamiliar with parametric design. Although these participants had considerable design experience, their contribution to this study was rather limited. TN, by making use of coordinates of points and dimensions of components, suggested a solution for the first two tasks: line and Bezier curve models.
The other members of the architects, who had parametric design experience, were able to provide plausible solutions, and in some cases presented approaches overlapping with what we presented in this study. For the first example, in Line model they were able to calculate the ratio by utilizing (graph) variables in GC. In addition, they were able to visualize the change by using text, lines on the model (Figure 5-20), or watching the change in graph variable ‘window’.

Particularly, the solutions presented by YM included a ‘reference-target model pair’. The visualization calculated the change and displayed on the same model (Figure 5-20a). RS, on the other hand, showed a control feature that keeps the change in a one-unit range and he suggested visualization of the change on a line (Figure 5-20b).

For the second task of Set-I—the point displacements on a Bezier curve—BS used his earlier strategy: he used spreadsheets to calculate the displacement values. He exported the point coordinates to a spreadsheet from GC; and as he changed the positions of control points, he exported the new positions of the points to the same spreadsheet. In a sense, the participant kept a ‘reference model’ in the spreadsheet such that the difference between initial and changed states could be calculated. However, this method
was confusing and time consuming considering export operations and manually calculating the effect of a change. *WD* presented his solution on a sketch that showed a ‘control’ circle that was attached to a pole of the curve (Figure 5-21).

![Figure 5-21 WD’s solution for Bezier curve example](image)

He suggested ‘recording each segment’s position’ to calculate the change. However, neither *BS* nor *WD* presented a visualization mechanism. The architects presented their ideas by sketching or implementing the solution in *GC*. *NN* used a variable ‘array’ that is generated by *GC* (Figure 5-22).

![Figure 5-22 NN’s solution for the Bezier curve and the variable array showing displacement](image)
The length of the array was equal to the number of points on the curve. Each cell (index) stores ‘the displacement value’ of each point after calculating the change in points with respect to a reference model that was kept intact. This approach fundamentally overlaps with how we propose utilizing target-reference models. Although visualization was limited in NN’s solution, it included the elements that are similar to the ones used in our method: it calculates change of interest and displays the change.

Figure 5-23 ST’s solution for the Bezier curve using ‘vector field’ to calculate displacement

ST and DR also used a ‘reference’ and ‘target’ model, but their visualization utilized a vector field on the model rather than ‘text-based’ display (Figure 5-23). In addition, ST integrated a controller to limit the position change on the control points.

As the models’ complexity increased, the participants’ solutions lack clarity. For the third task, the Torsion model, although the participants understood the concept of torsion in the ‘beam’, most of their solutions were rather abstract. DR, BS, GL, and WD suggested changing the position of the control points on the XY-plane and avoiding movement in the Z-direction. However, they were unable to show how to control the change or display the ‘torsion’. On the other hand, RS solution proposed using ‘fixed pole points’ in the Z-direction using GC’s ‘fix point’ function. He used a ‘control’ point that was connected to the fixed poles with two lines. Along the lines, he created two
additional points one unit away from the fixed poles. Similar to our method, he proposed to use a new Bezier curve as target model. This curve used the two new points as its poles (Figure 5-24). He assumed that the change in ‘torsion’ would not take place hence he ignored the visualization.

![Figure 5-24 ST's solution using a ‘target model’ to apply changes without changing torsion](image)

ST and BS sketched their ideas as they explored solutions for the forth task, the paper-folding model. ST set up three ‘sliders’ to control change on each control points in one unit range. However, it was not clear how the direction of change would be controlled (Figure 5-25).

![Figure 5-25 ST's solution for Paper-folding example](image)

He also suggested a ‘heat map’ for visualizing the sensitivity. ST’s solution shows similarities to the ‘magnitude change’ through ‘sliders’ that we showed in the case
studies. BS proposed using spreadsheet following the mathematical approach to solve this problem.

5.5.1.2 Set II: Use of pre-created models

In Set II of Part I the participants were asked to perform the same tasks in Set I, but this time we provided them with the previously created models adapting the ViSA method using the control and visualization features developed during the case studies. They interacted with the models and gave feedback as to how the structural components of the method helped them to achieve the tasks. They also had chance to compare their own solutions to the ones adapting ViSA method. We summarize their feedback in the following section.

5.5.1.3 Structural Issues: Composition, Control and Visualization Features

The comments we received included references to the approaches proposed by the participants when evaluating the interaction with the models. The group of engineers emphasized that utilizing the proposed method was easier than working with spreadsheets, programming, and other methods in the engineering software. Architects had mixed reports, but mainly found visualization and control helpful. Below we categorize the comments in three groups: composition of the structure, controller and visualization features.

Composition of the structure: Overall, the participants considered the structure of ViSA method straightforward and logical and agreed that it showed changes visually while making it easier to observe the extremes. However, they also found the composition of the structure hard to formulate, especially for complex designs. GL said
the method was helpful in simplifying design modifications. Similarly, DR, an architect, described the model structures as “expressive, explicit and manageable”. EN described the behaviour of system as “apparent”, and added that the method provides means for “exploration”. On the other hand, WD mentioned adapting the method requires a “solid geometric background” because geometric dependencies are not obvious on the model itself. ZN emphasized a need for occasional scripting to achieve visual description of the change-made problems.

Controllers: The control features presented in the user study were limited to mainly the ‘Angle-Length’ controller. Therefore, we observed that participants who were less familiar with parametric modelling had confusion about their role in applying change, and more importantly the other possibilities for control mechanism. For example, BS complained about how the (circle) controllers made the structure “complex” by isolating what is controlled from what controls. TN and NB, who were architects unfamiliar with parametric modelling, argued that although the control features enabled them to manage directly the effects of change on the design model, they made the interaction more complex. WD added also that if there are multiple controllers the sequence of applied change becomes difficult to track. However not all comments were negative about the controllers. EN observed that the controllers allow methodological “exploration” of design variations, but he pointed out the lack of affordances and precision on these features. WD found change-control using control features was “direct and intensive”.

Visualization: The participants found the visualization features generally helpful. EN’s comments emphasized the visibility of “the consequences of changing on design
model quickly”. *EN* noted a high vulnerability of confusion for the designers, as the model, control and some visualization features are all geometric properties and that they are all created as part of the same model. *RS* emphasized that he liked the visualization and noted it served its purpose. He suggested that he might “add and modify it a little bit and playing (sic) more with colours”. Although *NB* and *TN* complained about the increased complexity of models due to visualization features, both agreed with others on their usefulness in change-control. All in the group of engineers found the visualization features “informative and supportive” of change-control, and suggested improvements for precision input.

The common opinion was that ViSA method augments change-control by helping them to predict the consequences of change applied on the design models. Although, the participants found the structure with proposed control-visualization features complex, they noted several times that the complexity could be justified by its utility. The main perceived disadvantage was “cluttered” views and potentially expensive computation, which can slow down the design process. We acknowledge that the added ‘geometric’ features to enable sensitivity analysis makes models larger, but on the other hand, they reduce complexity in searching the effects of parametric changes.

### 5.5.2 Findings: Procedural Evaluation

This time participants were asked to re-execute a transaction file in order to get insight about how the proposed structure can be composed of control features, target model and visualization features. The transaction file creates a quadrangular-meshed surface, which the edge lengths are of interest (c.f. Section 5.3.5). The participants
completed a questionnaire following this part. Below we summarize our findings regarding how the participants address the procedural issues related to implementing and using ViSA method. The issues are grouped under the groups of preparation, change control and visualization, and analysis.

5.5.2.1 Preparation phase

To evaluate the overall ease of using ViSA method, we asked the participants to assess the overall ease of implementing and utilizing this method on the models on a five-level Likert scale, where “1” indicates extremely difficult and “5” extremely easy. The results are shown in Figure 5-26. Figure 5-26a shows the ranking of the overall ease of utilizing ViSA method and Figure 5-26b shows the ranking of the overall ease of implementing ViSA method.

![Figure 5-26 The level of overall ease of (a) utilizing and (b) implementing ViSA method](image)

The main criticism for the preparation phase was about how the target model was created and how this creation added to the complexity of the design model. DR from the group of architects mentioned that although creating the target model through deep copy was interesting it could be very expensive to apply it to the entire model. EN added that the model has to be purposefully constructed from the outset if analysis is intended. NN
added that she preferred “free controllers” instead of the Angle-Length controller thus enabling changes other than one unit. BS mentioned he would use ViSA method if it comes as part of the platform and he added he would not implement it himself. ZN said, “the transactions were mostly straight forward... and [the sequence] followed logical steps... [that was] easy to follow”.

The participants’ feedback show us that although implementing The ViSA method was logical and easy to apply, there were some concerns with cloning the reference model to create the target model in the preparation phase. The limitations rise mainly due to the increase in the complexity and cost of the design model.

5.5.2.2 Change and change-visualization phase

The engineers and architects showed different reactions to visualization. While the engineers were looking for specific visualization of changed values, the architects were interested in improving the change visualization by better use of colours, shapes and texts as part of the control and visualization features, and on the model itself. The engineers preferred to see additional ‘data’ or other types of information on the visualization features, and the architects focused on how the data can be represented. Regardless, however, both groups found the control-visualization tasks logical. They also proposed improvements to the techniques we implemented. WD suggested visualizing a range of possible changes for each manipulated component. In addition, he suggested a way to visualize the effects of a change on each parameter separately. EN also suggested ‘more distinguishable control and visualization features’ from the model itself. EN also added ViSA method extends the GC capabilities.
The participants commented on the need for showing small and invisible changes and their observation. EN noted, “it [the model with ViSA method implemented] showed real-time magnitude of changes without changing the system’s clarity”. GL agreed that the visualization features helped to perceive the “change magnitude and direction”. ST stated the structure “helps seeing and observing change beforehand [before committing to a change]” and facilitate “thinking more about the behaviour of the model and communicating more effectively in a team by showing the effects of change on model”. Similarly, BS and WD described the representation of change on visualization features is “helpful” and “a logical means to express change”. However, EN added that it was hard to tell where “infinitesimal small changes” occurred in the model. BS like EN was also interested in seeing “unclear small changes [invisible on the model]”. These comments are parallel with the objectives of this method for adapting the visual analytics approach: ‘discovering unexpected’ changes. We also agree with the participants who noted that there must be a strategy adapted for selecting control and visualization features as part of analysis activity, so the process can yield useful result.

5.5.2.3 Analysis phase

The concept of ‘sensitivity analysis’ was ambiguous and confusing for some of the participants. For instance, even though NN followed similar solutions in Set-I of Part-I to our proposed solutions, she did not call what she did “sensitivity analysis”.

DR from group of architects stated that she had some problems in freezing and analyzing the visualization features; a screen capture engine or text representation could be a better option. VS’s suggestion to create a series of change states is a parallel
comment to DR’s. EN proposed a second level analytical support that “provides additional details required for [further] analysis of sensitivity”. ST suggested using multiple windows in the analysis phase to avoid view cluttering.

5.5.3 Findings: Usefulness Evaluation

To evaluate how ViSA achieves the objectives set, we conducted a post-study survey. We asked the participants if they believe the introduced method enhanced their change-control on the models. The question in the survey used a five-level Likert scale where “1” indicating strong disagreement and “5” indicating strong agreement. Three out of ten participants strongly agreed that ViSA improved change-control, while five agreed. Two participants who were architects with no parametric design experience selected “3” (Figure 5-27a).

![Figure 5-27](image)

**Figure 5-27** (a) The level ViSA method enhanced participants’ change-control on the models, (b) the level ViSA method enhanced participants’ perception of the effects of change on the models

Similarly, we asked the participants to evaluate if the proposed method enhanced their perception of the effects of change on the given models. Four participants strongly agreed with the statement and three participants agreed by selecting “4” (Figure 5-27b). Three participants were neutral: one was from group of engineering and two were from group of architects. In general, the participants agreed on ViSA’s capability in enhancing
change perception. In the questionnaire, we also asked the participants if ViSA method enabled gaining insights about the behaviour of model and making decision in the change process. Four participants strongly agreed and the other four participants agreed that their understanding of the models’ behaviour improved. The remaining two participants from group of architects remained neutral (Figure 5-28a). These answers show ViSA has potential in supporting insight gaining and decision-making.

The participants were asked to rank on a five-level Likert scale their agreement with the prospective usefulness of ViSA in controlling and predicting complex design models. On the scale, “1” corresponds to strong disagreement and “5” corresponds to strong agreement. Nine participants agreed with ViSA’s potential usefulness, which out of nine, six selected strong agreement and other three selected agreement. One participant who was unfamiliar with parametric modelling chose neutral (Figure 5-28b). The participants mainly agreed on the potentials of ViSA in supporting control and predicting the behaviour of complex design model.

Like the previous questions, we asked the participants to choose how likely they would use the proposed method in their design. On a five-level Likert scale “1” indicated
most unlikely and “5” most likely. Five participants chose that they would likely or most likely use the method in design while four participants stayed neutral. One of the architects mentioned that because she was not interested in using parametric modelling in design, she would be unlikely to use ViSA (Figure 5-29).

![Figure 5-29 The level of the likelihood using proposed method in future by participants](image)

### 5.6 Conclusion

When the participants were working on the simple models, their suggested solutions were plausible. However, their need to have a set of robust and reusable tools for sensitivity analysis became clear as the complexity of the models increased. The user-study provided important feedback about ViSA for sensitivity-analysis on parametric modelling. A summary of findings indicating the benefits and limitations of this method as well as some general suggestions are given in the following.

#### 5.6.1 Benefits of ViSA

The participants found ViSA method and its elements intuitive and logical. These elements present potentials to augment change-control and demonstrate consequences of changes before committing them. ViSA elements are means for exploration by extending the $GC$ capabilities. In particular, the control features were similar to what the
participants suggested in their solutions. These features can directly manage the properties of a change on the design model such as magnitude, direction, location etc. The visualization features were noted to be informative and supportive toward change-perception and insight gaining about models’ behaviour. In the analysis phase, most of the participants noted that composition of different visualization features enhanced their understanding of where changes could take place and the possible measure of these changes.

The structure of the ViSA was found expressive, explicit and manageable. It can eliminate relying on external tools to evaluate change effects by using ‘pre-defined’ features in the same system. Therefore, ViSA seems plausible. The engineers expressed their wish to have ready-to-use ViSA-type functions in the existing design tools. Through utilizing ViSA, real-time change on models can be visualized without changing the integrity of the model. In addition, ViSA can facilitate focusing on the behaviour of the model when introducing discrete changes as part of ‘what-if’ scenarios. The utility of ViSA is expressed by the participants noting they were likely to use ViSA in their future designs.

5.6.2 Current Limitations

The main limitation of ViSA is cluttered views and potentially expensive computation, which can slow down the design process. Particularly, the criticism focuses on the preparation phase as to how the target model is created and how this creation added to the complexity and cost of the design model. In relation to this, there is a high vulnerability of confusion for the designers: the target model, control and visualization features are geometry-based or –driven. They are created as part of the same design
model and the same ‘transaction’ history regardless of the intention of decoupling these from each other.

Since particularly visualization features receive model elements as input and do not change model states, isolating them from the model when needed is easy by toggling their view or suppressing transaction. However, control features are usually tightly coupled with the target model, hence disabling them can require removing their relationship to the model. For novice users, this becomes a complex task. In most of the cases, in particular in the complex designs, adapting the method requires a solid geometric background as the composition of the structure can easily become hard to formulate.

Furthermore, visual descriptions of these geometries need occasional scripting that requires a different mode of thinking during design. Combining multi-level interaction with the ViSA elements, and therefore with the design model itself increases the cognitive load dramatically for the new and novice users and users not familiar with programming methods. ‘Reuse’ and ‘modularity’ of ViSA elements can reduce this load, yet it is hardly eliminated.

5.6.3 Suggestions for improvements

The user study revealed important limitations and encouraging benefits of ViSA. However, most importantly, the results of this study suggest possible improvements that can make ViSA more robust and applicable. For instance, editable view properties can help designers manage the ‘appearance’ of geometric elements on the screen during sensitivity analysis and the decision making process. The ‘editing’ could be at any stage
during ViSA, for instance, during cloning the reference to create the target model or during creating the control or visualization features. The modifications consist of the changes in the colour, transparency and scale of the parts of features.

In order to edit and modify, it is suggested to use palette and lenses, which are the set of objects, or properties those can be added to view to support the designers during sensitivity analysing the dataset (Figure 5-30). The palette consist of lenses, the virtual looking glasses, which can be dragged directly into a visualization, and edit the way design geometries are drawn. These lenses can be used to emphasize details with different colors. Some applications allow using multiple lenses, and allow combining them by overlapping (Bier, et al., 1993).

As can be seen in Figure 5-30, the palette and tools can be designed to apply changing the graphical properties, such as line style, transparency or color of the selected design geometry. Designers also by using the functions on the shape and property palette will be able to replace geometric elements and their scales in control and visualization features—through polymorphism—as such precision can be taken under control.

![Color Palette](image)

Figure 5-30 Sketch of palette to modify properties of view

As part of the analysis activity, there is a need for selecting control and visualization features that can manage changes at different resolutions. This becomes more pronounced when observing small changes that need a different type or configuration of
visualization feature than relatively large changes. This can be resolved differently. One solution is to provide a set of predefined ‘customizable features’ where resolution can be adjusted. The visualization and control features we use have this property, but in terms of types, they are limited. The other option is to introduce functions at the system level. For example, the participants suggested incorporating a ‘magnifier’ tool that can query details of data in a certain scope and location (Figure 5-31).

![Figure 5-31 Magnifier lenses on visualization feature](image)

Alternative visualization features can also help. One way to achieve this would be to provide patterns for selecting visualization features for a specific purpose (Woodbury et al., 2007). The flexibility of ‘parametric modelling’ systems can enable users to create their own visualization features, but a set of predefined visualization features for general-purpose change-controls is required (c.f. Section 3.4).

We acknowledge that the most important issue with ViSA is the cluttered or ‘over-loaded’ model-view, not to mention the visual design of the system’s user-interface. A model-view includes target and reference models, control and visualization
features, and reference geometry such as point handles all at once. Some participants suggested using multiple views to organize the layout.

Although using different views might logically partition the screen, it can result in other problems related to changing the locus of attention during the course of sensitivity analysis (Raskin, 2000). Therefore, we believe that this suggestion would make the problem further. Although there is no direct remedy to solve the cluttering problem—as it also exists in other application areas—a partial solution can be toggle-able views of features and transparency (Freiler, 2008). To reduce visual clutter without removing elements from the scene, adjustable view details of control and visualization features can be used (Figure 5-32a). However, toggling off a feature neither should make it completely disappear from the view nor should it be confused with other geometric features. In addition to toggle-able and transparent views, ‘brushing’ is suggested. It is a technique to help highlighting and detecting possible unclear changes (Keim, 2002). This technique can be useful in visualizing the sensitivity on the model when we want to demonstrate both reference and target models together (Figure 5-31b).

![Figure 5-32](image)

**Figure 5-32** (a) Details of control points and control feature (b) Details of control points and control feature removed from the view
The ViSA method aims at demonstrating the individual effects of each parameter on the design model. For instance, in quadrangular-meshed surface example, we can move and reset each Angle-Length controller independently and separately observe the effects of changing each controller on the entire design model. In relation to this, the participants suggested tracking applied changes in a hierarchical structure such that the changes and their results can become traceable. The structure can be expressed in different forms, which a simple representation can be an ordered set. For example, if the $x$ value of point $P_k$ changes $n$ times, it can be shown as $P_k \cdot x = \{v_0, v_1, v_2, ..., v_n\}$. Information about the history of applied change is demonstrated in Figure 5-33, which changing the location of $P_{01}$ equal to one unit, causes the length of $L$ to vary in the range of (min Length, max Length) with $L_{\text{min}}=9.659$ and $L_{\text{max}}=11.428$, respectively.

![Figure 5-33 Additional information; history of change](image)

Another suggested improvement relates to displaying ‘magnitudes’. In order to increase the precision in the visualization and control features, additional enrichments like adding change-value or other optional information can be utilized. Figure 5-33 to Figure 5-35 show some sketches of additional enrichments. Figure 5-34 shows the addition of information in the control feature while Figure 5-35 highlights the addition of the change-value to the visualization feature.
In addition, cloning the reference model to create the target model without increasing the complexity needs special attention. This is perhaps the most difficult activity suggested in ViSA. We have also experienced serious problems during the cloning phase when we were working on the case studies and the models used in the user study. ‘Cloning’ is relatively easy and straightforward for simple geometric models. The participants noted the problem but we did not receive any suggested solutions. However, our suggestion is to implement a ‘system’ level functionality that can duplicate the model by passing ‘replacement’ parameters. This functionality can implement a ‘deep-copy’ technique used in object-oriented programming. However, implementation of such a
function will be ‘tool-’ and ‘programming language-’ specific and goes against the generalization aimed at ViSA method. We have described the benefits, current limitations and suggested improvement for ViSA based on carried out user studies. However, the need exists for advance investigation of ViSA and more formal studies to reveal other suggestions for improvement. Currently, we mainly focus on the usability of ViSA, not on the solution for control and visualization techniques for different types of design; however, these techniques are open for further improvements. We have only demonstrated some solutions to describe the problem areas more clearly.
Chapter 6
6: SUMMARY AND CONCLUSION

6.1 Summary

Switching to new generation of parametric CAD systems from traditional non-parametric CAD tools gives designers an opportunity to explore design variation without the need to rebuild the design model every time from scratch. These tools with their generative nature help designers, especially architects and engineers, to change and create new models on a real time basis. When the complexity of the design models increases, the dependencies between components of model also become complicated. Although the new generation of parametric CAD system can improve designers’ performance, these systems create new challenges.

The unclear relation between components of parametric design models, which requires frequently switching and manipulating the views to discover these relations while changing the components of the design model, makes for some difficulties in using these tools. Knowing the consequences of change, especially in complex design models and before committing to apply it directly to the design models, assists designers during their design process and decision making. As mentioned, the complex nature of parametric design exploration needs to evaluate model’s response to any change of its inputs. We are informed about the effects of changing input on the output of the design model by using sensitivity analysis techniques.

Although, traditional sensitivity analysis methods such as statistical, mathematical as well as earlier visual methods provide powerful feedback during sensitivity analysis,
they are weak on the functionality needed to facilitate and visualize the sensitivity of the design model to an applied change. Other problems in using these methods are their inability to keep the original model intact while controlling the change, as well as their inability to control change history. In addition, these sensitivity analysis methods increase designers’ cognitive load while analyzing the sensitivity of the design model. In order to get insight about the behaviour of the design model during change-control and decision-making processes of finding design solutions, we first tried to adapt sensitivity analysis methods in parametric design models and discover the advantages and disadvantages of this approach.

Later, based on our findings from the literature of previous studies, we created a method that visually analyses the sensitivity of parametric design models by combining visual analytics techniques and sensitivity analysis methods. We were particularly interested, at this stage, in understanding whether or not utilizing visual analytics techniques would improve the way that the designers analyze the sensitivity of the design models.

Then, we set up a study by introducing the so called ViSA method. We presented a prototyping study to formalize the change-control process on parametric design models with the main focus on how to augment designers’ cognition during sensitivity analysis of design models. Initially, we implemented ViSA on simple parametric models such as Bezier curves and B-Spline surfaces. Gradually, we created various parametric models, which included ViSA capabilities. In addition, we used parametric models that were created by other designers and applied the ViSA method on them. In order to show the process of implementing ViSA, part of these parametric models were included as case
studies in Chapter 4 of this thesis. In order to understand how ViSA method can be received by designers, we carried out a preliminary user study, which its results shown in Chapter 5. The utilized parametric models in the case studies and the user studies were generated in GC using GC modelling and scripting functionalities.

6.2 Conclusion

ViSA, by combing the sensitivity analysis and visual analytics techniques, provides a logical means of interactively demonstrating the consequences of change on parametric design models. It represents an informative method, which supports designers in change-perception and insight gaining during investigating behaviour of the parametric design models. Utilizing ViSA can reduce designers’ need to use external tools for analyzing sensitivity of the design model. Our findings have also shown that ViSA has the potential to be useful in controlling and predicting the behaviour of complex parametric design model.

In addition, ViSA provides continuous feedback on the change-analysis cycle, which can enhance design cognition. It also provides an interactive means of visual searching that supports decision-making in finding design solutions. Additionally, the integrity of the model is maintained during applying and controlling change on the parametric design models. Through all of these features, ViSA supports designers to predict behaviour of the design model before committing to change it. The structure of ViSA consists of intuitive and straightforward steps in creating both control and visualization features, as well as building target model during preparation, visualization and analysis phases.
6.3 **Recommendations for Future Study**

While ViSA method facilitates focusing on the behaviour of design models during change control and “what-if” scenarios, there are some concerns regarding the creation of the target model in the preparation phase. These concerns are compounded by the increasing complexity of the design model. As a partial solution to cloning the reference model, a deep-copy feature, which requires application-level programming, can be developed. Most of the parametric CAD tools come with an application programming interfaces (APIs) that enables such extension. Further study is needed to find a better solution in cloning the reference model.

Another enhancement needed is improving the design of visualization and control features. As visual objects on design environment, they may be confusing or difficult to utilize.

It also seems difficult to implement ViSA in complex design models, especially complex geometries that need occasional scripting and different mode of thinking. With regard to this problem, more research is needed to create “re-useable” and “modular” ViSA elements, which expected to reduce the difficulty.

At this stage, we mainly focus on usability of ViSA, hence we did not study the solutions for visualization techniques or the procedure of cloning the reference model. We described the advantages, disadvantages and current limitation of ViSA in this thesis. However, more research and formal investigation needs to be carrying out to reveal the other aspects for improving the introduced ViSA method.
Appendices
APPENDICES

Appendix A

Pre-test survey: Please fill the pre-test questionnaire form and forward it to nhassanz@sfu.ca.

Participant Alias: Date:
Gender (optional): Age (optional):

Please rank your expertise in the following design domains and note the years of experience.

<table>
<thead>
<tr>
<th>Domain</th>
<th>None</th>
<th>Novice</th>
<th>Expert</th>
<th>Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architectural</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Product Design</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Engineering</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Graphics/Visual</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Software/IT</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Other(s)</td>
<td>☐</td>
<td>1 2 3 4</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

How do you evaluate yourself in parametric design modeling?

Novice 1 2 3 4 5 Expert

Please rank the relevance of the following areas in which you use parametric modeling tools.

<table>
<thead>
<tr>
<th>Area</th>
<th>Least Relevant</th>
<th>Most Relevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Research</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Teaching</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Designing</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Software development</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Other [_______________]</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

How do you evaluate your knowledge and skills in using the following parametric modeling tools?

<table>
<thead>
<tr>
<th>Tool</th>
<th>Low</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catia</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>SolidWorks</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>GenerativeComponents</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Revit</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
<tr>
<td>Other [_______________]</td>
<td>1 2 3 4 5</td>
<td></td>
</tr>
</tbody>
</table>

How familiar are you to Sensitivity-Analysis in general?

Not familiar 1 2 3 4 5 Highly familiar

Comments and special notes
Appendix B

Simon Fraser University

Form 2- Informed Consent by Participants in a Research Study

The university and those conducting this research study subscribe to the ethical conduct of research and to the protection at all times of the interests, comfort, and safety of participants. This research is being conducted under permission of the Simon Fraser Research Ethics Board. The chief concern of the Board is for the health, safety and psychological well-being of research participants.

Should you wish to obtain information about your rights as a participant in research, or about the responsibilities of researchers, or if you have any questions, concerns or complaints about the manner in which you were treated in this study, please contact the Director, Office of Research Ethics by email at hweinber@sfu.ca or phone at 778-782-6593.

Your signature on this form will signify that you have received a document which describes the procedures, whether there are possible risks, and benefits of this research study, that you have received an adequate opportunity to consider the information in the documents describing the study, and that you voluntarily agree to participate in the study.

Title: Visual Sensitivity-Analysis of Parametric Design Models: Improving Agility in Design

Investigator Name: Nahal H.Salmasi

Investigator Department: Faculty of Applied Science- School of Interactive Arts and Technology

Having been asked to participate in the research study named above, I certify that I have read the procedures specified in the Study Information Document describing the study. I understand the procedures to be used in this study and the personal risks to me in taking part in the study as described below:

Purpose and goals of this study: Developing a method as well as set of tools in parametric design modeling with the intention of augmenting designers’ cognition in predicting the effect of changes on design models without changing the design model itself.

What the participants will be required to do: This user study is in three parts. In Part-I participants are asked to work with 4 (four) parametric models at different levels of complexity built in GenerativeComponents\textsuperscript{11}. There are primarily two sets of four ordered tasks not ed on the handout. After completing the first set of tasks by applying their own method, they will be shown an alternative method and complete the second set of tasks following this method.

In Part-II participants are asked to replay and make observations on previously recorded transaction sequence of a parametric design model. The transaction captures the alternative method mentioned above that aims change-control and change-visualization for sensitivity analysis in which the effect of an input on the partial or entire model as output is of interest.

At the end of the study, participants will be asked to complete a short questionnaire in Part-III that intends to capture their evaluation of the modeling method shown in the second section of Part-I and Part-II.

\textsuperscript{11} Generative Components is a registered trademark of Bentley Systems, Incorporated.
Risks to the participant, third parties or society: There is no risk or harm to the participant, third parties or society.

Benefits of study to the development of new knowledge: This study will present a prototyping study to formalize the process of change control on parametric design models with a particular focus on geometric dependencies. This field of study is fairly new and if the proposed method complies successfully with the intention of change in participants, it will be very helpful for designers to support their decision during parametric design process and will have an enduring impact in this field of study.

Inclusion of names of participants in reports of the study: The participants’ comments and their names might be referred and included in my study to validate the proposed method in my research.

Contact of participants at a future time or use of the data in other studies: It will be addressed accordingly where their design model or their comments will be used. The contact details of participants will remain confidential.

I understand that I may withdraw my participation at any time. I also understand that I may register any complaint with the Director of the Office of Research Ethics.

Dr. Hal Weinberg  
Director, Office of Research Ethics  
Office of Research Ethics  
Simon Fraser University  
8888 University Drive  
Multi-Tenant Facility  
Burnaby, B.C. V5A 1S6  
hal_weinberg@sfu.ca

I may obtain copies of the results of this study, upon its completion by contacting:  
Nahal H. Salmasi  
nhassanz@sfu.ca  
Ph: 604 722 4678  
I understand the risks and contributions of my participation in this study and agree to participate:  
(The participant shall fill in this area. Please print legibly)

Participant Last Name:  
Participant First Name:  
Participant Contact Information:  
Participant Signature:  
Date (use format MM/DD/YYYY):
Appendix C

Visual sensitivity-analysis of parametric design models: Improving agility in design

– User Study –
Total Expected Duration: 60-80 minutes

This user study is in three parts. In Part-I you are asked to work with 4 (four) parametric models at different levels of complexity built in GenerativeComponents. There are primarily two sets of four ordered tasks noted on the handout. After completing the first set of tasks by applying your own method, you will be shown an alternative method and complete the second set of tasks following this method.

In Part-II you are asked to replay and make observations on previously recorded transaction sequence of a parametric design model. The transaction captures the alternative method mentioned above that aims change-control and change-visualization for sensitivity analysis in which the effect of an input on the partial or entire model as output is of interest.

At the end of the study, you will be asked to complete a short questionnaire in Part-III that intends to capture your evaluation of the modelling method shown in the second section of Part-I and Part-II. We wish to receive your feedback on the method demonstrated by the transaction sequence and its possible pros and cons for sensitivity analysis as well as in design.

During the course of this user study, you will be assisted by one of our group members. The assistant, while helping you, will record your interaction with the computer and your oral and written comments on written notes and video. Your identity and performance will be strictly confidential and in the publications and documentations following this study, you will be referred with the nickname you choose or given. You are free to take break or end the study when you wish.

We appreciate your help and contribution to this study.
Thank you,

Participant’s Name: ___________________________ Preferred Nickname:

Signature: ___________________________
Date: ___________________________

---

12 GenerativeComponents is a registered trademark of Bentley Systems, Incorporated.
Task 1: Find Line elongation-Angle Ratio

Model: The given line $L$ (Figure 1) is constructed by using $P_{01}$ and $P_{02}$.

Goals: We are interested in analyzing the length-change and $\alpha$-angle ratio and finding about the sensitivity of this ratio to the unit displacement\(^{13}\) of $P_{01}$.

Please demonstrate how you would achieve this goal and dynamically show the ratio as you change the model.

---

13 Unit displacement: A point can be moved anywhere in the range of [0,1] unit distance from its original location
Task 2: Displacement in a Bezier curve

Model: The Bezier curve $B$ (Figure 2) is constructed using 4 poles ($P_{01}$-$P_{04}$) as the control points. The curve is divided into 10 (ten) segments.

Goal: We are interested in the sensitivity of the displacement of these segment points to the unit displacement on $P_{02}$. Demonstrate how you would perform this analysis.

<table>
<thead>
<tr>
<th>Sketches:</th>
</tr>
</thead>
</table>

Figure 2: Bezier curve $B$ with 4 poles ($P_{01}$-$P_{04}$)

Comments:
Task 3: Torsion control in the Bezier curve

Model: A Bezier curve B (Figure 3) was created by using 4 poles (P₀₁-P₀₄). Assume this curve connects the centre of the cross sections of a rectangular beam along the beam length (Figure 4).

Goal: Changing the location of control points might cause torsion in beam (Figure 5). Demonstrate how we can move the control points P₀₂ and P₀₃ equal to one unit, while controlling the change and avoiding creating torsion on the curve.

<table>
<thead>
<tr>
<th>Comments:</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀₁</td>
</tr>
</tbody>
</table>

**Figures:**

- Figure 3: Centric Bezier curve B
- Figure 4: Beam and centric Curve without torsion
- Figure 5: Torsion in beam and its centric Curve
Task 4: Edge length control in paper folding example of place holder pattern

Model: The base surface $S$ for creating placeholders is created by 9 control points (Figure 6a). The placeholders are generated as polygons in nature, based on an array of point grids, considering these points as the vertices on the base surface (Figure 6b). By applying the paper-folding module (Figure 7) to the place holders the paper-folding example will be constructed (Figure 8).

Goal: Through changing the location of control points, different paper folding model will be created.

We are looking for a way to control the edge lengths of polygons; those are used in creating the paper-folding module on the surface $S$ (Figure 8). Please demonstrate the effect of unit displacement of control points $P_{02}$ and $P_{03}$ on edge lengths of the constructing polygons.

Use the back of paper for your sketches and comments.
Part-II: Elongation of the components of a meshed surface

Model: A B-Spline surface $S$ (Figure 9) was created by 6 control points ($P_{01}$-$P_{06}$) and a quadrangular meshed net was constructed on the surface using lines.

Goal: Demonstrating elongation of each component of the meshed net during unit displacement of control points $P_{03}$ and $P_{04}$

![Figure 9: Quadrangular-meshed surface](image)

Please follow the provided transactions step by step and describe your observations verbally, written, or by sketching.

Comments:
Questions and comments:

Please complete the survey by selecting your agreement with the statements and writing your comments about the method used in the second set of tasks in Part-I and demonstrated in Part-II.

The method enhanced my change-control on the models.

<table>
<thead>
<tr>
<th>I strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>I strongly agree</th>
</tr>
</thead>
</table>

Comments:

The method enhanced my perception of the effects of change on the models.

<table>
<thead>
<tr>
<th>I strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>I strongly agree</th>
</tr>
</thead>
</table>

Comments:

The method helped me gain insights about the models’ behavior and make-decision in the change process.

<table>
<thead>
<tr>
<th>I strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>I strongly agree</th>
</tr>
</thead>
</table>

Comments:

The method has potentials to be useful in controlling and predicting the behavior of complex design models.

<table>
<thead>
<tr>
<th>I strongly disagree</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>I strongly agree</th>
</tr>
</thead>
</table>

Comments:

Please list the three main advantages and disadvantages that you noticed with the method demonstrated in evaluating sensitivity of parametric design models.

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The method proposed for sensitivity analysis consists of three main steps: model preparation, change and change-visualization, analysis. Please give us your feedback on each of these phases if you have noticed them.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparation</td>
<td></td>
</tr>
<tr>
<td>Change and Change-Visualization</td>
<td></td>
</tr>
<tr>
<td>Analysis</td>
<td></td>
</tr>
</tbody>
</table>

The overall ease of using the method is…

| Extremely Difficult | 1 | 2 | 3 | 4 | 5 | Extremely Easy |

The likelihood of using proposed method in your future designs is…

| Most unlikely | 1 | 2 | 3 | 4 | 5 | Most likely |

What are your suggestions to improve the proposed method? Please write your comments on the back of this survey form or describe them to your assistant.
Appendix D

Utilized codes in user study models on GC version “08.11.07.66”

D.1 Elongation-Angle Ration on Line Model

```plaintext
transaction modelBased "Add line01, point01, point02"
{
  feature point02 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    HandlesVisible = true;
    XTranslation = <free> (8.38383446167285);
    YTranslation = <free> (5.00969182825485);
    ZTranslation = <free> (0.0);
  }
  feature point01 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    HandlesVisible = true;
    XTranslation = <free> (1.87395451482949);
    YTranslation = <free> (2.01614473684211);
    ZTranslation = <free> (0.0);
  }
  feature line01 Bentley.GC.Line
  {
    EndPoint = point02;
    StartPoint = point01;
    LineWeight = 2;
  }
  feature line02 Bentley.GC.Line
  {
    Direction = baseCS.YDirection;
    Length = 2;
    StartPoint = point02;
    Color = 1;
    LineWeight = 1.5;
  }
  transaction modelBased "Control feature"
  {
    feature delta01 Bentley.GC.GraphVariable
    {
      LimitValueToRange = true;
    }
  }
}
```
RangeMaximum  = 1.0;
Value         = 1;
}
feature circle01 Bentley.GC.Circle
{
  CenterPoint   = point01;
  Radius        = delta01;
  Support       = baseCS.XYPlane;
  Color         = 1;
  LineWeight    = 1.5;
}
feature point03 Bentley.GC.Point
{
  Curve        = circle01;
  HandlesVisible = true;
  T            = <free> (0.360177665999315);
}
}
transaction modelBased "PREPRATION: Add line03"
{
feature line03 Bentley.GC.Line
{
  EndPoint      = point02;
  StartPoint    = point03;
}
feature line03 Bentley.GC.Line
{
  Visible       = false;
}
transaction modelBased "VISUALIZATION TEXT: Add point04, point05, text"
{
feature point04 Bentley.GC.Point
{
  CoordinateSystem = onLine_baseCS;
  HandlesVisible   = true;
  XTranslation     = <free> (0.446117190721649);
  YTranslation     = <free> (0.570518675288612);
  ZTranslation     = <free> (0.0);
}
feature point05 Bentley.GC.Point
{

CoordinateSystem = onLine_baseCS;
HandlesVisible = true;
XTranslation = <free> (0.446117190721649);
YTranslation = <free> (0.312436874988159);
ZTranslation = <free> (0.0);
}

feature theTextStyle Bentley.GC.TextStyle
{
  Height = 0.05;
  Width = 0.05;
  ExtendedTextColor = 2;
  InterCharacterSpacing = 0.01;
}

feature text01 Bentley.GC.Text
{
  Placement = point04;
  TextString = "Ref.Ratio=" +Round((
    line01.Length)/Objects.Angle(point02, point01, line02.EndPoint),3);
  TextStyle = theTextStyle;
  LineWeight = 0.75;
}

feature text02 Bentley.GC.Text
{
  Placement = point05;
  TextString = "Target.Ratio=" +Round((
    line03.Length)/Objects.Angle(point02, point03, line02.EndPoint),3);
  TextStyle = theTextStyle;
  LineWeight = 0.75;
}

feature point05 Bentley.GC.Point
{
  YTranslation = <free> (1.69632132963988);
}

feature point04 Bentley.GC.Point
{
  Visible = false;
}

feature onLine_baseCS Bentley.GC.CoordinateSystem
D.2 Displacement Control on Bezier Curve

```plaintext
transaction modelBased "ControlPoints and Curve"
{
  feature bsplineCurve01 Bentley.GC.BSplineCurve
  {
    Poles = {point01, point02, point03, point04};
    Order = 4;
    LineWeight = 2.5;
  }
  feature point01 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    XTranslation = <free> (2.00012820512817);
    YTranslation = <free> (3.91743191196698);
    ZTranslation = <free> (0.0);
    HandlesVisible = true;
  }
  feature point02 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
  }
}```
APPENDICES

20 XTranslation = <free> (4.70393491124258);
21 YTranslation = <free> (6.97076891334251);
22 ZTranslation = <free> (0.0);
23 HandlesVisible = true;
24 }
25 feature point03 Bentley.GC.Point
26 {
27 CoordinateSystem = baseCS;
28 XTranslation = <free> (10.2276627218935);
29 YTranslation = <free> (2.93837276478679);
30 ZTranslation = <free> (0.0);
31 HandlesVisible = true;
32 }
33 feature point04 Bentley.GC.Point
34 {
35 CoordinateSystem = baseCS;
36 XTranslation = <free> (13.6115680473373);
37 YTranslation = <free> (7.25287070151307);
38 ZTranslation = <free> (0.0);
39 HandlesVisible = true;
40 }
41 transaction modelBased "Segments on Curve"
42 {
43 feature point05 Bentley.GC.Point
44 {
45 Curve = bsplineCurve01;
46 T = Series(0,1,0.1);
47 }
48 }
49 transaction modelBased "Control Feature"
50 {
51 feature delta02 Bentley.GC.GraphVariable
52 {
53 LimitValueToRange = true;
54 RangeMaximum = 1.0;
55 Value = 1;
56 }
57 feature circle01 Bentley.GC.Circle
58 {
59 CenterPoint = point02;
60 Radius = delta02;
61 Support = baseCS.XYPlane;
62 }
63 feature point06 Bentley.GC.Point
64 {
65 Curve = circle01;
66 HandlesVisible = true;
67 T = <free> (0.367486052959356);
68 }
69 transaction modelBased "PREPRATION: Add bsplineCurve02, line01, point07"
70 {
71 feature bsplineCurve02 Bentley.GC.BSplineCurve
72 {
Order                     = 4;
Poles                     = {point01,point06,point03,point04};
Visible                   = false;

feature point07 Bentley.GC.Point
{
  Curve                     = bsplineCurve02;
  T                         = Series(0,1,0.1);
}

feature line01 Bentley.GC.Line
{
  EndPoint                  = point07;
  StartPoint                = point05;
  Color                     = 10;
  LineWeight                = 2;
}

feature point07 Bentley.GC.Point
{
  Visible                   = false;
}

transaction modelBased "VISUALIZATION ON UNIT LINE: Add line02, line03, onaLine_baseCS, point08"
{
  feature onaLine_baseCS Bentley.GC.CoordinateSystem
  {
    ModelName                 = "OnaLine";
  }
  feature line02 Bentley.GC.Line
  {
    Direction                 = onaLine_baseCS.XDirection;
    Length                    = 1;
    StartPoint                = onaLine_baseCS;
    LineWeight                = 1.5;
  }
  feature point08 Bentley.GC.Point
  {
    Curve                     = line02;
    T                         = Series(0,1,0.1);
  }
  feature line03 Bentley.GC.Line
  {
    Direction                 = line01;
    Length                    = line01.Length;
    StartPoint                = point08;
    LineWeight                = 1.5;
    Color                     = 1;
  }
  feature onaLine_baseCS Bentley.GC.CoordinateSystem
  {
    Visible                   = false;
  }
  feature point08 Bentley.GC.Point
  {
    Visible                   = false;
  }
}
D.3 Torsion control in a B-Spline
1 transaction modelBased "Add bsplineCurve01, point01, point02, point03, point04"
2 {
3  feature point01 Bentley.GC.Point
4   {
5    CoordinateSystem = baseCS;
6    HandlesVisible = true;
7    X Translation = <free> (0.714286010048849);
8    Y Translation = <free> (7.4238833008139);
9    Z Translation = <free> (0.0);
10   }
11  feature point02 Bentley.GC.Point
12   {
13    CoordinateSystem = baseCS;
14    HandlesVisible = true;
15    X Translation = <free> (4.17149697277654);
feature point03 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (6.81615763756676);
  YTranslation = <free> (4.63538216807381);
  ZTranslation = <free> (0.0);
}

feature point04 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (7.08171950700013);
  YTranslation = <free> (-0.242033227874634);
  ZTranslation = <free> (0.0);
}

feature bsplineCurve01 Bentley.GC.BSplineCurve
{
  Order = 4;
  Poles = {point01,point02,point03,point04};
  LineWeight = 2;
}

transaction modelBased "Add coordinateSystem01"
{
  feature coordinateSystem01 Bentley.GC.CoordinateSystem
  {
    Curve = bsplineCurve01;
    T = Series(0,1,0.02);
  }
}

transaction modelBased "Control Feature"
{
  feature delta02 Bentley.GC.GraphVariable
  {
    LimitValueToRange = true;
    RangeMaximum = 1.0;
    Value = 1;
  }
  feature delta03 Bentley.GC.GraphVariable
  {
    Value = 1;
  }
  feature circle01 Bentley.GC.Circle
  {
    CenterPoint = point02;
    Radius = delta02;
    Support = baseCS.YZPlane;
    Color = 1;
    LineWeight = 1.5;
  }
  feature circle02 Bentley.GC.Circle
APPENDICES

feature point05 Bentley.GC.Point
{
  Curve = circle01;
  HandlesVisible = true;
  T = <free> (0.132513639185295);
}

feature point06 Bentley.GC.Point
{
  Curve = circle02;
  HandlesVisible = true;
  T = <free> (0.167726437201143);
}

transaction modelBased "PREPARATION: Add bsplineCurve02, coordinateSystem02"
{
  feature bsplineCurve02 Bentley.GC.BSplineCurve
  {
    Order = 4;
    Poles = {point01,point05,point06,point04};
    Visible = false;
  }

  feature coordinateSystem02 Bentley.GC.CoordinateSystem
  {
    Curve = bsplineCurve02;
    T = Series(0,1,0.02);
    Visible = false;
  }

  transaction modelBased "VISUALIZATION ON LINE Z DIRECTION: Add line01, line02, onLine_baseCS, point07"
  {
    feature onLine_baseCS Bentley.GC.CoordinateSystem
    {
      ModelName = "OnLine";
    }

    feature line01 Bentley.GC.Line
    {
      Direction = onLine_baseCS.XDirection;
      Length = 10;
      StartPoint = onLine_baseCS;
      LineWeight = 1.5;
    }

    feature point07 Bentley.GC.Point
    {
      Curve = line01;
      T = Series(0,1,0.02);
    }
  }
feature line02 Bentley.GC.Line
{
  Direction = coordinateSystem02.ZDirection;
  Length = 5;
  StartPoint = point07;
  Color = 10;
  LineWeight = 1;
}

feature onLine_baseCS Bentley.GC.CoordinateSystem
{
  Visible = false;
}

feature point07 Bentley.GC.Point
{
  Visible = false;
}

D.4 Edge-Length Control in Paper-Folding Model

transaction modelBased "Add bsplineSurface01, point01, point02, point03, point04, point05, point06, point07, point08, point09"
{
  feature point01 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    HandlesVisible = true;
    XTranslation = <free> (1.07921074684943);
    YTranslation = <free> (6.87015027700831);
    ZTranslation = <free> (0.0);
  }
  feature point02 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    HandlesVisible = true;
    XTranslation = <free> (3.66273881048892);
    YTranslation = <free> (5.46292728531856);
    ZTranslation = <free> (0.0);
  }
  feature point03 Bentley.GC.Point
  {
    CoordinateSystem = baseCS;
    HandlesVisible = true;
    XTranslation = <free> (8.43331290819946);
    YTranslation = <free> (7.07483725761773);
    ZTranslation = <free> (0.0);
  }

feature point04 Bentley.GC.Point {
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (0.912944089288474);
  YTranslation = <free> (3.08344113573407);
  ZTranslation = <free> (0.0);
}

feature point05 Bentley.GC.Point {
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (5.10797975698032);
  YTranslation = <free> (2.81478947368421);
  ZTranslation = <free> (0.0);
}

feature point06 Bentley.GC.Point {
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (9.92971282624808);
  YTranslation = <free> (2.4821731301939);
  ZTranslation = <free> (0.0);
}

feature point07 Bentley.GC.Point {
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (2.01286197776865);
  YTranslation = <free> (0.422510387811632);
  ZTranslation = <free> (0.0);
}

feature point08 Bentley.GC.Point {
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (4.90334387075145);
  YTranslation = <free> (0.0898940443213287);
  ZTranslation = <free> (0.0);
}

feature point09 Bentley.GC.Point {

CoordinateSystem = baseCS;
HandlesVisible = true;
XTranslation = <free> (8.88095390932512);
YTranslation = <free> (0.0387222991689734);
ZTranslation = <free> (0.0);

feature bsplineSurface01 Bentley.GC.BSplineSurface
{
Poles =
    {{point01,point02,point03},{point04,point05,point06},{point07,point08,point09}};
UOrder = 4;
}
transaction modelBased "Add point10, polygon01"
{
feature point10 Bentley.GC.Point
{
Replication = ReplicationOption.AllCombinations;
Surface = bsplineSurface01;
U = Series(0,1,0.25);
V = Series(0,1,0.25);
}
feature polygon01 Bentley.GC.Polygon
{
Points = point10;
}
feature bsplineSurface01 Bentley.GC.BSplineSurface
{
Visible = false;
}
transaction modelBased "Centric Point"
{
feature CentPt Bentley.GC.Point
{
SetToFindCentroidFrom =
    {point10[0][0],point10[5][0],point10[0][5],point10[5][5]};
SymbolXY = {99, 103};
}
}
APPENDICES

108 transaction modelBased "Determine up direction"
109 {
110 feature directionA Bentley.GC.Direction
111 {
112    Origin = CentPt;
113    DirectionPoint = point10[5][5];
114    SymbolXY = {99, 105};
115 }
116 feature directionB Bentley.GC.Direction
117 {
118    Origin = CentPt;
119    DirectionPoint = point10[5][0];
120    SymbolXY = {101, 105};
121 }
122 feature directionUp Bentley.GC.Direction
123 {
124    Origin = CentPt;
125    XDirection = directionA;
126    YDirection = directionB;
127    SymbolXY = {100, 106};
128 }
129 }
130 transaction modelBased "Height to control"
131 {
132 feature height Bentley.GC.GraphVariable
133 {
134    Value = 1;
135    LimitValueToRange = true;
136    RangeMinimum = 0;
137    RangeMaximum = 8.0;
138    RangeStepSize = 0.25;
139    SymbolXY = {102, 101};
140 }
141 }
142 transaction modelBased "Add plHo_PaperFolding01"
143 {
144 feature plHo_PaperFolding01 User.PlHo_PaperFolding
145 {
146    baseShape = polygon01;
147    height = height;
148    upDirection = directionUp;
149    LineWeight = 1.5;
transaction modelBased "Hide directionA, directionB, directionUp, point10, polygon01"
{
  feature point10 Bentley.GC.Point
  {
    Visible = false;
  }
  feature polygon01 Bentley.GC.Polygon
  {
    Visible = false;
  }
  feature directionA Bentley.GC.Direction
  {
    Visible = false;
  }
  feature directionB Bentley.GC.Direction
  {
    Visible = false;
  }
  feature directionUp Bentley.GC.Direction
  {
    Visible = false;
  }
}

transaction modelBased "Add delta04, deltaRadiusSosphere"
{
  feature delta04 Bentley.GC.GraphVariable
  {
    LimitValueToRange = true;
    RangeMaximum = 1.0;
    Value = 1;
  }
  feature deltaRadiusSosphere Bentley.GC.GraphVariable
  {
    LimitValueToRange = true;
    RangeMaximum = 1.0;
    Value = 1;
  }
}

transaction modelBased "CONTROL FEATURE I: Add circle01, point11"
feature circle01 Bentley.GC.Circle
  {  
    CenterPoint               = point04;
    Radius                    = delta04;
    Support                   = baseCS.XZPlane;
    Color                     = 1;
    LineWeight                = 1.5;
  }
feature point11 Bentley.GC.Point
  {  
    Curve                     = circle01;
    HandlesVisible            = true;
    T                         = <free> (0.558658400745228);
  }
}
transaction modelBased "CONTROL FEATURE II: Sophere"
{  
  feature controllerOrigin Bentley.GC.CoordinateSystem
  {  
    CoordinateSystem          = baseCS;
    XTranslation              = <free> (-5);
    YTranslation              = <free> (-5);
    ZTranslation              = <free> (-5);
    SymbolXY                  = {101, 100};
  }
  feature azimuthCircle Bentley.GC.Circle
  {  
    CenterPoint               = controllerOrigin;
    Radius                    = deltaRadiusSophere;
    Support                   = controllerOrigin.XYPlane;
    Visible                   = true;
    SymbolXY                  = {102, 101};
    LineWeight                = 2;
  }
  feature azimuthPlane Bentley.GC.Plane
  {  
    Curve                     = azimuthCircle;
    T                         = <free> (0.129788051241254);
    Visible                   = true;
    HandlesVisible            = true;
    SymbolXY                  = {102, 102};
  }
feature altitudeCircle Bentley.GC.Circle
{  
  CenterPoint = azimuthCircle.CenterPoint;
  Radius = deltaRadiusSphere;
  Support = azimuthPlane;
  Visible = true;
  SymbolXY = {103, 103};
  LineWeight = 2;
}

feature altitudePoint Bentley.GC.Point
{  
  Curve = altitudeCircle;
  T = <free> (0.333947465521347);
  HandlesVisible = true;
  SymbolXY = {103, 104};
}

feature controlDirection Bentley.GC.Direction
{  
  Origin = altitudeCircle.CenterPoint;
  DirectionPoint = altitudePoint;
  SymbolXY = {104, 105};
}

feature controlLine Bentley.GC.Line
{  
  StartPoint = altitudeCircle.CenterPoint;
  Direction = controlDirection;
  Length = deltaRadiusSphere*2.0;
  Color = 3;
  FillColor = -1;
  LevelName = "Default";
  LineStyleName = "0";
  LineWeight = 2;
  SymbolXY = {105, 106};
}

feature controllerOrigin Bentley.GC.CoordinateSystem
{  
  XTranslation = <free> (-0.0134017532316149);
  ZTranslation = <free> (2.03783842704258);
}

feature altitudePoint Bentley.GC.Point
{
feature altitude Bentley.GC.GraphVariable
{
  Value = Objects.Angle(altitudeCircle.CenterPoint, azimuthPlane, altitudePoint);
}

feature azimuth Bentley.GC.GraphVariable
{
  Value = Objects.Angle(azimuthCircle.CenterPoint, azimuthZero, azimuthPlane);
}

feature azimuthZero Bentley.GC.Point
{
  Curve = azimuthCircle;
  T = 0.0;
  HandlesVisible = true;
  SymbolXY = {101, 102};
}

feature lineControlTarget Bentley.GC.Line
{
  StartPoint = point05;
  Direction = controlLine;
  Length = 2;
  Color = 10;
  LineWeight = 2;
}

transaction modelBased "PREPRATION: creating target model"
{
  feature bsplineSurface02 Bentley.GC.BSplineSurface
  {
    Poles = 
      {{point01,point02,point03},{point11,lineControlTarget.EndPoint,point06},{point07,point08,point09}};
    UOrder = 4;
    Visible = false;
  }
  feature point12 Bentley.GC.Point
  {

Replication               = ReplicationOption.AllCombinations;
Surface                   = bsplineSurface02;
U                         = Series(0,1,0.25);
V                         = Series(0,1,0.25);
}
feature polygon02 Bentley.GC.Polygon
{
    Points                    = point12;
}
feature CentPt02 Bentley.GC.Point
{
    SetToFindCentroidFrom     =
        {point12[0][0],point12[5][0],point12[0][5],point12[5][5]};
}
feature directionA2 Bentley.GC.Direction
{
    Origin                    = CentPt02;
    DirectionPoint            = point12[5][5];
    SymbolXY                  = {99, 105};
}
feature directionB2 Bentley.GC.Direction
{
    Origin                    = CentPt02;
    DirectionPoint            = point12[5][0];
    SymbolXY                  = {101, 105};
}
feature directionUp2 Bentley.GC.Direction
{
    Origin                    = CentPt;
    XDirection                = directionA2;
    YDirection                = directionB2;
}
feature plHo_PaperFolding02 User.PlHo_PaperFolding
{
    baseShape                 = polygon02;
    height                    = height;
    upDirection               = directionUp2;
    Color                     = 7;
    LineWeight                = 1.5;
    Visible                   = false;
}
feature point12 Bentley.GC.Point
APPENDICES

354  {
355   Visible = false;
356  }
357 feature polygon02 Bentley.GC.Polygon
358  {
359   Visible = false;
360  }
361 }
362 transaction modelBased "PREPARING FOR VISUALIZATION"
363 {
364 feature numOfIntervals Bentley.GC.GraphVariable
365  {
366   LimitValueToRange = true;
367   RangeMaximum = 10.0;
368   RangeMinimum = 1.0;
369   RangeStepSize = 1.0;
370   Value = 7.0;
371  }
372 feature visStatistics_baseCS Bentley.GC.CoordinateSystem
373  {
374   ModelName = "visStatistics";
375  }
376 }
377 transaction modelBased "PREPARING FOR VISUALIZATION: GRAPH FUNCTIONS"
378 {
379 feature LengthsToArray Bentley.GC.GraphFunction
380  {
381   Definition = double [] function (PlHo_PaperFolding mPoly)
382    {
383     PlHo_PaperFolding [] placeHolders = Flatten(mPoly);
384     Polygon [] allPolygons = {};
385     Object [] edges = {};
386      foreach (PlHo_PaperFolding pH in placeHolders)
387       {
388       edges = ExtractEdgesFromPlaceHolder (pH, edges);
389       }
390 double [] lengths = {};
391      foreach (Object[] o in edges) {
392    lengths = Add ( lengths, Distance(o[0], o[1]));
393       }
394 double [] Statistics = {};
395    Statistics = GetStatisticsInArray (lengths);
Print (Statistics); // for testing
DisplayStatistics (Statistics, lengths); //Add
}

feature DisplayStatistics Bentley.GC.GraphFunction
{
    Definition = function (double [] Statistics, double []
        dataSeries)
    /** order: min -> 0, max ->1, mean->2, standard deviation->3 */
    {
        Point p_min = new Point ("minimum_of_model");
p_min.ByCoordinateList (visStatistics_baseCS, {Statistics[0],0,0} ,
            visStatistics_baseCS.Origin);
p_min point p_max = new Point ("maximum_of_model");
p_max.ByCoordinateList (visStatistics_baseCS, { Statistics[1],/*max*/
            ,0,0} , p_min);
        Line BaseLine= new Line ("base");
        BaseLine.ByPoints(p_min , p_max );
        Point p_interval = new Point ("p_interval");
p_interval.ByNumberAlongCurve(BaseLine, numOfIntervals+1);
        DrawStackHistogram (dataSeries,Statistics,p_interval);
        DrawStackHistogram.Color =1;
        Point p_mean = new Point ("mean_of_reference_model");
p_mean.ByCoordinateList (visStatistics_baseCS, {Statistics[2],0,0} ,
            visStatistics_baseCS.Origin);
        Line MeanLine= new Line ("mean");
        MeanLine.ByStartPointDirectionLength(p_mean,visStatistics_baseCS.YDirecti
            on,10);
        MeanLine.Color = 6;
        MeanLine.LineStyle=4;
    }
}

feature DrawStackHistogram Bentley.GC.GraphFunction
{
    Definition = function (double [] lengths, double
        [] Statistics, Point p)
    /** order: min, max, mean, standard deviation */
    {
        int [] stackCount = FilledList (numOfIntervals, 0);
        int index = 0;
double d = (Statistics[1]-Statistics[0])/(numOfIntervals);
        foreach (double v in lengths)
{432} stackCount [ Floor ((v - Statistics[0])/d) ]++;
{434} }
{435} for (int i=0; i < stackCount.Count; i++)
{436} {
{437} Line BaseLine= new Line ("l"+i) ;
{438} BaseLine.ByStartPointDirectionLength(p[i],
   //add 1/10 as lines were to long
   visStatistics_baseCS.YDirection, 1/10*stackCount[i]);
{439} BaseLine-LineWeight=7;
{440} }
{441} ;
{442} }
443 feature ExtractEdgesFromPlaceHolder Bentley.GC.GraphFunction
{444} {
{445} Definition = Object [] function (PlHo_PaperFolding pHolder, Object [] es) {//PaperFolding as PlaceHolder
{446}   Polygon [] poly = {pHolder.foldShapeA,
{447}   pHolder.foldShapeB,
{448}   pHolder.foldShapeC,
{449}   pHolder.foldShapeD,
{450}   pHolder.foldShapeE,
{451}   pHolder.foldShapeF};
{452}   for (int i=0; i<poly.Count; i++) {
{453}     es = ExtractEdgesFromPolygon(poly[i], es);
{454}   }
{455}   return es;
{456} }
457 }
458 }
459 feature ExtractEdgesFromPolygon Bentley.GC.GraphFunction
{460} {
{461}   Definition = Object [] function (Polygon p, Object [] e)
{462}   {
{463}     Point [] v = p.Vertices;
{464}     for (int i=0; i < v.Count-1; i++)
{465}     {
{466}       Point [] newEdge = {v[i], v[i+1]};
{467}       if ( !IsDublicatedEdge (e, newEdge))
{468}       {
{469}         e = Add(e, newEdge);
{470}       }
471 Object [] newLastEdge = {v[v.Count-1], v[0]};
472 if ( !IsDublicatedEdge (e, newLastEdge))
473     {
474         e = Add(e, newLastEdge);
475     }
476 return e;
477 }
478 }
479 }
480 feature IsDublicatedEdge Bentley.GC.GraphFunction
481 {
482     Definition = boolean function (Object [] edges, Point [] newEdge)
483     {
484         /**We need to figure out why object comparision does return as invalid
485          operation below is a workaround*/
486         for (int j=0; j<edges.Count; j++)
487         {
488             Object [] toCompare = edges[j];
489             if ( AreSamePoints (newEdge[0], toCompare[0]) || AreSamePoints
490                 (newEdge[0],toCompare[1])
491                 {
492                 if ( AreSamePoints (newEdge[1], toCompare[0]) || AreSamePoints
493                     (newEdge[1], toCompare[1]))
494                 {
495                     return true;
496                 }
497             }
498             return false;
499         }
500     }
501 }
502 feature AreSamePoints Bentley.GC.GraphFunction
503 {
504     Definition = boolean function (Point a, Point b) {
506     }
507 }
508 feature GetStatisticsInArray Bentley.GC.GraphFunction
509 {

Definition = double function (double [] dataArray )
{
    /** order: min, max, mean, standard deviation */
    double [] statistics = { };
    statistics = Add( statistics, GetMinimum (dataArray));
    statistics = Add( statistics, GetMaximum (dataArray));
    statistics = Add( statistics, CalculateMean (dataArray));
    statistics = Add( statistics, CalculateStandardDeviation (dataArray));
    return statistics;
}

feature CalculateMean Bentley.GC.GraphFunction
{
    Definition = double function (double [] dataArray )
    {
        /**Returns populations Mean*/
        double total = 0.0;
        foreach (double l in dataArray)
        {
            total += l;
        }
        return total/dataArray.Count;
    }
}

feature CalculateStandardDeviation Bentley.GC.GraphFunction
{
    Definition = double function (double [] dataArray )
    {
        /**Returns populations standard deviation*/
        double mean = CalculateMean (dataArray);
        double squaredSum = 0;
        foreach (double l in dataArray)
        {
            squaredSum += Pow( (l-mean), 2);
        }
        return Sqrt(squaredSum/dataArray.Count);
    }
}

feature GetMinimum Bentley.GC.GraphFunction
{
    Definition = double function (double [] dataArray )
    {

APPENDICES

552        /**<Returns min */
553double min = dataArray[0];
554foreach (double l in dataArray)
555 {
556 if (min > l) min = l;
557 }
558return min;
559 );
560 } //GetMinimum
561feature GetMaximum Bentley.GC.GraphFunction
562 {
563     Definition = double function (double [] dataArray )
564     {
565         /**<Returns max*/
566double max = dataArray[0];
567foreach (double l in dataArray)
568 {
569 if (max < l)
570 {
571 max = l;
572 }
573 }
574return max;
575 }
576 } //GetMaximum
577 }
578transaction modelBased "VISUALIZATION: Bar charts Statistics"
579 {
580feature text10 Bentley.GC.Text
581 {
582     Function = LengthsToArray;
583     FunctionArguments = {plHo_PaperFolding01};
584 }
585feature theTextStyle Bentley.GC.TextStyle
586 {
587     Height = 0.15;
588     Width = 0.15;
589     ExtendedTextColor = 2;
590     InterCharacterSpacing = 0.04;
591 }
592feature text12 Bentley.GC.Text
593 {
Placement = Cpoint1;
TextString = "MinLength";
TextStyle = theTextStyle;
}

feature text13 Bentley.GC.Text
{
Placement = Cpoint2;
TextString = "MaxLength";
TextStyle = theTextStyle;
}

feature text14 Bentley.GC.Text
{
Placement = Cpoint3;
TextString = "StackCount";
TextStyle = theTextStyle;
}

feature text15 Bentley.GC.Text
{
Placement = mean.EndPoint;
TextString = "MeanLine";
TextStyle = theTextStyle;
}

feature Cpoint1 Bentley.GC.Point
{
CoordinateSystem = visStatistics_baseCS;
HandlesVisible = true;
XTranslation = minimum_of_model.X-0.2;
YTranslation = minimum_of_model.Y-0.4;
ZTranslation = <free> (0.0);
}

feature Cpoint2 Bentley.GC.Point
{
CoordinateSystem = visStatistics_baseCS;
HandlesVisible = true;
XTranslation = maximum_of_model.X+0.1;
YTranslation = maximum_of_model.Y-0.4;
ZTranslation = <free> (0.0);
}

feature Cpoint3 Bentley.GC.Point
{
CoordinateSystem = visStatistics_baseCS;
HandlesVisible = true;
transaction modelBased "Hide Constructions: Cpoint1, Cpoint2, Cpoint3, mean_of_reference_model, p_interval, visStatistics_baseCS"

feature visStatistics_baseCS Bentley.GC.CoordinateSystem

feature Cpoint3 Bentley.GC.Point

feature mean_of_reference_model Bentley.GC.Point

feature Cpoint1 Bentley.GC.Point

feature Cpoint2 Bentley.GC.Point

feature p_interval Bentley.GC.Point

D.5 Quadrangular-Meshed Surface

transaction modelBased "Add bsplineSurface01, point01, point02, point03, point04, point05, point06"

feature point01 Bentley.GC.Point

CoordinateSystem = baseCS;
HandlesVisible = true;
XTranslation = <free> (3.37440032911713);
YTranslation = <free> (7.97779798878355);
ZTranslation = <free> (0.0);

feature point02 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (7.67187951710635);
  YTranslation = <free> (7.74662252938817);
  ZTranslation = <free> (0.0);
}

feature point03 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (1.75005551699026);
  YTranslation = <free> (4.36904220929333);
  ZTranslation = <free> (0.0);
}

feature point04 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (8.84857024715101);
  YTranslation = <free> (4.17713423656067);
  ZTranslation = <free> (0.0);
}

feature point05 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (2.7860549640948);
  YTranslation = <free> (0.479707295244723);
  ZTranslation = <free> (0.0);
}

feature point06 Bentley.GC.Point
{
  CoordinateSystem = baseCS;
  HandlesVisible = true;
  XTranslation = <free> (8.17069406571224);
  YTranslation = <free> (0.198242268570153);
ZTranslation = <free> (0.0);

feature bsplineSurface01 Bentley.GC.BSplineSurface
{
  Poles =
    {{point01,point02},{point03,point04},{point05,point06}};
}

transaction modelBased "Add point07"
{
  feature point07 Bentley.GC.Point
  {
    Replication = ReplicationOption.AllCombinations;
    Surface = bsplineSurface01;
    U = Series(0,1,0.2);
    V = Series(0,1,0.2);
  }
}

transaction modelBased "MESH BY LINES: Add line01, line02; change bsplineSurface01"
{
  feature bsplineSurface01 Bentley.GC.BSplineSurface
  {
    Visible = false;
  }
  feature line01 Bentley.GC.Line
  {
    LacingMethod = LacingOption.DirectMapping;
    Points = point07;
    LineWeight = 2;
  }
  feature line02 Bentley.GC.Line
  {
    Vertices = point07;
    LineWeight = 2;
  }
}

transaction modelBased "Control Feature"
{
  feature deltaP03 Bentley.GC.GraphVariable
  {
  }
}
 LimitValueToRange = true;
 RangeMaximum = 1.0;
 RangeStepSize = 0.001;
 Value = 1;
 }

 feature delta04 Bentley.GC.GraphVariable
 {
  LimitValueToRange = true;
  RangeMaximum = 1.0;
  RangeStepSize = 0.001;
  Value = 1.0;
 }

 feature delta06 Bentley.GC.GraphVariable
 {
  LimitValueToRange = true;
  RangeMaximum = 1.0;
  RangeStepSize = 0.001;
  Value = 1.0;
 }

 feature circle01 Bentley.GC.Circle
 {
  CenterPoint = point03;
  Radius = deltaP03;
  Support = baseCS.XYPlane;
  Color = 216;
  LineWeight = 2;
 }

 feature circle02 Bentley.GC.Circle
 {
  CenterPoint = point04;
  Radius = delta04;
  Support = baseCS.XYPlane;
  Color = 216;
  LineWeight = 2;
 }

 feature circle03 Bentley.GC.Circle
 {
  CenterPoint = point06;
  Radius = delta06;
  Support = baseCS.XYPlane;
  Color = 216;
  LineWeight = 2;
feature point08 Bentley.GC.Point
{
  Curve = circle01;
  HandlesVisible = true;
  T = <free> (0.359300152871895);
}

feature point09 Bentley.GC.Point
{
  Curve = circle02;
  HandlesVisible = true;
  T = <free> (0.172062749717233);
}

feature point09a Bentley.GC.Point
{
  Curve = circle03;
  HandlesVisible = true;
  T = <free> (0.172062749717233);
}

transaction modelBased "PREPRATION: Add bsplineSurface02"
{
  feature bsplineSurface02 Bentley.GC.BSplineSurface
  {
    Poles = {{point01,point02},{point08,point09},{point05,point09a}};
  }

  feature point10 Bentley.GC.Point
  {
    Replication = ReplicationOption.AllCombinations;
    Surface = bsplineSurface02;
    U = Series(0,1,0.2);
    V = Series(0,1,0.2);
  }

  feature line03 Bentley.GC.Line
  {
    LacingMethod = LacingOption.DirectMapping;
    Points = point10;
    Color = 176;
    RoleInExampleGraph = null;
    SymbolicModelDisplay = null;
  }
}
feature line04 Bentley.GC.Line

  Vertices = point10;
  Color = 176;
  RoleInExampleGraph = null;
  SymbolicModelDisplay = null;
 }
 }

transaction modelBased "VISUALIZATION: Add point11, point12"

feature point11 Bentley.GC.Point

  Direction = line01;
  DistanceFromOrigin = line01.Length/2;
  Origin = point07;
 }

feature point12 Bentley.GC.Point

  Direction = line02;
  DistanceFromOrigin = line02.Length/2;
  Origin = point07;
 }

feature circle04 Bentley.GC.Circle

  CenterPoint = point11;
  Radius = (line03.Length-line01.Length);
  Support = baseCS.XYPlane;
  LineWeight = 2;
 }

feature circle05 Bentley.GC.Circle

  CenterPoint = point12;
  Radius = (line04.Length-line02.Length);
  Support = baseCS.XYPlane;
  LineWeight = 2;
 }

feature circle04 Bentley.GC.Circle

  Color = line03.Length> line01.Length? 10:1;
 }

feature circle05 Bentley.GC.Circle


Color = line04.Length > line02.Length? 10:1;
REFERENCE LIST


Bates-Brkljac, N. (2007) 'Investigating perceptual responses and shared understanding of architectural design ideas when communicated through different forms of visual representations', Proceedings of The 11th International Conference Information Visualization, Zurich, Switzerland, Institute of Electrical and Electronics Engineers Inc. United States: 348-353.


REFERENCE LIST


