

UNIVERSITY OF CALGARY

Requirements and Reuse in an Agile Environment for Tabletop-based GIS Applications

by

Elaf Sameh Selim

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

APRIL, 2012

© Elaf Selim 2012

Abstract

This work studies an approach for gathering requirements, designing and developing an interactive collaborative multi-touch tabletop Geographic Information System (GIS) in an agile software development environment to support the operation of control centers in utility companies and potentially other similar domains. The research is based on two case studies. The first study focuses on agile approaches for gathering requirements and iteratively designing gesture-based, multi-touch GIS applications utilizing digital tables, where a prototype – eGrid – is developed for supporting the operation of the control centre in a local electricity company. The design is iteratively enhanced and assessed through interviews and observation sessions. The second study proposes an agile approach for generalizing the design and extracting a library of reusable components – GISforTT – to support the development of other interactive GIS applications. The two studies proposed agile software engineering practises that were found helpful in designing and supporting reuse of interactive tabletop applications.

“In the Name of Allah, the Most Merciful, the Most Compassionate”

Acknowledgements

First and above all, all praise and thanks go to Allah for teaching me, nurturing me and helping me go through this journey and for giving me the strength and patience to carry all the competing responsibilities in my life.

Special thanks go to Dr. Frank Maurer for his continuous support, for teaching me, motivating me and inspiring me throughout my research and for being so patient and flexible. I was really blessed to have you as my supervisor.

Special thanks also go to TRILabs for supporting my thesis and especially to Rainer Iraschko for being so helpful and supportive. Many Thanks also go to Natural Sciences and Engineering Research Council of Canada (NSERC) foundation for supporting my industrial partnership with TRILabs and the collaborating power corporation.

For all my friends and fellow researchers in the Agile Software Engineering Lab in the University of Calgary, thank you so much for being such good friends and for helping me, participating in my studies and supporting my research. Special thanks to Michael Chen and Patrick King for helping in the implementation of the application and to Dr. Yaser Ghanam for inspiring me and giving me very valuable feedback on my research.

Dedication

I dedicate this work to my loving and caring parents Sameh and Kariman for always encouraging me to work hard and for always having very high expectations from me which makes me aim for excellence in all aspects of my life. I love you so much and I owe my success to your prayers and continuous support.

I dedicate this work to my husband Mohamed and my son Yehia for being so kind and for putting a smile on my face when life is so stressful and crazy. I love you both so much and may God bless you both.

Publications

Portions of the materials and ideas presented in this thesis may have appeared previously in the following peer reviewed publications:

- Selim, E. and Maurer, F. (2010) eGrid: Supporting the Control Room Operation of a Utility Company with Multi-Touch Tables. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Saarbrucken, Germany, November 7-10, 2010). ITS'10, ACM, New York, NY.
- Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and Extreme Programming (XP 2011), Madrid, Spain, 2011.

Table of Contents

Abstract.....	i
Acknowledgements.....	iii
Dedication.....	iv
Publications.....	v
Table of Contents.....	vi
List of Tables.....	ix
List of Figures and Illustrations.....	x
CHAPTER ONE: INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Problem of Interest.....	3
1.3 Research Goals.....	4
1.4 Methodology.....	5
1.4.1 Case Study 1: Requirement Gathering, Assessment and Prototype Design.....	5
1.4.2 Case Study 2: Reusable Library Extraction and User-Feedback-based Enhancement.....	7
1.5 Thesis Structure.....	8
CHAPTER TWO: RELATED WORK.....	9
2.1 Interactive GIS Applications.....	9
2.1.1 Virtual Globes and Spatial Thinking.....	10
2.1.2 Environments for Geocollaboration.....	13
2.1.3 Interactive Walls and Access Rights.....	14
2.1.4 Combining Speech and Gesture Recognition.....	15
2.1.5 Combining Hand and Foot Gestures.....	22
2.1.6 GeoSpatial Information on Touch Tables.....	23
2.2 Tabletop Environments and Gesture Design.....	26
2.3 Reuse and Extracting Reusable Assets.....	34
2.3.1 Overview.....	34
2.3.2 Relevant Reuse Approaches.....	36
CHAPTER THREE: CASE STUDY 1 – EGRID.....	42
3.1 Application Domain.....	42
3.2 Approaches for Requirement Gathering and Design.....	47
3.2.1 Initial User Requirements.....	48
3.2.2 Low Fidelity Sketches.....	48
3.2.3 Interactive Mock Demos.....	53
3.2.4 Iterations of Advanced Prototypes.....	56
3.2.5 Observation Sessions and Interviews.....	62
3.2.5.1 Overview and Study Setup.....	63

3.2.5.2 Study Results	67
3.3 Technologies Used in eGrid	77
3.4 Design Details	80
3.4.1 User Interface Elements	82
3.4.1.1 Background Map.....	82
3.4.1.2 Corner Menus	82
3.4.1.3 Map Windows.....	83
3.4.1.4 Window Shadows	87
3.4.1.5 Annotation Tools	89
3.4.1.6 Map Window Layers List	91
3.4.1.7 Environment Configuration Manager	92
3.4.1.8 Trouble Report Pins	94
3.4.1.9 Trouble Report Info Boxes	95
3.4.1.10 Trouble Report List Windows	97
3.4.1.11 Service Crews Layer.....	97
3.4.2 Interactions and Multi-touch support	98
3.4.2.1 Creating Map Windows	98
3.4.2.2 Manipulating the Map Window	99
3.4.2.3 Panning and Zooming the Map.....	101
3.4.2.4 User Feedback Regarding Gesture Interactions.....	102
3.4.3 Enhancement Ideas	102
3.4.3.1 Displaying Crew Information	102
3.4.3.2 Creating and Editing Features.....	103
3.4.3.3 Searching Through Annotations	103
3.4.3.4 Searching for Equipment by Type or ID	104
3.4.3.5 Spatial Analysis	104
3.4.3.6 Textual Input Device	105
3.5 Summary.....	105
CHAPTER FOUR: CASE STUDY 2 – GISFORTT	107
4.1 Background and Motivation	107
4.2 Sky Hunter and Reuse Analysis.....	110
4.3 Extraction Approach	114
4.4 Extraction Example from the Case Study.....	119
4.4.1 Writing Acceptance Tests for Sky Hunter.....	120
4.4.2 Comparing the User Stories of eGrid and Sky Hunter	122
4.4.3 Analyzing the Acceptance Tests of eGrid and Sky Hunter	124
4.4.4 Refactoring the Map Window Based on the Analysis.....	125
4.4.5 Moving the Map Window to the Library	125
4.4.6 Refactoring eGrid Classes that Interact with the Map Window	126
4.4.7 Using the Extracted Map Window in Sky Hunter	126
4.5 Description of the Resulting Library	127

4.6 Enhancing the Library by Gathering User Feedback.....	128
4.6.1 Study Design	129
4.6.1.1 Participants and Recruitment	129
4.6.1.2 Study Tasks.....	129
4.6.1.3 Questionnaire Content	132
4.6.2 Responses of Participants and Enhancements	134
4.6.3 Limitations.....	138
4.7 Summary.....	139
CHAPTER FIVE: CONCLUSION.....	142
5.1 Summary of Results.....	142
5.2 Thesis Contribution.....	143
5.3 Limitations and Future Work.....	143
REFERENCES	145

List of Tables

Table 3-1: Specializations of Interview Participants	64
---	-----------

List of Figures and Illustrations

Figure 2-1: Snapshot for interacting with spatial data from Schöning et al research [Schöning2008b].....	11
Figure 2-2: Snapshot for the LIFE-SAVER prototype for emergency spatial data visualization by Nóbrega et al [Nóbrega2008].	13
Figure 2-3: Snapshot showing DAVE_G used by two users interacting with GIS data using pointing gestures [Rauschert2002a].	17
Figure 2-4: A snapshot of GeoMIP where the user can interact with spatial data using speech and gestures [Agrawal2004].	19
Figure 2-5: A snapshot from the research by Benko et al, showing the devices the user can use to interact with the data [Benko2004].	20
Figure 2-6: Two snapshots showing users interacting with visualized 3D terrain data by Benko et al [Benko2003].	21
Figure 2-7: Snapshots from DTMap where users are using MagicLens to display different views of the data [Furuichi2005].	25
Figure 2-8: A list of the tasks presented to the users in the study by Wobbrock et al.....	28
Figure 2-9: Examples for different information spaces that can be navigated using the concept of a PaperLens by Spindler et al [Spindler2009].....	32
Figure 2-10: Snapshots from the research by Hesselmann et al representing half-pie menus and how users would rotate the menu to reveal more buttons on each layer [Hesselmann2009].	33
Figure 3-1: A diagram representing the current setup of the environment in the control center of the collaborating utility company.	44
Figure 3-2: First sketch showing a small map window, a trouble report list and four corner menus, one of them is expanded to show multiple tools.	50
Figure 3-3: Second example sketch that represents two individual map windows with toolbox icons on the frame.....	51
Figure 3-4: Third example sketch describing the color wheel concept, which is used to allow the user to select the color of the annotation pen.	52

Figure 3-5: Fourth example sketch used to show the annotations that are typically added by the operators of the control center on the maps.....	53
Figure 3-6: A screenshot from the first ArcGIS interactive demo of eGrid.	55
Figure 3-7: Another screenshot from the first ArcGIS interactive demo of eGrid.	55
Figure 3-8: Snapshot from an early version of eGrid featuring trouble report pins and the Calgary map background.	57
Figure 3-9: Another snapshot of the same version of eGrid featuring different map layers in different frames and annotations added on the maps.	58
Figure 3-10: eGrid on the Microsoft Surface Table.....	77
Figure 3-11: eGrid on the Evoluce Table.	80
Figure 3-12: A snapshot for DTMap showing how the user changes the zoom level of the data inside the MagicLens using a stretch gesture. [Forlines2005]	84
Figure 3-13: Snapshot for a MagicLens from DTMap showing the different tools available when the lens is locked [Forlines2005].	85
Figure 3-14: Snapshot 1 showing parts of the design of eGrid.....	87
Figure 3-15: Snapshot 2 showing parts of the design of eGrid.....	92
Figure 3-16: Snapshot 3 showing parts of the design of eGrid.....	94
Figure 3-17: Representations for the gestures defined by the users in the study of Wobbrock et al [Wobbrock2009].	100
Figure 4-1: A diagram representing how multiple applications can participate in creating and enriching the library by iterative refactoring.....	113
Figure 4-2: A flowchart summarizing the extraction of the library from App#1, App#2 ...etc.	115
Figure 4-3: A diagram representing part of the design in eGrid before any extraction is done	120
Figure 4-4: This diagram explains how the acceptance tests of eGrid and Sky Hunter interact with the class and mock classes in the implementation of eGrid.....	121

Figure 4-5: Example Acceptance test to capture the user story related to using the map window component in eGrid.....	123
Figure 4-6: Example Acceptance test to capture the user story related to using the map window component in Sky Hunter.....	124
Figure 4-7: This diagram explains how the MapWindow class and its associated Mock class and test is moved into the library layer and used from there in both eGrid and Sky Hunter.	127

Chapter One: Introduction

1.1 Overview

Geospatial information is used by urban and regional planners, teams involved in crisis management, command and control, operators in utility companies, in oil and gas industry and other domains. In many cases, the use of geospatial data is very critical to effective and collaborative decision-making. The term geocollaboration refers to group work in relation to geographic problems facilitated by geospatial information technologies [MacEachren2003], such as GIS (Geographic Information Systems).

Traditionally, teams working in the control centers of utility companies use large paper maps unfolded on tables to accomplish a variety of tasks in a collaborative environment, such as troubleshooting emergency power outages and discussing alternative solutions.

Lately, digital GIS systems are becoming more popular, and teams working on geospatial information have been changing their traditional methods, embracing different GIS applications for storing and managing critical data. Desktop GIS applications solved the problem of data management and offered up to date data. However, these applications have not completely replaced large printed paper maps up until this point, since they are not designed for multi-user access and high-level abstract queries. Current GIS interfaces provide limited interaction tools and most of these tools only support a single user, which makes them not suitable for geocollaboration and very restrictive in terms of extracting important information or developing planning scenarios effectively [Brewer2000]. In

addition, decision makers do not always have the real time spatial information they need. They end up requesting GIS analysts to produce PDF maps manually to be printed, often leading to overlapping requests with slow delivery times [Rauschert2002]. To be able to overcome these limitations, interactive collaborative GIS applications are much needed.

A few research studies attempt to design different interactive geospatial applications. Some of these applications are multimodal, accepting different types of input such as voice and hand gestures. Some other attempts use multi-touch technology on interactive walls and some use horizontal touch surfaces. The main goal of these studies is to overcome analyst-driven, menu-controlled, keyboard and mouse operated GIS by designing GIS interfaces which are better for collaboration and can put geospatial data directly in the hands of decision makers [Rauschert2002]. In a collaborative environment, sharing a single mouse is very awkward. Furthermore, providing multiple mice makes it hard to figure out which pointer belongs to which user. Touch screen technologies are more promising in this case [Esenther2002]. In fact, the domain of geospatial data analysis provides an interesting area of research for multi-touch applications [Schöning2008b], since managing geographic space and operations such as selection, modification and annotation of geospatial data are complicated tasks and can definitely benefit from new interaction techniques to make them simpler and more user-friendly [EuropeanCommission1998]. Furthermore, horizontal multi-touch tabletops are more advantageous for geocollaboration as they help different users comfortably interact with

the data at the same time and provide awareness for different user activities. Users can actually place paper, devices, and even coffee cups on top of them. Using tabletops for collaboration also support better eye contact and better seating arrangements. The information flow and interaction is flexible and seated meetings typically are less exhausting than standing at a wall, and support more socially acceptable interpersonal arrangements [Furuichi2005].

1.2 Problem of Interest

The research described in this thesis explores the requirements and design decisions needed to create an interactive multi-touch tabletop environment for geocollaboration in the control center of utility companies and possibly other domains. More importantly, this research explores agile software development approaches that can be used for designing and developing similar interactive systems or reusing parts of them across different domains. Some research attempts exist in relation to designing interactive GIS interfaces, yet research on agile design approaches and practices to be used in these types of innovative interactive projects is a fairly new and interesting research area.

This research explores the possibility of using agile approaches to design an interactive collaborative GIS application to be used by a local utility company. The study suggests some agile practices that can be used to gather the requirements for this new system and evaluate those requirements. It also gets user feedback regarding the suitability of using

multi-touch tabletops as the interactive technology of this new application and some design ideas that can potentially ease the adoption of this new system and allow the users to quickly achieve operational proficiency. Examples of the agile practises used are user interface sketches, interactive prototypes, frequent demos to potential users and iterations of design enhancements, observation sessions in the control center and interviews with different users. Moreover, the study attempts to generalize the requirements and create a library of reusable components that can be helpful to the design of new interactive GIS applications for other geospatial domains. The library is created using a proposed agile approach for extracting reusable components.

1.3 Research Goals

This research aims at using agile software engineering approaches to reach the following goals:

- Gathering and evaluating the requirements for an interactive collaborative GIS tabletop system supporting the control center of the collaborating utility company.
- Designing the new system, developing an advanced prototype as a proof of concept and using user feedback to iteratively evaluate and enhance the design.
- Generalizing the design and extracting a library of reusable components which can potentially be used to support the development of similar GIS tabletop applications.

1.4 Methodology

This research is composed of two case studies. The first case study consists of gathering the requirements, designing and developing eGrid, which is the interactive system prototype. This thesis will explain the agile approaches used to achieve the goals of the case study such as using different prototyping techniques, observation sessions and user interviews. The design decisions resulting from user feedback throughout the iterations will be discussed as well. The second case study consists of creating a library of reusable components extracted from eGrid, called GISforTT, to benefit new interactive tabletop GIS applications. The extraction approach used is guided by the requirements of a new application with an industrial partner, Sky Hunter, and is supported by agile testing practices and existing usage examples from eGrid. The design of the library is iteratively enhanced using some preliminary usability feedback.

1.4.1 Case Study 1: Requirement Gathering, Assessment and Prototype Design

The first case study in this research attempts to design and develop an environment for geocollaboration on interactive multi-touch surfaces using agile software development methods. Using agile practises for designing and developing multi-touch surface applications is not studied well in the literature, especially when the design uses novel interaction techniques and focuses on user collaborative. The prototype is a table-based environment – eGrid – that aims at supporting the operation of the control center of a collaborating industrial partner, local electricity company. The purpose behind creating

eGrid is to complement the use of paper maps in the control center of the company and potentially replace paper maps in the long run. The project is motivated by the disadvantages of using large paper maps reported by the operators of the control center of the electricity company, such as having to repeatedly print new versions and copy all the critical annotations by hand, as well as frequently working on data which is not up to date. The idea for the new environment is also motivated by the fact that the electricity company was in the process of converting the geospatial data into newer formats and using ArcGIS systems for managing this data [ArcGIS2011].

This case study focuses on requirement gathering, iterative design and continuous assessment of the design based on users' feedback. Requirements for the prototype are gathered using different methods such as iterations of low-fidelity and high-fidelity prototypes and demos to potential users. The design is enhanced through multiple iterations and further assessed through interviews and observation sessions in the control center of the collaborating utility company. Designing and developing eGrid addresses the challenges of iteratively developing tabletop applications using an agile software development process and the result of this case study forms the basis for the second case study.

1.4.2 Case Study 2: Reusable Library Extraction and User-Feedback-based Enhancement

The second case study of this research consists of creating a library of reusable components generalized and extracted from eGrid, which was developed in the first study, in order to use such components in the development of new geospatial tabletop applications. The goal of this study is to test the feasibility of creating libraries of reusable components in an agile manner by extracting these components from existing applications.

Reuse in general has many benefits [Mili1995], including reduced problem solving time and fast delivery of products, reduced maintenance costs, reduced number of defects, and providing better project estimates [Jacobson1997]. In agile software engineering, little time is spent on gathering requirements and doing upfront design to facilitate the creation of reusable components [Washizaki2003]. In order to gain the cost effective benefits of reuse and agile methods at the same time, a structured refactoring approach is suggested in this case study. Refactoring is a common agile practise for cleaning up artifacts to make them more reusable [Sugumaran2008]. The approach proposed is used to extract potential reusable assets from eGrid based on the requirements of a new interactive geospatial application called Sky Hunter. The extracted components are grouped in a library to benefit other geospatial multi-touch tabletop applications and the design of the library is iteratively enhanced by getting users initial usability feedback.

1.5 Thesis Structure

This dissertation is structured as follows. After the introduction chapter, Chapter 2 includes an overview of related research in the areas of designing interactive geospatial applications, designing gestures and interactions on multi-touch surfaces, and finally reuse and extracting reusable assets. Chapter 3 describes the first case study including the description of the requirement gathering, design and assessment approach used for creating eGrid. Chapter 4 describes the second case study including the approach used for extracting reusable assets and enhancing the design based on user feedback. Chapter 5 is the conclusion of this research describing the results, contributions, limitations and future work.

Chapter Two: Related Work

2.1 Interactive GIS Applications

This section describes some interactive Geographic Information System (GIS) applications found in the literature. In this context, the term ‘interactive applications’ refers to applications that use innovative user interfaces other than the traditional menu-driven approach to generally enhance the user experience. A GIS is used for capturing, storing, managing, analyzing, and displaying all forms of geographically referenced information. Unlike general-purpose virtual globe applications, such as Google Earth [GoogleEarth2011], any GIS implementation serves a specific purpose depending on the application it is used for. GIS applications usually handle very complex datasets and operations; therefore user interface design is a hard task [Coltekin2003]. The applications described here try to get away from the common GIS interfaces that sometimes rely on a background in Computer Aided Design (CAD) such as engineering sketches and drafts. Some of these applications utilize multi-touch tables, large touch-enabled walls or interactive white boards (IWB). A few other applications attempt to create an interactive multi-modal experience accepting a combination of touch interactions and voice recognition. Other applications use three dimensional virtual reality environments. A Virtual Reality (VR) environment is an environment where the human perception of a simulated environment is as close to the perception of the real world as possible [Coltekin2003]. VR is a new technology that is still evolving and it is still trying to

satisfy the human perception fully. The following subsections describe the design of some of these applications.

2.1.1 Virtual Globes and Spatial Thinking

Schöning et al researched the possibility of using multi-touch virtual globes to do spatial thinking tasks, helping the user inquire about both the “what is where” and “why” of spatial distribution [Schöning2008]. They developed a prototype using a multi-touch wall-sized interface and they focused on maximizing the ease of use of the system. The display is a low-cost, large-scale (1.8 x 2.2 meter) multi-touch surface that utilizes the principles of frustrated total internal reflection (FTIR), presented originally by Jeff Han [Han2005]. The application is designed to help users perform simple spatial tasks such as navigating the map and executing simple spatial queries as opposed to more complex GIS editing features that are not provided by this application. The basic map interaction tasks such as pan, rotate, zoom and tilt are implemented similar to the video by Jeff Han [Han2005] and how it is commonly implemented in the iPhone [Apple2012].

One important observation from this study [Schöning2008c] is that users initially preferred simple gestures, which are familiar from systems with mouse input using the WIMP (windows, icons, menus, pointers) desktop metaphor. After experiencing the potential of multi-touch, users move towards more advanced gestures [Wilson2008], but users usually choose to use single hand gestures in which the non-dominant hand just sets

a frame of reference to determine the navigation mode and the dominant hand specifies the amount of movement. Figure 2-1 shows a snapshot from Schöning et al research where a user is interacting with virtual globe data on an interactive wall [Schöning2008b]. This design proposed by this study does not solve the problem of working in a collaborative team. It is also a general purpose prototype for interacting with the spatial data without proposing solutions for supporting annotations or domain specific requirements.



Figure 2-1: Snapshot for interacting with spatial data from Schöning et al research [Schöning2008b].

Another research presents a prototype called the LIFE-SAVER developed by Nóbrega et al to support visualization and interaction in the field of Emergency Management [Nóbrega2008]. The focus in this design is to allow the users to interact in a visually rich environment that allows a fast perception of the emergency that is being simulated while at the same time being an easy to use environment such that people with no special computer skills can use it effectively. This prototype uses a graphical game engine to simulate a natural disaster scenario where the user can observe and interact with the data. It supports the visualization of multiple maps, including seeing the map in false colors view, in satellite view, military charts view or black and white. The map can be scrolled in any direction, rotated, zoomed and tilted which is useful with 3D maps [Nóbrega2008].

The LIFE-SAVER prototype loads data from ArcGIS ShapeFiles [ArcGIS2011]. However, the data loaded is three dimensional, the interface is similar to Google Earth's configuration and it utilizes an interactive White Board (IWB) that supports only one touch at any given time. Because of the hardware limitations, the interface is menu driven, only one user can use the system at a time and no synchronous collaboration is supported. Map navigation is also limited due to synchronization problems between the touch point and the cursor position. Figure 2-1 displays a snapshot for a user interacting with spatial data on the interactive white board using LIFE-SAVER. [Nóbrega2008]



Figure 2-2: Snapshot for the LIFE-SAVER prototype for emergency spatial data visualization by Nóbrega et al [Nóbrega2008].

2.1.2 Environments for Geocollaboration

MacEachren et al described a group of prototype systems in an attempt to investigate the challenges of supporting group work with geospatial information [MacEachren2003]. They have developed a conceptual approach to geocollaboration; a term they used to refer to the activities involved when a group of users collaborate in interacting with geospatial data. They focused on same-time, same-place group work environments; otherwise known as collocated collaborative teams, using large-screen displays and natural interaction techniques such as hand gestures. They compared between two environments. One adopts a white board metaphor (or wall map) while the other adopts a drafting table metaphor and they considered two types of users: group data exploration (by scientists and analysts) and group decision-making (by crisis managers and planners)

[MacEachren2003]. The first metaphor is used in situations where one or two individuals take a lead role in presenting information and steering a group discussion. According to this research, this interface, which resembles a traditional white board or black board, affords the action of walking up and drawing or writing, then giving way to another actor. The second metaphor is the drafting/work table, which is similar to the prototype described in this thesis. It affords group activity around the map display creating an environment similar to working on a large map on a drafting table. They further explain that this metaphor is typical of military and emergency management personnel in the field or urban planners in the office.

MacEachren et al [MacEachren2003] also describe a third metaphor in which activity spaces allow the users to enter and behave within them; otherwise called immersive environments or virtual reality environments for group work. Neves et al [Neves1997] developed an immersive virtual workspace based on a GIS room metaphor (a room in which maps can be mounted on the wall or placed on a digitizing tablet for encoding in the system). They implemented the environment only for individual users but, conceptually, the metaphor could support multiple users.

2.1.3 Interactive Walls and Access Rights

Schöning et al researched access and security problems, in the context of collaborating teams working on geospatial data, especially if larger teams operate touch surfaces with

different access rights [Schöning2008c]. Their research focuses on strategies to authenticate and interact with sub-regions of a large-scale multi-touch wall, when team members have several levels of authority or specific roles, which determine what functions and objects they are allowed to access. Their system is a several square meter interactive wall and they discussed possible solutions to address two main challenges: User identification and the definition of the region of influence. Examples of solutions to the user identification problem include using biometric identification, such as finger print identification, tracking user movements using a camera, using pressure sensing plates, or using external devices or tokens, such as a RFID-tag or a mobile phone. If the user position is identified, the natural area of influence can be defined by a radius of the length of one arm with a centre in front of the user. This research may be especially useful in military and highly critical geospatial data environments where maintaining access rights is essential.

2.1.4 Combining Speech and Gesture Recognition

This section describes a few research attempts to create innovative multi-modal geospatial applications especially combining speech recognition and gesture recognition. The first example is an emergency management prototype developed by Rauschert et al in which they attempted to overcome analyst-driven, menu-controlled, keyboard and mouse operated GIS by designing a multimodal, multi-user GIS interface that puts geospatial data directly in the hands of decision makers [Rauschert2002a]. This

prototype, called DAVE_G [Rauschert2002b], uses a large screen display for data visualization and multi-user collaboration and it uses speech and gesture recognition coupled with a knowledge-based dialogue management system for storing and retrieving geospatial data.

Their Dialogue-Assisted Visual Environment for Geographical Information (DAVE_G) uses different interaction modalities, domain knowledge and task context to support collaborative group work with GIS and it is based on the multimodal interface framework by Sharma et al [Krahnstoever2002]. The design of the prototype combines speech and gesture interaction and the argument presented by the authors indicate many advantages over systems that are using only one input modality (e.g. speech) or standard input devices (keyboard and mouse). Speech is not self-sufficient [Rauschert2002b], which is why the DAVE_G prototype uses simple pointing gestures as well as an effective second input modality that is more suitable for expressing spatial relations and is less error prone than if expressed in words. The set-up of DAVE_G uses a large screen display, ceiling microphone domes and active cameras that allow multiple users to move freely in front of the system and issue queries to the GIS. Spoken commands could be chosen freely within the definition of an annotated grammar while natural gestures such as pointing and outlining areas of interest on the large screen display help to specify the spatial concept of many geographical queries [Rauschert2002b]. Figure 2-3 shows two users interacting with the environment using pointing gestures [Rauschert2002a].

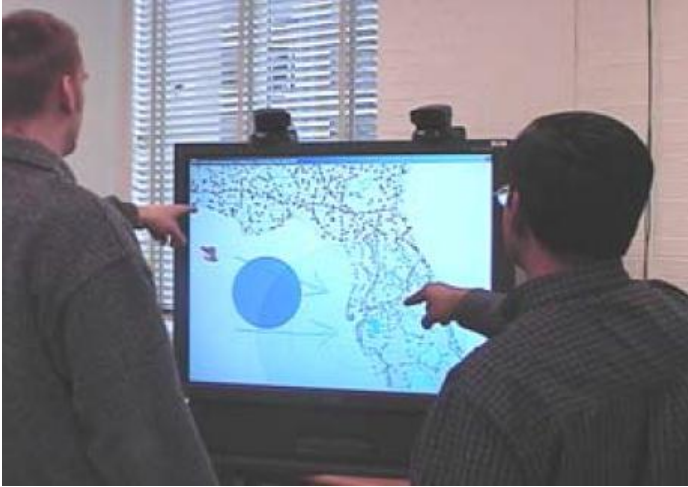


Figure 2-3: Snapshot showing DAVE_G used by two users interacting with GIS data using pointing gestures [Rauschert2002a].

In designing this prototype, domain experts were involved in the earliest stages of system design [Rauschert2002a]. Using questionnaires, the researchers were able to gather information about the most required operations in the emergency management field and they grouped the required GIS functionality into three categories: data query, viewing and drawing. They also used onsite visitations to emergency management operations centers; an approach that is also used by the research described in this thesis.

In the environment of the control center, using spoken commands may be confusing but being able to contact service crews using speech through the environment can be helpful. In addition, using pointing gestures to interact with geospatial data may not be as

accurate as needed in practical life situations. For example, writing annotations using pointing gestures is very hard and not accurate. This prototype also uses a medium sized vertical display which may not be the best option to support more than two collaborating users.

Another system that combines speech and gestures to access geospatial data was introduced by Agrawal et al and it is called GeoMIP [Agrawal2004]. It has been developed to address some of the critical needs in crisis management. They focused the design on vision sensing algorithms, speech integration, multimodality fusion, and rule-based mapping of multimodal user input to GIS database queries. This project was motivated by the fact that common interfaces for crises management are hard to use, require extensive training, and often impede rather than support teamwork [Agrawal2004]. This research attempted to overcome these issues by designing a multimodal GIS interface utilizing a large screen display for data visualization, and collaboration, supporting high-level information requests through voice and gesture recognition. Such a system would provide a collaborative environment where multiple people will be able to share relevant information. Figure 2-4 is a snapshot from the paper by Agrawal et al to show a user interacting using speech and gestures with GeoMIP [Agrawal2004]. This prototype has similar limitations to the DAVE_G prototype.



Figure 2-4: A snapshot of GeoMIP where the user can interact with spatial data using speech and gestures [Agrawal2004].

The VITA system is another system designed as part of an archaeological project to allow scientists to collaborate in a hybrid environment space [Benko2004]. Users of this system wear tracked displays on their heads to visualize 3D spatial terrain data and they utilize a large, high-resolution display, tracked handheld displays, and a multi-user, multi-touch, projected table surface as well. Figure 2-5 shows the devices the user is equipped with to visualize and interact with the data [Benko2004]. The tabletop surface used here is the Diamond Touch table that allows for identifying users provided that users need to stay seated or in specially prepared locations [Esenther2002]. This is one of the applications

that aim at creating a virtual reality environment; a technology that began in military and university laboratories more than 20 years ago [Coltekin2003].

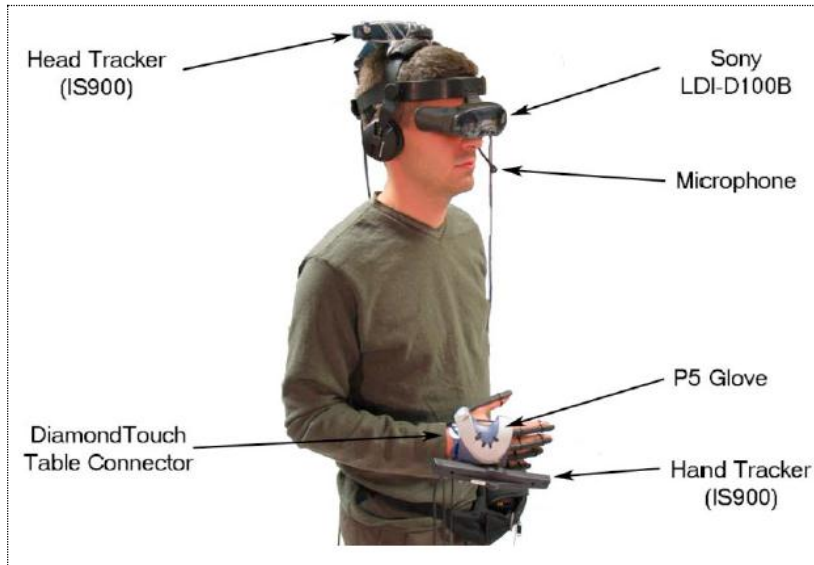


Figure 2-5: A snapshot from the research by Benko et al, showing the devices the user can use to interact with the data [Benko2004].



Figure 2-6: Two snapshots showing users interacting with visualized 3D terrain data by Benko et al [Benko2003].

In this particular system, the users are allowed to use speech commands and also interact with the multi-touch sensitive surface multimodally to collaborate in navigating and viewing the archaeological data. The users are also identified by the application which is a feature supported by the technology of the Diamond touch table. Multiple users, wearing tracked, head-worn, see-through displays, can interact with the environment using tracked, instrumented gloves [Benko2003]. The system combines speech with

head, hand, and arm gestures to aid the users. Figure 2-6 shows two snapshots of users interacting with 3D terrain data by Benko et al [Benko2003]. As interesting as this environment is, it may not be practical enough to be used in day to day control center and decision making environments. The technology is also expensive and may not be available easily to the public [Coltekin2003].

2.1.5 Combining Hand and Foot Gestures

This section describes a couple of research attempts to combine hand gestures with foot gestures in geospatial applications. The first attempt is by Schöning et al, in which they show how multi-touch hand gestures in combination with foot gestures can be used to perform typical basic spatial tasks within a Geographic Information System (GIS) [Schöning2009, Schöning2008b]. Their hypothesis is that combining hand and foot gestures has several advantages over pure hand-based multi-touch systems. Hand gestures are better for precise input especially with point and area information and it is worse for continuous data input for a long period of time, such as panning a map on a large multi-touch wall. In such situations, the research suggests using foot interaction to providing continuous input by just pushing the body weight over the respective foot. They also explain that in addition to providing continuous input, combining multi-touch hand and foot interaction can also help to rethink the use of the dominant and non-dominant hand.

The second system which combines multi-touch hand and foot gestures is presented by Daiber et al [Daiber2009]. Their research explains some approaches to perform typical basic spatial tasks within a GIS such as pointing, panning, zooming, rotating, tilting and cutting by using multi-touch gestures in combination with foot gestures with a focus on studying how non-expert users interact with such multi-touch surfaces. They propose that foot gestures can be more economic in the sense that pushing once weight over from one foot to another is less exhausting than using one or both hands to directly manipulate the application on the surface.

2.1.6 GeoSpatial Information on Touch Tables

This section describes a couple of environments to interact with geospatial information on touch tables. The first prototype application is DTMap, which was developed to illustrate the power of combining visual data with a multi-user tabletop environment [Furuichi2005, Forlines2005]. DTMap was developed for the DiamondTouch table™, which is developed at the Mitsubishi Electric Research Labs (MERL) [Esenther2002]. The DiamondTouch table is a large interactive touch screen that allows simultaneous input of up to four users while it recognizes which user is touching it. The environment of a multi-user interactive tabletop facilitates direct-touch manipulation of user interface elements and provides a shared focus of attention for collaborating users [Furuichi2005]. Tabletop displays are well suited for visual data, such as maps, as opposed to text-based

data because of the orientation and readability issues found with text-based data on horizontal surfaces.

DTMap is a prototype for multi-user, tabletop cartographic analysis application [Furuichi2005]. The interface contains a satellite map image and different views of the same area can be overlaid atop portions of the map. This specific design feature allows multiple users to work on separate detailed areas of the map. The design of DTMap uses the concept of a Magic Lens [Bier1993]. This technique can provide access to multiple layers of information, combining multiple maps into a dynamic, interactive display. Figure 2-7 includes a couple of snapshots of users using MagicLens to display different views of the spatial data [Furuichi2005]. A similar concept to the MagicLens is used in the design of the application in the first case study of the research reported in this thesis.

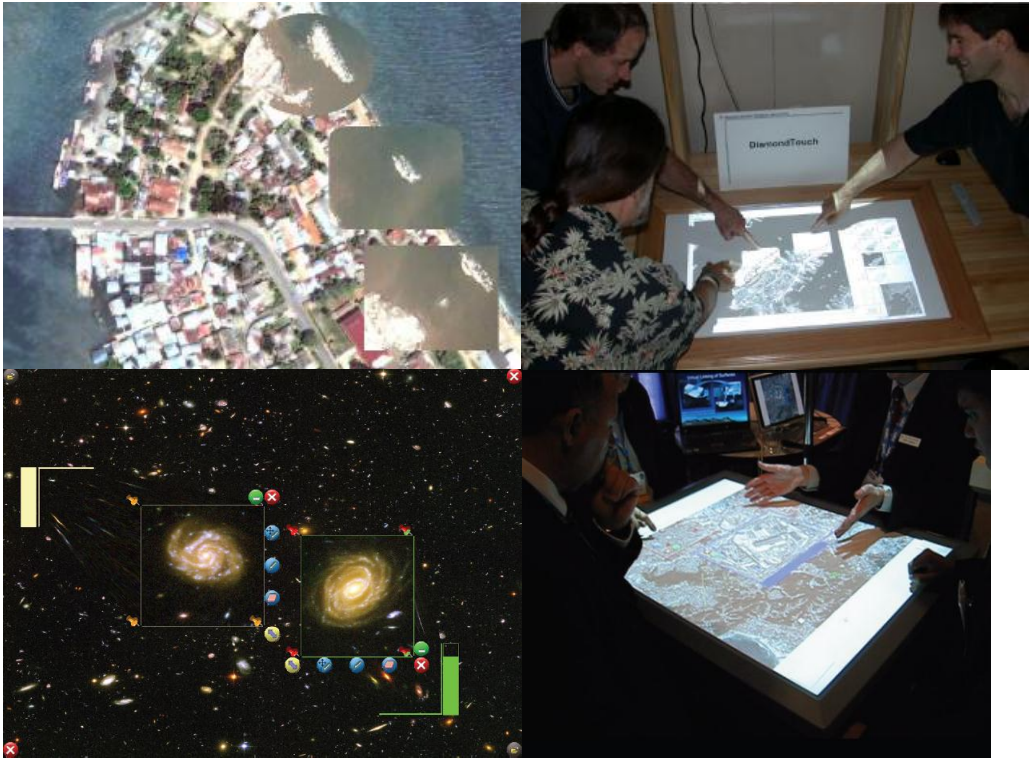


Figure 2-7: Snapshots from DTMap where users are using MagicLens to display different views of the data [Furuichi2005].

Another application incorporates GIS-based tools and a touch table to effectively use spatial information in the decision process for the participatory development of land use plans [Arciniegas2009]. Maps are used to communicate and exchange knowledge among policy-makers and stakeholders and the touch table is used to support stakeholder collaboration through common visualization and spatial information handling. The system also uses the DiamondTouch table™. The decision to use multi-touch table technology was based on a number of factors [Arciniegas2009]. Touch tables have the

potential of supporting face-to-face group collaboration since the users are allowed to maintain eye contact while simultaneously interacting with the display and discussing issues with each other. They also provide a common map interface that complements, rather than replaces, printed maps, which would increase the user comfort level. The tools implemented in this environment are based on ArcGIS 9.3 [ArcGIS2011].

In the research by Arciniegas et al, a number of surveys were conducted at the end of Design and Analysis workshops, which reflected that the majority of participants preferred the touch table over printed maps. Participants indicated that their decision is due to the fact that they can choose and combine background maps, navigate and zoom across the area and see what other participants do. Printed maps were preferred by very few participants in that study. In addition, a number of survey participants also found that the touch table increased their awareness of new aspects of problems in the area and a large proportion of the participants indicated that using a touch table can significantly stimulate the discussions, which results in better collaboration [Arciniegas2009]. The results inspire the choice of using touch tables in place of paper maps in the case study presented in this thesis.

2.2 Tabletop Environments and Gesture Design

This section describes research on different tabletop environments and designing multi-touch hand gestures. There is a large body of research literature concerning the design of

hand gestures to be used on touch sensitive interfaces. One of the important challenges of surface computing is designing interaction techniques to be used as an alternative to traditional input using the keyboard, mouse, and mouse-based widgets. Gestures are versatile and can be used to perform various tasks. This section describes a few studies related to interactive tabletop applications, multi-touch gestures and interaction design.

Wobbrock et al presented an approach for designing tabletop gestures based on non-technical user preferences [Wobbrock2009]. The aim of the study is to consider surface gestures that users make without regard for recognition or technical concerns. The study first portrayed the effect/result of a gesture, and then asked users to perform the gesture which possibly caused the effect/result on tasks such as word processing and managing documents. The findings of the study indicated that users rarely care about the number of fingers they employ. They preferred using one hand over using two hands, given that the study used a Microsoft Surface table which is of a relatively smaller size compared to other multi-touch tables. The study also revealed that desktop idioms strongly influence users' mental models. Some of the commands they used in the study elicited little gestural agreement between the participants which suggest the need for an on-screen widget for extra guidance. In this study, the authors presented a user-defined gesture set for some of the common commands in the list presented in Figure 2-8. This set of gestures was very helpful in the design process of the first case study in this thesis.

1. Move a little	15. Previous
2. Move a lot	16. Next
3. Select single	17. Insert
4. Rotate	18. Maximize
5. Shrink	19. Paste
6. Delete	20. Minimize
7. Enlarge	21. Cut
8. Pan	22. Accept
9. Close	23. Reject
10. Zoom in	24. Menu access
11. Zoom out	25. Help
12. Select group	26. Task switch
13. Open	27. Undo
14. Duplicate	

Figure 2-8: A list of the tasks presented to the users in the study by Wobbrock et al.

Another research which examines user preferences for tabletop gesture interaction is the research by Epps et al, which focuses on the use of different hand shapes to open a new vocabulary of interaction [Epps2006]. An observational study is done with the main objective of determining the types of hand shapes users preferred for different types of tasks performed on a tabletop. The hypothesis was that users would reuse simple, primitive hand shapes for a variety of different tasks and this hypothesis was supported by the results of the observational study.

Building on top of the two previous research studies, Micire et al explored the type of gestures which the user preferred but in the areas of command and control and crises

management [Micire2009]. Common gesture sets were previously often tailored around detectability and repeatability depending on the multi-touch technology used and the capabilities of the touch sensor mechanisms. In this research, the focus is more on the gesture ease of learning and user comfort as opposed to detectability. In an ideal design, a naive user would be able to interact with the multi-touch interface quickly, naturally, and without explicit instructions [Micire2009]. This is especially crucial in the fields of command and control for military operations or disaster response, since the users typically do not have a lot of time to learn the system. Their study used a robot-controlling environment and resulted in understanding that prior experience of the participants, i.e. the standard mouse paradigm, introduced some bias into the gestures that they wanted to use. Additionally, participants who had experience with iPhones used significantly more pinch gestures for zooming, and heavy users of real time strategy games expect to have similar controls.

Another research concerned with intuitive and comfortable hand gestures when using multi-touch sensitive displays is by Koskinen et al [Koskinen2008]. The user study presented was part of an “Affordance table” concept design process. It focused on gestures for manipulating industrial equipment representations such as valves and switches. Simple gestures were preferred by the participants over more complex ones. Participants preferred one-finger/handed interaction and a minor contact with the surface and the dimensions of gestures such as duration, force and direction had an impact on

participants' evaluations. In addition, the study reported that as the motion increased, the participants started to report lack of tangibility and feedback that made the interaction feel unnatural. This research also classified touches and gestures into three groups: presses, straight sweeping motions and rotating or curved motions. Participants were more comfortable using presses than more complex rotating gestures. They reported that complex gestures were difficult and uncomfortable.

Current multi-touch surface technologies introduce several challenges for interaction designers, such as discrepancies in the resolutions of the visualization, the tabletop's display, and its sensing technologies. Vaida et al studied some of these challenges which cause most of the interactive tabletop applications to have limited functionality and power compared to their desktop counterparts [Vaida2009]. They studied design considerations to help build practical tabletop applications, especially when the data sets become massive; when scientists' fingers are much larger than the visualization details they want to manipulate; when interface controls for controlling the view into the data compete with interface controls for manipulating the data, themselves; and when collaborators sitting or standing around a tabletop each have a different perspective on the tabletop's display contents [Vaida2009].

In that research, they try to address the "fat fingers" problem with low-resolution input. One of the approaches discussed is providing flexible zooming or magnification

capabilities to allow users to show areas of interest in a resolution that is better supported by the display hardware and consequently allow for more fine-grained touch interaction. A similar approach has been used in the design of the application presented in this thesis based on the concept of a MagicLens [Bier1993]. They also discuss some design considerations related to facilitating face-to-face collaboration. They compare between the approaches of treating the entire surface as a single, stretchable sheet, providing tools to manipulate the scale of individual objects, or providing individual magnification lenses for different users [Voids2009]. They also discuss similar approaches to handling the orientation of the visualization such that it can support participants standing around different sides of the table.

Spindler et al presented the concept of PaperLens to improve the three-dimensional (3D) exploration of virtual spaces above a tabletop [Spindler2009]. They developed a set of navigation techniques using a handheld magic lens, which is a tracked sheet of paper used to navigate the Z-dimension (height above the tabletop). The original concept of Magic Lenses [Bier1993] as a detail and context technique can be extended to be used in tabletop applications, but it does not solve the problem of exploring the volume above a display in which case the PaperLens can be useful. They distinguish between different information spaces to explore above the table: volumetric, layered, zoomable, and temporal, shown in Figure 2-9. By moving the tracked sheet of paper above the table, the displayed content adapts according to the current position with regard to the contextual

tabletop display [Spindler2009]. This concept can be used with GIS data to navigate through different layers of data or different zoom levels.

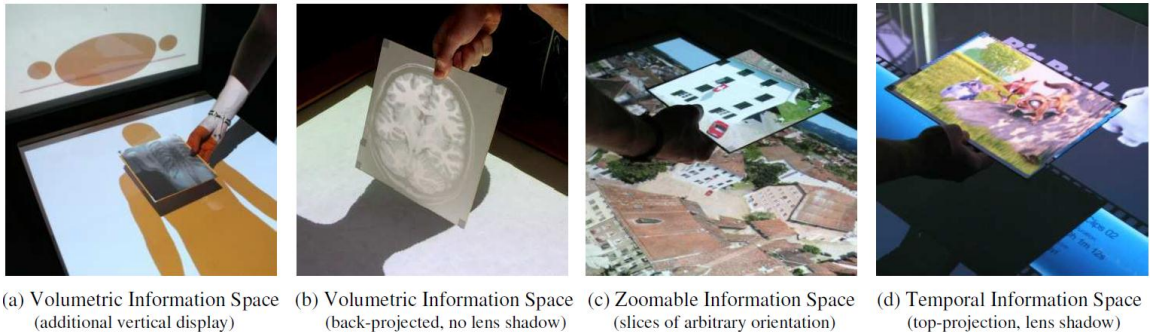


Figure 2-9: Examples for different information spaces that can be navigated using the concept of a PaperLens by Spindler et al [Spindler2009].

Another interesting research was done by Hesselmann et al in which they explored the design of stacked half-pie menus to be used in tabletop user interfaces in place of regular drop down menus [Hesselmann2009]. Stacked Half-Pie menus allow the visualization of an unlimited number of hierarchical menu items as well as interactive navigation and selection of these items by touch. The special advantage of pie menus lies in the fact that movement distances needed to reach an item in the menu are generally less than in linear menus, such as drop-down menus. Figure 2-10 shows snapshots from the implementation of stacked half-pie menus [Hesselmann2009].

In the case of tabletop interfaces, this design may be useful to avoid the occlusion and orientation problems caused by regular drop down menus. However, the design presented by Hesselmann et al occupies a lot of space on the screen and a practical alternative needs to be designed. The application presented in this thesis as part of the first case study utilizes simple quarter pie menus on the corners to help users access common application tools. If more tools need to be added to the menus, inspiration can be drawn from stacked half-pie menus to rotate the menu and reveal more tools.

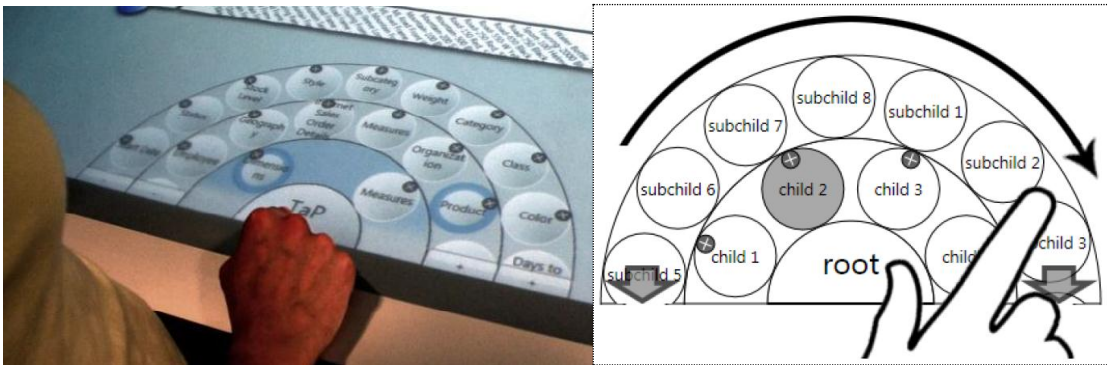


Figure 2-10: Snapshots from the research by Hesselmann et al representing half-pie menus and how users would rotate the menu to reveal more buttons on each layer [Hesselmann2009].

2.3 Reuse and Extracting Reusable Assets

Software Reuse has been introduced long ago with the advent of Software Engineering itself. The ability to reuse existing assets and maximize the return on investment with regards to programming effort is studied extensively in research. It is the only realistic approach to bring about the gains of productivity and quality that the software industry needs [Mili1995]. Software systems in a given domain solve similar problems; and therefore, there is a high potential for reuse. Even for software systems across different domains, similarities exist [Ghanam2011]. Reusing common parts/component can reduce the development cost of new systems, and reduce the maintenance cost associated with the support of these systems [Washizaki2004b]. There is a large body of literature on the concept of reuse, its approaches and techniques. Jones identified four types of reusable artifacts [Jones1984], namely data reuse, architecture reuse, design reuse, and program reuse. This section discusses reuse in terms of code reuse especially in an agile software development environment.

2.3.1 Overview

The second case study in this thesis attempts to extract a library of reusable components in an agile manner from the system developed in the first case study. Reuse approaches are commonly categorized under planned reuse [Wartik1992, Prieto-Diaz1996] or opportunistic reuse [Prieto-Diaz1996]. Planned reuse is a proactive approach in which the organization dedicates the resources necessary to plan reuse upfront. In this type of reuse,

the software process defines what assets are to be reused and how to adapt them [Wartik1992, Prieto-Diaz1996]. Some work in the literature addresses the development of reusable assets as a proactive activity in which reusable assets are planned for upfront [Clements2001, Pohl2005] and these approaches involve the efforts of designing for reuse [Bieman1995] and also managing reusable assets [Henninger1997]. On the other hand, opportunistic reuse refers to a reactive reuse approach that happens only when the opportunity for reuse avails itself [Prieto-Diaz1992]. In agile software development methods such as Extreme Programming (XP), the focus is not on developing software for possible future reuse in order to avoid overhead during development. The focus is on keeping it simple and developing only what the customer really wants at the current point in time, which is why opportunistic reuse seems like the natural choice. The approach described in this thesis is an opportunistic reuse approach that is done reactively when an opportunity for reuse appears.

The extraction approach proposed in the second case study of this research relies on refactoring which is an essential part of the development cycle in Extreme Programming [Fowler1999]. Agile Methods in general promote refactoring since it is supposed to improve understandability and maintainability of source code. In his book on refactoring [Fowler2000], Fowler stresses that refactoring is important in making the code easier to understand and to read. This should facilitate reusing pieces of software that are not designed for reuse. Refactoring can enhance the reusability of project assets [Mens2004].

It is actually sometimes defined as the process of “cleaning up” artifacts to make them more reusable [Sugumaran2008]. A study conducted by Moser et al. assesses whether refactoring in agile environments improves the quality and reusability of – otherwise hard to reuse – classes [Moser2006]. Their hypothesis is that the practice of continuous refactoring may improve internal quality metrics and affect reusability of a software system in a positive way [Caballero2002] Their results support the hypothesis that continuous refactoring improves quality metrics thus promoting ad-hoc reuse of object-oriented classes, which is why this thesis employs refactoring heavily in the development process to improve the reusability of the design and make the extraction easier.

2.3.2 Relevant Reuse Approaches

In the second case study of this thesis, an opportunistic reuse approach is used and reuse related activities are not attempted until there is a demand to provide a reusable asset to be used across a second application. It is also an extractive approach that attempts to extract the potentially reusable components from existing applications. Related extractive approaches are described in literature that attempt to identify potentially reusable assets in existing applications and extract those assets [Krueger2006, Burd1996]. For example, Lanubile et al. applied a program decomposition method to the problem of extracting reusable functions from ill structured programs [Lanubile].

Another approach by Ning et al. also relied on segmenting the programs into manageable pieces before the extraction process [Ning1993]. Other efforts focused on searching for and retrieving reusable assets to be used in new applications [Frakes1994]. These methods are different from the approach described in this research and they are not intended for agile software development teams. The approach proposed in this thesis uses the same two general steps of focusing and then extracting, however the needed focus is achieved differently using user stories and tests. The reuse approach proposed here includes code components as well as other artifacts such as use cases and test cases [Mohagheghi2004].

Washizaki et al. proposed a refactoring approach which can be used under agile methods for extracting candidate reusable classes from object oriented programs and modifying the surrounding parts of the extracted parts in the original programs [Washizaki2003, Washizaki2004a]. The study resulted in a tool that analyzes Java source code and attempts to automatically extract reusable components using refactoring. These components can be reused independently in other projects. The tool focuses on using Java as the programming language and JavaBeans as the fundamental component architecture. This approach also extracts usage examples along with the extracted components. Usage examples are important due to the fact that even when the components are extracted from existing programs, it is difficult to identify the appropriate use of an extracted component

only by referring to the source code or public interface of that component.

[Washizaki2004b]

The extraction process used in this thesis is different from the approach by Washizaki et al since it takes into consideration new user stories from new applications in order to enhance the design or/and extend the implementation of the extracted assets. In addition, it is not specific to any language and the definition of the component to be extracted relies on its functional usage in the system rather than how it is defined in the architecture.

The approach used in this thesis also leverages tests as a focusing mechanism and as a safety net. A study by Nasehi et al discussed using unit tests as API Usage Examples [Nasehi2010]. Learning about APIs and application components can be done through standard documentation and sample code. This study, however, concluded that unit tests can be extremely helpful, especially when the task is complicated and involves multiple classes and methods. According to this study, well-written tests are good source of examples. This is not limited to unit tests but also includes acceptance tests. Acceptance tests document the requirements and represent the customer's interests [Miller2001]. Therefore, they can also be used to understand the usage of software components.

Within the context of agile software development [Agile Manifesto], Sugumaran et al. proposed the construction of a knowledge-based framework to enable agile teams make better decisions when selecting and customizing software components for reuse [Sugumaran2008]. It addresses the challenge of how to select software components from repositories to meet an application requirement, so it is not an extractive approach. It also focuses on how to use such a framework in an agile context but it is not clear on how to build the proposed framework within an agile context. The approach described in this thesis is simpler and more bottom up.

McCarey et al developed a tool for agile reuse named RASCAL, which promotes reuse in agile development through a technology called “software recommendation” [McCarey2005]. RASCAL is an Eclipse plug-in that uses collaborative and content-based filtering techniques to suggest method invocations to developers. The tool monitors method invocations in the class currently under development to predict method calls that are likely to be soon needed and suggests them to the developer. This tool promotes reuse in a proactive recommendation-oriented way. Based on this concept of agile reuse, another approach was presented by Hummel et al and termed ‘Extreme Harvesting’ [Hummel2007]. This approach tries to tightly integrate reuse into agile methods using test cases to influence the component search process. It adds systematic reuse to agile methods, thus building on top of the notion of agile reuse presented by McCarey et al.

Another agile reuse approach was presented by Lemos et al, in which test cases are used as an interface for automating code search and reuse [Lemos2009]. The approach is called Test-Driven Code Search (TDCS) and in this approach test cases are used to determine the behavior of the desired functionality to be searched and to test the matching results for suitability in the local context. This approach is specific to code search queries and not related to extracting reusable assets from existing software implementations.

Another related area of research is studying software product lines. A software product line is a family of products sharing some core assets that accommodate for some variability according to the requirements of different applications [Clements2001]. Combining the concept of software product lines with agile software engineering necessitates figuring out the requirements along the way rather than investigating all the requirements upfront (which is not always possible). In this case, the product line is built one application at a time, a concept that is similar to the approach presented in this thesis. Ghanam et al attempted to reduce the conflicts between agile software development and software product line engineering by proposing a framework which enables agile organizations to introduce variability into systems when it is needed and constructing variability profiles for existing and new systems [Ghanam2009, Ghanam2010]. In this research, an approach is proposed in which agile organizations can instantiate various products from a core system. The approach utilizes test artefacts that are produced in

agile environments. The resulting group of software products are managed as one product line with variation points.

Chapter Three: CASE STUDY 1 – eGrid

The first case study of this research consists of gathering the requirements and designing an environment, eGrid, to support geocollaboration in the control center of a utility company using multi-touch tables as the hardware technology. The main goal of this case study is to investigate the possibility of using agile approaches to design interactive collaborative tabletop applications and to suggest agile practices which can be used to engage the users in gathering the requirements for this new system and evaluating those requirements. Another goal is to experiment with the technology of multi-touch tabletops to solve some geocollaboration challenges in the collaborating utility company.

3.1 Application Domain

The term Spatial Data or Geospatial Data refers to a type of information that identifies the geographic location of features and boundaries on Earth. It is usually stored as coordinates and related information and it is often accessed, manipulated or analyzed through GIS. In recent years, using GIS has become commonplace in industry. GIS applications have been developed to help analysts manage and update the spatial data and viewer applications were designed to give users access of this data for viewing or editing purposes. However, until this point in time some industries are still reliant on printed paper maps especially when it comes to fulfilling the needs of group collaboration. For example, the industrial partner of this research is a major utility company and they are still using printed paper maps unfolded on a table in the control center to help the

operators analyze huge amounts of data, keep track of their updates and collaboratively handle emergency situations. These paper maps are printed from electronic versions that are created and updated manually by GIS analysts. The main responsibilities of the control center are to receive support center complaints, analyze power outages and problems reported by customers, and guide field crews into fixing these issues.

Inspired by the popular adoption of GIS technologies in multiple domains, the company initiated a project several years ago to convert their map data from an old format called MicroStation [Microstation] to the GIS technology of ESRI®. MicroStation is a Computer Aided Design software product for two-dimensional and three-dimensional designing and drafting. The migration to ESRI ArcGIS technologies is almost complete now. Analysts will utilize a GIS database to store and manage the data that includes details of the circuits, power-lines, switches, etc. However, operators in the control center are not GIS experts and thus they do not use ArcGIS tools for accessing the data they need as these tools require some GIS background. Typical GIS desktop applications are very powerful from a GIS analyst perspective; however they are not easy to use for non-expert users. In addition, common GIS applications are designed for single-user interaction and may not be suitable for group collaboration. [Rauschert2002b] Therefore, what the control center operators need is a specific application targeted towards their needs, allowing them to accomplish their job tasks easily without the added complexity of using typical GIS applications. A lot of problems facing the users can be solved by

creating a domain-specific GIS-based application, designed such that it is easy to use without adding functionalities that are not needed, and directly supporting domain-specific workflows instead of relying on generic GIS functionality thus allowing the users to be more efficient.

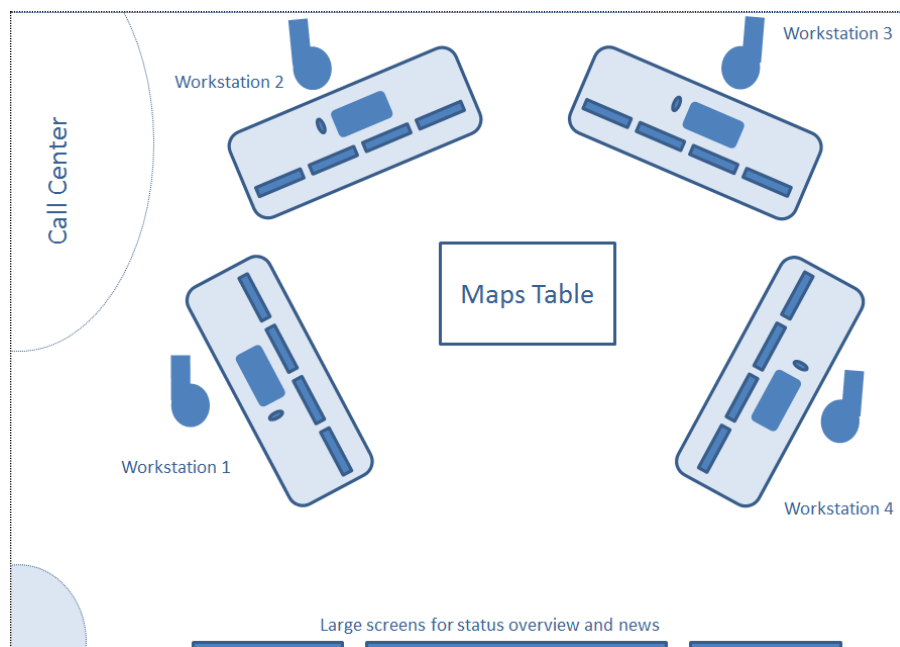


Figure 3-1: A diagram representing the current setup of the environment in the control center of the collaborating utility company.

Currently, the operators are still using print outs of the electricity circuits, maintaining them in drawers underneath a big table in the control center. Figure 3-1 is a diagram describing the current environment in the control center. Paper maps are kept in drawers inside a big central table and operators have their workstations around the center table

with each workstation having multiple monitors. Using paper maps is convenient for control center operators for a number of reasons. First of all, they are very valuable compared to the electronic versions from which they are printed, since they hold valuable comments and information about the actual state of the circuits. These comments are written by the operators on the paper maps in forms of annotations. Operators use the paper maps to collaborate in troubleshooting electricity outages and making emergency decisions since they find it easier and more convenient for group work and discussion to unfold the maps on a horizontal table as opposed to using complicated GIS desktop software. As explained previously, most of the operators are not GIS experts. They have been working with electricity grids for years but most of them are not experts in using complicated GIS applications, thus they prefer more traditional methods.

Using paper maps has some advantages but it definitely has disadvantages as well. Data management tasks, such as editing and updating data, become more complicated and error prone when done on paper and then transferred to GIS. It is hard to synchronize the electronic version of the data with the changes documented only on the paper maps.

Keeping track of pen annotations and making sure everything is up to date is not an easy task. This has actually led them to abandon changing the electronic versions for temporary or transitional changes. They only request updating the electronic versions if the changes are persistent or final and they keep transitional changes only on the paper version. As a result, the current state is only available inside the control center and it is

impossible to see it in the backup center unless the paper maps are physically brought into the backup center. Update delays and inconsistency in the maps can cause confusion when making emergency decisions and pose a threat to the safety of field workers who may be misguided by out-dated information. More details about the operation of the control center and insights gained through the interviews done with operators and GIS analysts as part of this case study are included later in this chapter.

As mentioned before, using paper maps in the context of geocollaboration is convenient and this convenience is due to the fact that these maps are typically large in size which allows everyone around the table to see the information comfortably, enables subgroups to work on different maps concurrently, and gives everyone concurrent access to editing these maps using pens and markers. To be able to solve the disadvantages of using paper maps for geocollaboration, a paradigm shift is needed to design a GIS interface which provides the advantages of using paper maps and at the same time avoids the problematic data management issues. Using the technology of multi-touch digital tables is the suggestion presented in this case study. Within the context of this case study, the suitability of using tabletops as an alternative for using paper maps on regular tables is explored. The assumption is that interactive digital horizontal surfaces can be used for displaying and managing these large digital maps for collaborative purposes. They encourage people to work collaboratively in co-located groups [Shen2006]. The solution provided by this case study in the collaborating utility company is creating a tabletop

environment called eGrid. The goal behind developing eGrid is to provide a convenient digital environment to address the needs of the co-located team of operators in the control center. This environment should allow the operators in the control center to browse through the stored GIS maps, analyze these maps to solve emergency issues and annotate the maps for future reference, which can potentially facilitate their collaboration without the need to print the maps on paper.

3.2 Approaches for Requirement Gathering and Design

This section describes the agile software engineering practices used for requirement gathering, iterative design and preliminary assessment. The project started by discussions with interested industrial partners from the utility company concerning the needs of the utility company and ideas which may be useful in enhancing the operation of the control center. The development process used was chosen to be a simple, lightweight extreme programming process that relied mainly on effective feedback for iterative product improvement. The process started by creating low fidelity prototypes such as sketches and simple mock demos, which were useful for brainstorming interface ideas but not useful in capture user interactions with the interface. One of the strengths of using a multi-touch surface environment is the ability to use gestural interactions. Gestures are finger and hand motions typically used to interact with multi-touch interfaces. Using gestures in the design of eGrid, created the need for more advanced prototypes to capture the interactive nature of the interface. These prototypes were created next to portray the

ongoing design of eGrid and help engage the users to get more design feedback. The design went through a number of iterations of enhancements guided by demonstrations to industrial partners. As a result, the interface and features of eGrid have been changing continuously to respond to the feedback received from frequent demos. In addition, observation sessions and interviews with operators, GIS analysts and management personal were conducted to gain more insight into the environment of the control center and to get more user feedback on the design of eGrid. Details about this process are included in the following subsections.

3.2.1 Initial User Requirements

The initial requirements were gathered through discussions and a couple of meetings with information technology analysts and GIS analysts in the utility company. Through these discussions, a basic understanding of the environment in the control center was formed and issues experienced with the existing system were pointed out. Basically the company was interested in using a digital tabletop environment to replace or complement the use of paper maps in the control center.

3.2.2 Low Fidelity Sketches

Since “a picture is worth a thousand words” [Memmel2008], the process started by creating user story cards and user interface sketches. User story cards usually have a short title and a well-written story using the standard “As a ... I want ... so that ...” format.

They were a great starting point capturing some basic requirements for eGrid that resulted from the preliminary meetings in the utility company. Whiteboards were usually used in the discussions to help focus attention and promote collaboration in the design process [Brown2008]. User interface sketches were also used to brainstorm user interface ideas.

To create the interface sketches, an application called ActiveStory Enhanced [Hosseini-Khayat2010] was used on a tablet computer (which brought it closer to an actual pen and paper design experience). This application is developed for designing and performing usability testing on an application in an agile software development environment. Using this tool, user interface ideas were easy to sketch. Simple tap/click interactions were captured in the form of embedded links to enable traversing the series of sketches as if they were actual screen images. The complete sketch series capturing the initial user interface design was exported using the same tool, ActiveStory Enhanced, to a website which could be browsed. The sketches were useful in brainstorming the design of the interface but had no value in capturing the gestures which were to be designed in eGrid and were also of limited value of actual users who were not motivated to browse through the created website and provide their feedback.

The following figures include some examples of the sketches created in this phase. These are the most important sketches capturing the important parts of the design at this stage.

Other sketches were also used as mostly transitional links between the sketches included here and they do not represent extra design elements. The sketch in Figure 3-2 represents an opened map window, which is a small window showing part of the background map, and a list of trouble reports, which are database records for trouble incidents to be fixed by the control center, along with four corner menus, one of which is expanded to show multiple tools.

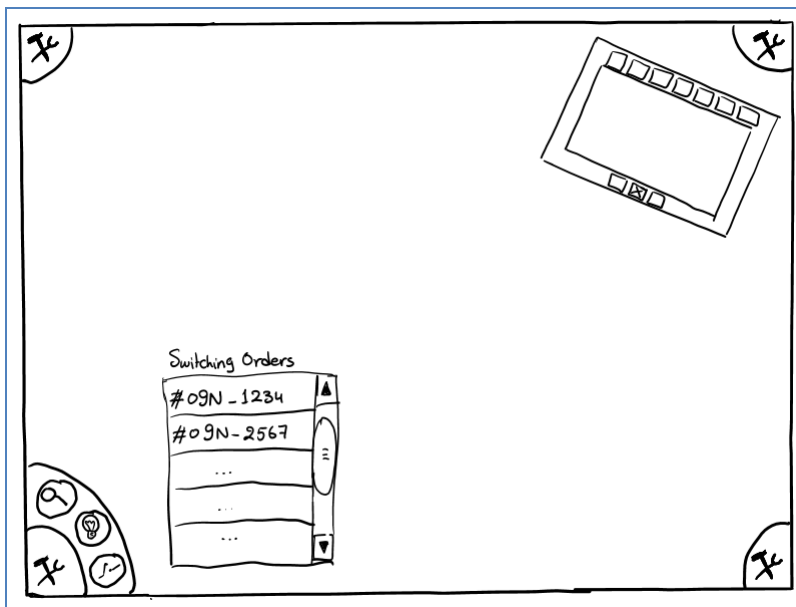


Figure 3-2: First sketch showing a small map window, a trouble report list and four corner menus, one of them is expanded to show multiple tools.

The sketch in Figure 3-3 represents a couple of map windows with the details of toolbox icons on the frame. It was used to show how two map windows can exist at the same time

on top of the background map and how each can have different sizes and orientations.

Each map window has a number of tools in the toolbar.

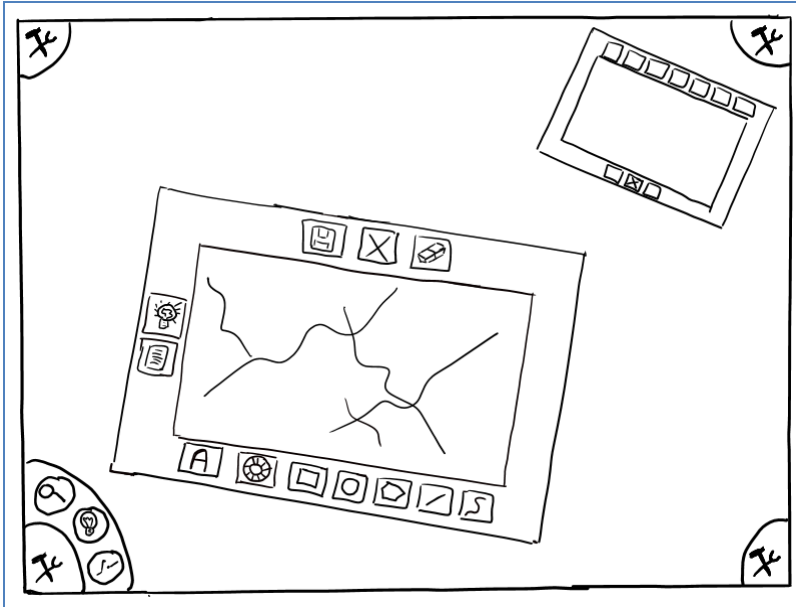


Figure 3-3: Second example sketch that represents two individual map windows with toolbox icons on the frame.

Figure 3-4 includes another sketch that shows one of the map windows with the color wheel in the middle allowing the user to choose the color of the annotation strokes.

Clicking on one of the icons in the toolbar can turn on the color wheel.

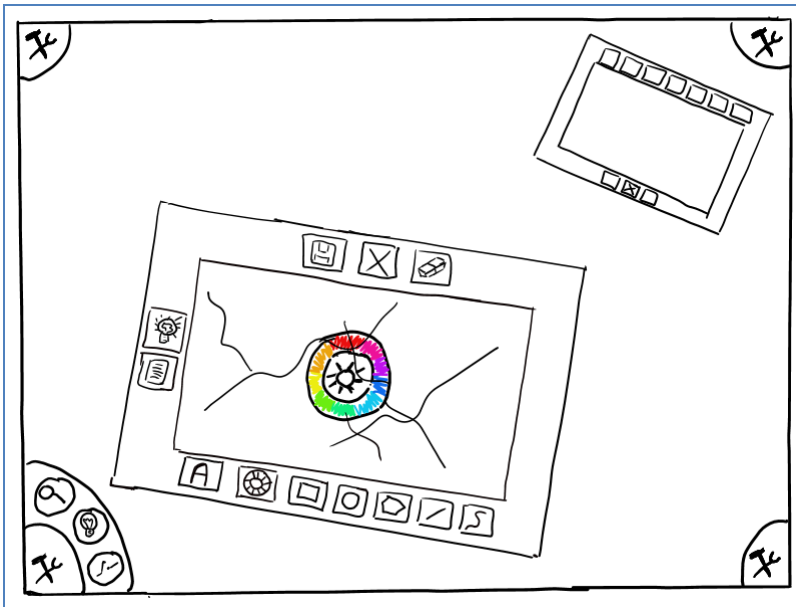


Figure 3-4: Third example sketch describing the color wheel concept, which is used to allow the user to select the color of the annotation pen.

The final example sketch in Figure 3-5 displays some annotations in red, similar to the annotations that would be drawn by control center operators opening and closing electric devices on the map.

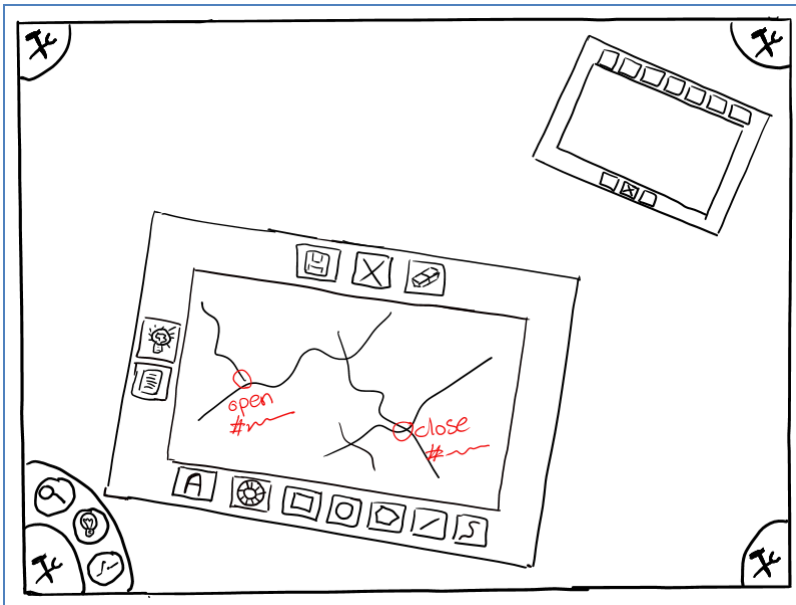


Figure 3-5: Fourth example sketch used to show the annotations that are typically added by the operators of the control center on the maps.

3.2.3 Interactive Mock Demos

The user interface sketches captured the initial design thoughts based on the discussions with some stakeholders. However, even though they were useful in brainstorming interface ideas, they did not fully capture the design of the interactions in this gesture-oriented application. The next step in the process was to create mock interactive demos. These were based on a simple hardcoded tabletop application developed quickly to capture the basic design features for the purpose of engaging industrial partners in discussions and providing feedback. The mock demos initially used images instead of actual GIS data and progressed iteratively into builds that actually integrated with sample

general-purpose GIS data from the online sample servers of ESRI™. The later versions of the demos used an application programming interface (API) from ESRI called ArcGIS API for Windows Presentation Foundation (WPF) and Silverlight [ArcGIS2011, WPF2011]. ArcGIS is described on the ESRI website as a complete system for designing and managing solutions through the application of geographic knowledge. These demos were deployed on a Microsoft Surface table [Surface2012]. The demos were interactive in the sense that users could actually touch the table and move the user interface elements around using simple hand gestures. This in turn increased the users interest and, through frequent demos, encouraged them to suggest changes to the interface and functionality of the environment based on how it fits within the environment of the utility company and also based on how easy to use and how intuitive the design is.

Figure 3-6 and Figure 3-7 include screenshots from one of the early demos of eGrid. This demo build was the first integration attempt with ESRI ArcGIS API for WPF. The images on the sides represent locations that are saved for future reference. They are repeated on each of the four sides to be easily accessible regardless of where the user sits around the table. These locations can be trouble report locations. In the first snapshot, the map window in the middle includes some annotations and the color wheel is shown in the toolbar allowing the user to select the color of the drawing strokes. The second snapshot shows different map windows with different map layers for street maps and satellite images.

3.2.4 Iterations of Advanced Prototypes

eGrid is a user-interface-intensive application. It was developed as a multi-touch user interface for a horizontal digital table with no specific orientation. The design of the user interface elements, touch events and hand gestures were done with this target hardware platform in mind. To capture the interactivity and multi-touch capabilities of the application, more advanced prototypes were developed in this phase. The prototypes were developed in multiple iterations guided by frequent demonstrations to industrial partners and other interested GIS professionals. Since the development of this application is an exploratory study, a focus group approach was used. Demonstrating the prototype to different users and allowing them to touch it and try using it motivated the users to provide feedback. User feedback was helpful in identifying potential design problems and possible enhancements which were then addressed in the next version of the prototype. So basically, each prototype version included new features developed and enhancements based on the feedback received from potential users on the previous prototype.

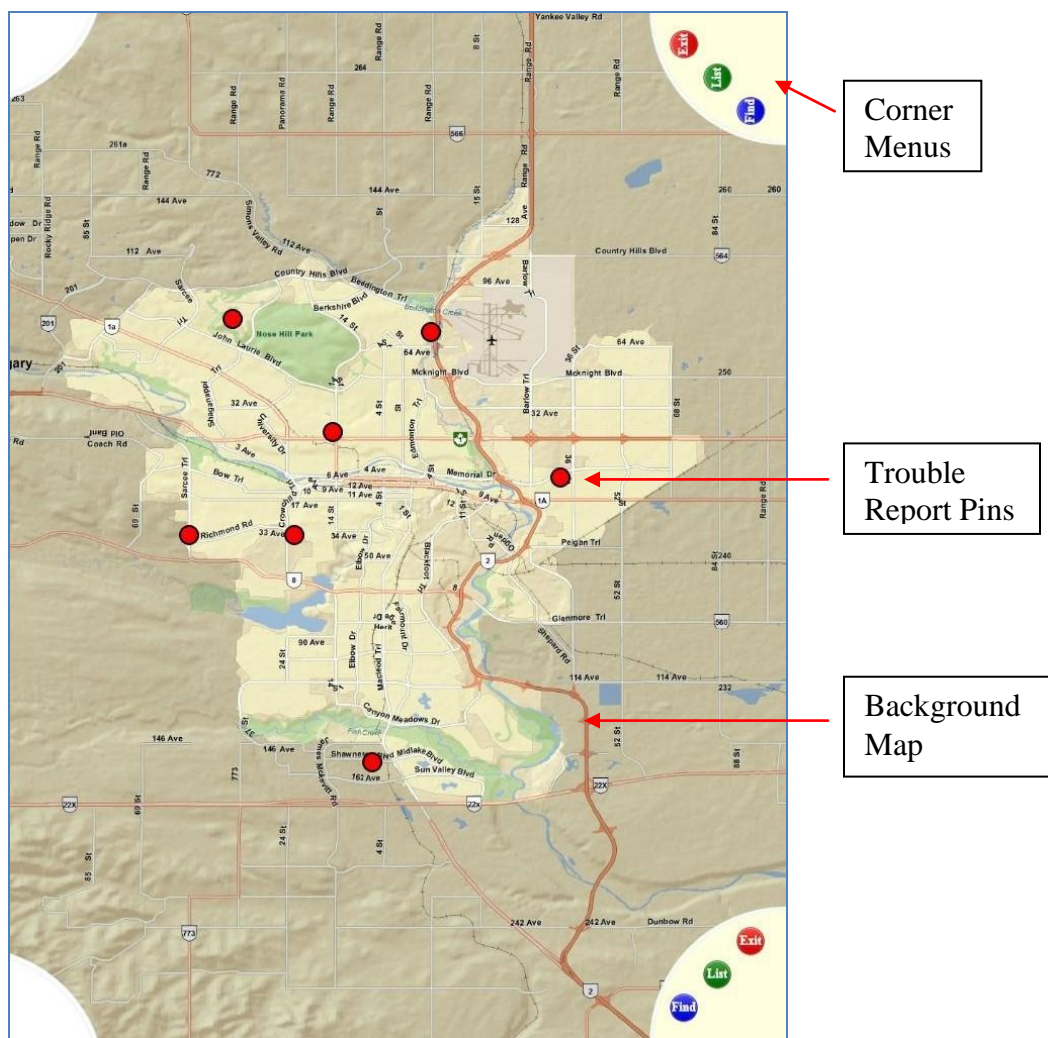


Figure 3-8: Snapshot from an early version of eGrid featuring trouble report pins and the Calgary map background.

Figure 3-8 **Error! Reference source not found.** is a snapshot from one of the prototype versions of eGrid. The background of the application has a GIS map of the city of Calgary. It serves as an overview of the current state of the electrical circuit that promotes awareness between the users in the collaborative environment. The red dots/pins

represent trouble report locations on the map. As mentioned before, trouble reports are service requests received from customers and are supposed to be completed by service crews dispatched by the control center.

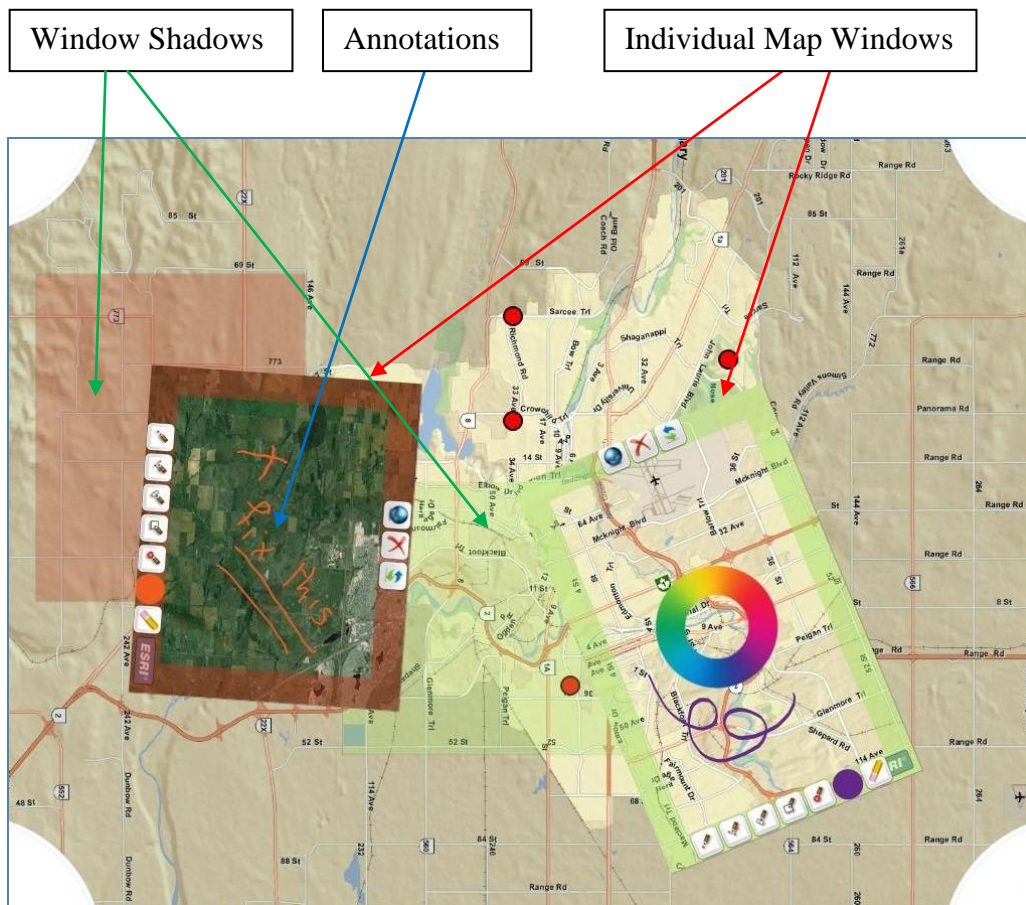


Figure 3-9: Another snapshot of the same version of eGrid featuring different map layers in different frames and annotations added on the maps.

Figure 3-9 shows another snapshot from the same prototype version. This snapshot shows two map windows, each showing a different layer configuration on the map. This is

important to give the user flexibility in modifying the settings inside individual map windows. Annotations are visible on the maps, where the user can choose the color of the annotation strokes using a color wheel. Semi-transparent rectangles matching the colors of the map windows are visible on the background map. These rectangles are called window shadows and they show the user the areas of the background map that currently appear inside of the map windows. More details about the design of eGrid and the reason behind design decisions are included in the design section.

The interface of eGrid and its features have been changing continuously throughout these iterations responding to user feedback to enhance some features and add new suggested features. A timeline of prototype demos of eGrid to electricity company users and users from other relevant industrial domains can be found in the appendix. To explain the prototype and demo iterations, the following list includes an overview of the features created or enhancements made in three of the main prototype versions of eGrid. The prototype versions and included features were not known in the beginning of the design process but were rather a result of the iterative feedback and enhancement process used. More details can be found in the implementation details section.

- Version 1.0:
 - Features:
 - City of Calgary Background map
 - Corner menus for common application functions

- Trouble Report pins saved in a graphics layer on top of the background map.
- A custom gesture for creating a map window composed of two touch points as corners of the rectangle.
- Individual map windows with different colored frames
- Basic annotation tools on the map window frame and the annotations are only saved in a graphics layer not in the database.
- Custom panning and zooming gestures to be used on individual map windows.
- Support for two map layers: Street maps and Satellite images, both are from ESRI online public server.
- Base Technologies:
 - Microsoft Surface SDK
 - ESRI ArcGIS API for WPF and Silverlight
- Version 1.1:
 - Features:
 - Window Shadows for increasing users awareness
 - Lasso gesture for creating new map windows
 - Trouble Report lists accessed from corner menus
 - Map layers list for each map window to change the visibility and transparency of different layers.

- Trouble Report info boxes to access details of trouble reports and open new map windows focused on affected areas.
- Enhancements to the accuracy of map manipulation gestures.
- Other bug fixes
- Base Technologies:
 - Gesture Toolkit for custom gesture definitions
 - Surface Toolkit instead of Surface SDK to potentially support other hardware platforms.
- Version 1.2:
 - Features
 - Trouble report pins are stored in an actual layer in the GIS database.
 - Service Crews layer showing the moving vehicles of the service crews on the background map, also saved in the database.
 - Annotations are actually persisted in a graphics layer in the database and can be retrieved when map windows are created.
 - Global Layer configuration dialog to allow the user to edit the layer configurations, add or delete layers for the entire environment (i.e. to be used by any map window).
 - More layers can be edited through the map window layers list, including trouble reports, service crews and annotations.

- Base Technologies:
 - Support for Evolve table as deployment environment [Evolve]
 - Support Windows 7 build-in touch capabilities.

3.2.5 Observation Sessions and Interviews

At some point afterwards when the design reached a fairly detailed level and new demos produced less user input, a series of observation sessions in the control center of the utility company and interviews with operators and GIS analysts were conducted to understand more about the environment of the control center and to further assess and enhance the design. This section describes the study setup, analysis methods and the insights gained in this study concerning the deployment environment of eGrid. The study results that are related to feedback on the design of eGrid will be described after describing the design of eGrid in a later section.

Two observation sessions were done in the control center of the headquarters of the utility company in Calgary. Each session was two hours long. Following these sessions, seven interviews were conducted with operators and GIS analysts working in the same company. The goals behind the observation sessions and interviews are as follows:

- More Requirements, to gain better understanding of the application domain and the actual environment in which the application will be deployed.

- Iterative Enhancements, to help in the iterative enhancement of eGrid to best suit the environment of the control center and the needs of the operators and to understand which features could be added to eGrid to help better support the users.
- Design Assessment, to assess the user interface design of eGrid, its relevance to the needs of the actual users and the effectiveness of the software engineering methods used in the project.
- Reusability Analysis, to understand the design aspects of eGrid, which may be useful to other multi-touch tabletop GIS applications for similar domains, based on participant experiences in other domains. This will prove helpful in the next chapter describing the second case study.

3.2.5.1 Overview and Study Setup

The study consists of two components: the first is the observation sessions and the second is the interviews. The two observation sessions were done in the control center of the collaborating industrial partner, a utility company in Calgary. Each session was two hours long which included observing the operators working in the control center as well as talking to a few of the operators about details of their job tasks. Following the observation sessions and through the span of one month, seven interviews were conducted with participants from the utility company.

3.2.5.1.1 Participants

Participants of the interviews were Control Center operators, trainers, GIS analysts who create GIS data from microstation format [Microstation], team leaders and managers. The participants were selected such that their work is related to the control center and also based on their availability and volunteer efforts. Table 3-1 includes the specialization of each participant. The biographical information of participants is not listed as per recommendation of the participants on their consent forms.

Table 3-1: Specializations of Interview Participants

Participant	Specialization
P1	Manager of Geomatics and Asset Records, Power Corporation
P2	Microstation and GIS Tech, Power Corporation
P3	Director, System Operations, Power Corporation
P4	Senior Engineer, System Operations, Power Corporation
P5	Supervisor of Control Center, Power Corporation
P6	Operator Training Specialist, Power Corporation
P7	Supervisor, Survey & Land Management, Power Corporation

3.2.5.1.2 Interview Nature and Questions

The interviews were semi-structured interviews. As opposed to structured interviews, semi-structured interviews do not have a formalized limited set of questions. It is a flexible approach in which the interviewer has a framework of themes to be explored but also allows new questions to be brought up as a result of flow of conversation. This type of interviews is less intrusive to the participants and provides

more opportunity for learning. The observation sessions were beneficial in developing relevant and meaningful semi-structured questions to be used in the interviews. Since the interviews contained open-ended questions and discussions could potentially diverge from the original basic themes, it was beneficial to tape-record the interviews and transcript the results for analysis.

Each interview started by giving the participant a brief introduction about eGrid. Some of the participants had already seen a live demo of eGrid and some of them only saw a video demo (which was necessary for stimulating the discussion). All participants described their work experience in relation to GIS and working with geospatial data and applications in general and they talked about their job responsibilities and whether they use GIS data in their daily work. Using a list of suggested questions, the interviews progressed, still allowing the participant to dive into various related topics. The following list includes some of the topics that were discussed in the interviews:

- Current processes used in the control center
- Nature of the environment including the technologies currently being used.
- Level of experience and job responsibilities of the operators working in the control center who are the target users of eGrid as well as the service crews on the field and the GIS analysts responsible for maintaining the map data.
- Examples of typical job operations performed by the operators.
- Map layers that are commonly used in the control center.

- Tasks commonly performed on a GIS system in the control center.
- Collaborative activities done in the control center.
- Ideas on how useful eGrid can be in the work environment of the control center.
- New features that eGrid needs to be functionally more useful to the operators.
- Level of comfort using multi-touch interactions and digital table versus using multiple screens and mice that is currently the case in the control center.
- Preferences on textual data entry methods such as using a regular keyboard, a virtual keyboard or hand writing recognition.
- Other application domains that may benefit from an environment similar to eGrid.

3.2.5.1.3 Analysis Methods

The interviews were voice recorded and then word-by-word transcripts of the interviews were written. The notes of the observation sessions and the interview transcripts were all added to a research web application called Saturate [SaturateApp]. Saturate was used to qualitatively analyze the notes from the observation sessions as well as the transcripts of the interviews by adding comments, and categorizing information and writing memos which included more insights and ideas. The analysis style used was a free analysis style and focused more on grouping participant comments into groups according to the general topics discussed. The results of the analysis can be loosely divided into two categories: (1) domain requirements and (2) design assessment and enhancement ideas. The following section includes details about the environment of the control center and

requirements resulting from this study. The results that are related to design assessment and enhancement ideas are described within the section detailing the design of eGrid.

3.2.5.2 Study Results

3.2.5.2.1 System Users

The call center of the control center is responsible for responding to customer calls, identifying problems and scheduling tasks with customers. The responsibility is then passed to control center operators to handle the trouble reports. This indicates that the actual users of a system like eGrid in the control center are the operators, not the call center employees. Usually two to five operators may be gathered around the table of printed maps discussing and analyzing trouble reports. Their job is to make sure that the service crews are turning on the proper equipment and that these crews are on the correct circuits. For example, sometimes the maps with the service crews are not up to date so the crew thinks they are on one circuit while the control center operators find out that they are on a different circuit. Thus, the control center operators are, as the name implies, in control of what is happening in the field. They give the service crews the authorization to perform certain jobs on the field.

Field workers of the service crews might suffer from the extreme weather conditions, which is why they do not take computers with them outside. They take paper maps that can be thrown away at the end of the day. They have laptops in the trucks and they print

out what they want from the truck before going out in the field. They can only see the default state of the circuit (they name it the ‘normal’ state of the circuit), which is not the up-to-date version in the control center containing all the temporary fixes. The default state is the original design of the circuit without any temporary fixes. Service crews communicate with control center operators concerning the changes they will do and they reference the paper maps during their discussions. The data collected in the interviews suggests that it would be beneficial for the field workers to be able to see the exact versions the operators in the control center see. It would also be useful if the control center operators can do some changes to the electronic maps and send the service crews new up to date versions. However, according to the statements in a couple of interviews, editing features should be a privilege given only to the operators who reside in the control center.

According to one of the participants, control center operators are accustomed to using paper maps, and it would take some time for them to develop trust in any new system such as a tabletop GIS system. They have to trust that when they do changes to the maps, the changes will persist and not get lost. One participant commented: “We all have bad experiences with computer systems so that trust has to be developed. It’s gonna be a generational cultural change.” The change into a new electronic GIS system will be tough for some of the operators in the control center since they will have to go through a learning phase to adjust to the new environment and its capabilities. However, as

explained by this participant, a large percentage of the staff is near retirement in the next three to five years which will create an opportunity for adopting a new technology when younger staff members join their team.

3.2.5.2.2 Current Processes

In the control center, operators still rely on paper maps even though they have software that allows them to access the microstation format maps and the new GIS server in which the data is gradually converted and added. The difference between paper maps and the electronic maps is that the paper maps have hand written annotations describing the decisions taken to solve emergency problems, such as changing the state of various types of electrical equipment. They also use the paper maps to find tie points that are areas that can be covered by alternate power sources. According to a number of participants, the importance of these paper maps is the annotations on them, since these annotations are not currently stored in the GIS database. Whenever the operators are trying to fix a new problem, they have to search for the needed paper maps, find any abnormalities, i.e. changes done to the map to fix other problems. They have to check for previous decisions in order to ensure the safety of the service crew working on site. In the electronic version of the data, they can find the condition of the electric circuit as it exists in its normal state. However, they cannot find the abnormal state of the circuit or the live state, which is more important and it only exists in the annotations they add to the paper maps.

There are two types of trouble reports that the operators are faced with. The first type is handling emergencies such as broken poles and wires. The second type is handling problems that can be delayed or solved over time due to cost and safety issues. The steps they take to solve these problems are to: (1) verify the circuit and check for abnormalities, (2) restore electricity and create a safe situation, (3) fix the source of the problem, (4) and finally restore the changes they have done to the original status.

In the control center, they need a number of different map layers. They mainly use the base circuitry layer including all the electrical circuits and the address and street layer that helps them in identifying the location of trouble reports and service crews. Other layers they need include underground, overhead, conductors, feeders, transformers and duct work and all the circuitry information diagrams as well as the special civil information, all spatially located. In addition to these important layers, the GIS team (as opposed to the operators of the control center) has access to other layers which can be useful, for instance they have layers to show Transportation information (roads, construction boundaries...etc) and Hydrology information (rivers, lakes..etc).

As previously mentioned, control center operators make sure that the service crews are on the correct circuits and they verify the required actions through an electrical verification procedure. Currently, when a crew contacts the control center and requests permission to energize (turn on) a piece of equipment, the operators have to verify that the crew is on

the correct circuit. They achieve this using a software application called Buyers that allows them to search for the electrical device and verify that it is in the correct location. The problem with this software is that it is not always up to date since it depends on map data created nightly rather than the actual data residing on the server, which is one of the reasons they decided to rely on PDF files created and updated manually from their map server until they can access the GIS data directly. PDF files allow them to view the most recent version of the data provided that a new version of the file is created whenever any significant change happens in the data.

One of the participants was a microstation and GIS expert. His role was to make sure the map data is up to date and to provide the operators in the control center with the versions they need. He uses the GIS data to verify changes and to make sure that it correlates to the microstation data on which they rely in most of the cases. At this point in time, the circuit maps which the GIS analyst creates in Microstation format are the most up to date, which is why they rolled all the data from these files into the new GIS. However, differences between the GIS data and the microstation data and discrepancies still exist which require him to continuously compare the two sources of information.

According to this participant, “Some people are more familiar with flipping pages on a magazine rather than reading it online. And it is funny that some of the guys there and not necessarily the younger guys have latched on to the electronic version and love it. They

are looking at the PDF all the time and other guys use the paper. One of the reasons I created the PDFs is that they look like a piece of paper on the computer screen so the guys are comfortable with that.” After watching the demo of eGrid, the participant expressed that it is best to gradually move the operators in the control center to electronic maps. He believes that, at this point in time, if what has been made available to the operators is what he saw in eGrid then that is about what they can handle right now. Given that none of the operators are GIS experts, trying to get them away from paper maps has to follow a gradual process without overloading them with a ton of new features.

3.2.5.2.3 Virtual Job Example

This section describes an example of what the operators would do to handle a trouble report. Trouble reports are created upon requests and complaint received by the control center from customers. The trouble report is assigned to a service crew that contacts the control center to request permission for the changes they need to do, such as energizing (turning on) a particular fuse and a transformer. The first step the control center operator would take is to use the Buyers software and find that piece of equipment such that the crew and the operators are talking about the same thing. The operator would then take out the paper copy of the map from a drawer full of paper maps underneath a big table and write notes on it including the instructions which were done on the circuit, which constitute its abnormal state. The operator would also add the time and date of the

changes, the crew which performed these changes and the trouble report identifier. He would then highlight the notes written on the paper using a regular marker to make them stand out. Then he would file that back into the cabinet which holds all the paper maps. After the problem is fixed, the crew contacts the control center to inform them about the status of the trouble report. The operators go back to the paper maps and scratch or erase the annotations representing the temporary changes done before. If some changes are permanent, the operator would then try to find the map online or take a snapshot of Buyers and send the changes to the microstation analyst who is responsible for changing the master copy of the map to include the changes done.

3.2.5.2.4 Motivation for using eGrid

This section describes some insights on how eGrid fits into the environment of the control center and the motivation behind using it. The motivation behind developing a system like eGrid resulted from a number of different problems which are encountered in the current technologies used in the control center. As mentioned before, the operators in the control center sometimes use a software product called Buyers. It is designed to view the output of a GIS system converted into microstation format. The Microstation format is used to create the drawings/maps. And in the control center, the paper maps are printed from the microstation maps. The operators also make use of quarter section circuit maps in a PDF format which can be searched. By saving the maps into PDF, they can search in them as a transitional step between using the paper maps and using the GIS system with

ESRI products. The PDF files are more up to date than the maps in the Buyers software since they are printed from the microstation maps that are updated more frequently than the maps on the GIS server. Two factors are holding back the transition into using GIS in the control center. The first factor is the comfort level of the operators working with GIS. The second factor is that they are still experiencing performance issues with using ArcViewer, which is a GIS viewer application from the ESRI suite of products.

Paper Maps

Operators in the control center add annotations on the paper maps, which are comments written on the paper maps to highlight some areas or specify details about trouble reports such as service crews, referenced equipment IDs. These annotations are so important to the operation of the control center but they are not persisted in the GIS system on the server. Writing the notes on paper maps poses the risk of losing these critical notes if anything happens to the paper copies. In addition, as soon as Microstation files are changed, fresh paper copies have to be printed and new PDF files have to be created. Then the operators in the control center have to transfer any notes that they had on the old paper maps onto the updated copies.

A system like eGrid will help solve this problem since it mimics the environment of using the paper maps but at the same time allows the users to add their annotations to special layers in the GIS system. The operators will take some time to adapt to the new system but it will be useful in the long run for disaster recovery since all the data will be

stored on the server rather than on paper and backup copies of the data will be stored in GIS.

ArcGIS Software

The operators in the control center are not comfortable yet using ArcGIS software [ArcGIS 2011] to access the GIS data on the server. According to the participants of the study, not everyone in the company, and even in the control center, has full privileges for viewing and updating all the data in ArcGIS. Since they are still migrating into the use of ArcGIS as an up-to-date technology, some participants complained that the data sometimes has discrepancies and missing information. By using a system like eGrid, the users may be more comfortable using GIS data since the interface is much simpler than regular full-featured GIS viewers designed for GIS analysts. In addition, since the operators of the control center are only interested in a few map layers like the circuitry layer and the street layer, these layers can be published into an online server, and accessing them from eGrid without the extra un-needed data sets could potentially enhance performance and motivate the users to use GIS data even more.

Buyers Viewer

The Buyers software does not support showing layers since it only shows the microstation data, i.e. the electricity circuits. Using a GIS environment like eGrid will allow the operators to overlay different map layers and potentially get better analysis

results. Buyers also does not possess any annotation capabilities which is why they still have to either add annotations to the PDF maps or print the paper maps to add the annotations on top of them. Operators usually prefer using paper maps and none of them adds annotations to the PDF copies. A system like eGrid is more beneficial in this case since it allows the users to add notes to trouble reports and annotations on the map.

3.2.5.2.5 Relevant Application Domains

eGrid is an environment for interacting with GIS data on a digital table and GIS has many applications in different domains. As part of the interviews, participants were asked about other domains that can benefit from similar applications to eGrid. The answers to this question were speculations as opposed to actual industrial experiences. One of the participants explained that it can be used in “any sort of situation that you want to manage a bunch of physical assets which are geographically placed”. Some participants stated that it can be used in the domains of transportation, traffic, facilities, utilities, water, road system, oil and gas, emergency response systems and so on. One of the participants mentioned that a system like eGrid is not only beneficial for the control center but also for their trouble department which handles crews that go out to respond to trouble calls and do inspections for the control center. In this department, the crews get together each day in the morning and then each crew proceeds to its destination. They would use a system like eGrid in a similar fashion to how the control center would use it. One of the participants stated that eGrid or the library extracted from it can be used in

“any other department in the company which use switching orders, inspections and pre-planned fixes”. He added that “especially when creating switching orders, they can make use of the big screen and the ability to see where the individual map relates to the whole map”.

3.3 Technologies Used in eGrid

The early demos and prototypes of eGrid were developed for the Microsoft Surface® table, which supports a large number of concurrent touch points. Figure 3-10 shows a version of eGrid running on the Microsoft Surface table. Microsoft Surface provides an SDK for developers to build their applications on top of it. It simplifies the touch-related development complexities like managing concurrent touch points and creating touch enabled widgets.



Figure 3-10: eGrid on the Microsoft Surface Table.

However, the Microsoft Surface table also has a number of disadvantages. First, it has a small surface area, the size of a coffee table, which may not be adequate to view very large maps. And secondly, the height of the table is also similar to the height of a coffee table. This makes it necessary for users to be seated around the table and bending over the table sometimes. It is not high enough for people to stand around it and work comfortably which makes it less practical for the control center environment. These issues pose some limitations on the collaboration of operators in the control center who are used to stand around a much larger regular table to collaborate around paper maps. eGrid needs to be deployed on a larger and higher table to comfortably support the collaboration of multiple users. Another disadvantage of this specific version of the Microsoft table is that it supports only WPF 3.5 and not the newer .NET 4.0 version that runs on Windows 7 platform. This implies that Windows 7 touch capabilities cannot be used on this version of the Microsoft table, and that any version of the software using .NET 4.0 will not run on this table. Another disadvantage is that, any version of the application using the Microsoft Surface SDK will not run on any other tabletop hardware.

To be able to solve some of these issues, the following version of the prototype was built on top of the Gesture Toolkit [Khandkar2010, Khandkar2010b], which supports a domain-specific language that defines multi-touch interactions, thus hiding the complexities of low-level implementation from the implementation of the application and

allowing the same code base to be deployed on different platforms including any Windows 7 enabled touch device. Being portable and flexible enough to be deployed on different multi-touch devices is important since the exact deployment environment in the control center of the utility company is not defined yet. The use of Windows 7 along with .NET framework 4.0 allowed eGrid to be deployed on any multi-touch screen or digital table supporting the Windows 7 operating system. The latest version uses the touch events provided by Windows 7 along with ArcGIS API for WPF 4.0. It is also built on top of the Microsoft Surface Toolkit for Windows Touch Beta, which is a set of reusable controls and APIs which can be used with the .NET Framework 4.0, and Windows Presentation Framework 4.0 (WPF). The toolkit can be used to implement multi-touch applications on Windows 7, and some of the widgets in the Microsoft Surface SDK are available in this toolkit, such as the scatter view item widget used to create the map windows. It is deployed on an Evoluce® table [Evoluce], which is an interactive 47-inch Multi-touch Full HD LCD Screen with multi-touch and multi-pen functionality. This table is significantly larger than the Microsoft Surface table and is more suitable for deployment in the control center of the electricity company. In addition, the underlying operating system is Windows 7. This makes this version of the system portable to any Windows 7 enabled multi-touch device. Figure 3-11 shows eGrid deployed on the Evoluce table.

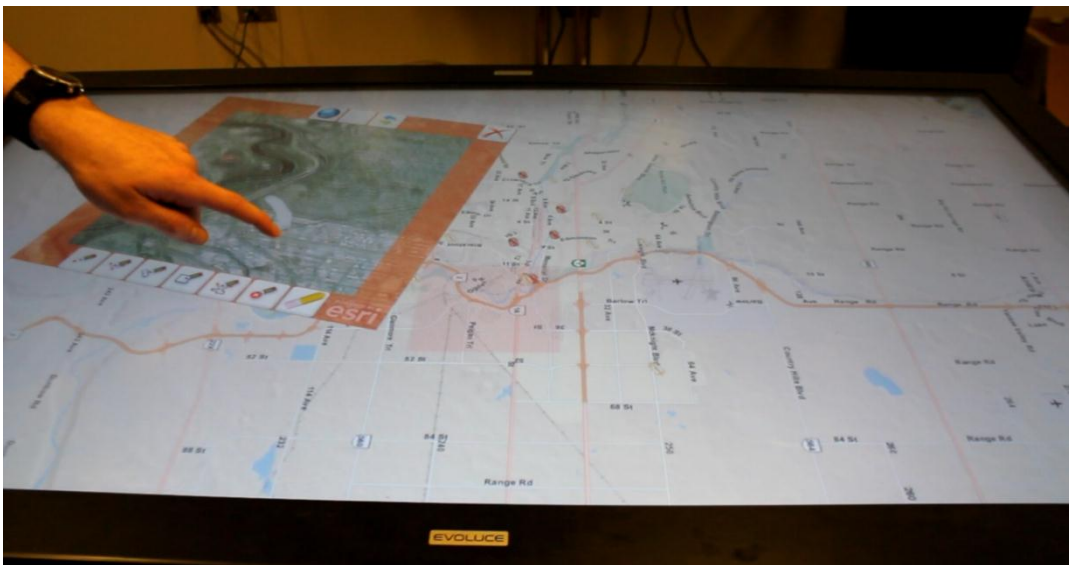


Figure 3-11: eGrid on the Evoluce Table.

3.4 Design Details

This section describes the final design of eGrid and explains the reasoning behind different design decisions. One of the bigger challenges for the tabletop community is the creation of new and innovative applications utilizing these new technologies. As interesting and innovative as these technologies are, until they are used to create practical applications solving actual real life problems, they are not that useful. Technological advances are just a “means to an end” [Buxton2010] and the purpose of having them is to actually use them in applications which further extend the boundaries of human and organizational capabilities. [Hevner2004] Therefore, the goal of the design process of eGrid was to produce a solution that is:

- Functional: satisfying all the functionalities that are needed by the control center operators in their day-to-day tasks.
- Practical: such that it can be favoured by the users over their other traditional practical methods and such that it can be used in their everyday operation and not on occasions.
- Simple: by adding only the features that are needed by the users rather than adding a lot of extra features which are nice to have but may be distracting.
- Flexible: allowing users to work individually or collaboratively in groups and not forcing users to adopt certain problem solving styles.
- Easy to learn: by using interactions which the users are most familiar with and which require little to no learning especially that most of the operators are not GIS experts.

The purpose of focusing on these qualities is to try to make this environment attractive enough to be used in the actual day-to-day operation of the control center thus becoming a solution to the problems faced currently by the users. As mentioned previously, eGrid is developed as a multi-touch user interface for a horizontal digital table with no specific orientation. User interface widgets and hand gestures were selected with this target hardware platform in mind. Users have participated actively in specifying some of the design details through the feedback received throughout the design iterations and some design decisions were inspired by ideas in other interactive GIS applications as discussed

in the literature review chapter. The following sub-sections describe the main components of the user interface of eGrid and the reason behind certain design decisions

3.4.1 User Interface Elements

3.4.1.1 Background Map

The background of eGrid is a GIS map of the city of Calgary. The purpose of having this map in the background is to help users keep track of trouble report locations and provide an overview of the state of the electrical grid of the city. It also works with the Window Shadows, which will be described later, to help users keep track of all of the opened map windows. Panning and zooming gestures are disabled on the background map to avoid confusion and maintain the coordinates of the map pointing to the working area.

3.4.1.2 Corner Menus

Corner menus are quarter pie menus, one on each of the corners of the screen. They allow users to access common application tasks which are not related to any specific map window. They were chosen instead of conventional drop down menus to achieve a number of benefits. Regular drop down menus were designed for vertical screen applications. If they were used on a tabletop application, the orientation becomes an issue and users would not be able to approach the table and use the application comfortably from any angle. Thus, the design of the menus would have to be different to accommodate for the nature of the hardware being used. One of the options that could be

used is the use of rotatable circular menus that are made visible on demand on any location of the screen. The design of choice, i.e. quarter pie corner menus, consumes a small area of the screen and does not get in the way of using the map windows. It is also accessible easily by any user no matter where the user is standing around the table. When the user taps on any of the corner menus, it expands to a slightly larger quarter circle menu showing a group of icons for different functionalities, such as browsing the trouble report list, editing layer configurations, or simply exiting the application. If more buttons need to be added to the menu, the design can be changed to accommodate rotating the menu to show the extra options on a central pivot point. A similar design is described in the context of stacked half pie menus by Hesselmann et al. [Hesselmann2009].

3.4.1.3 Map Windows

Having the ability to create multiple smaller map windows on top of the background map allow the users the flexibility to work on multiple different maps at the same time or work on one big map. This design was motivated by the fact that it mimics the nature of being able to work on multiple printed maps at the same time on an actual table. In addition, collaborating teams often toggle between different modes of collaboration, and this design is flexible in that regard and avoids imposing any specific mode on the users. For example, subgroups can work on different problems and users can have different perspectives on the same data visible in different windows at the same time. For achieving this flexibility, map windows are designed. Each map window has its own

layers settings and a different color for the semi-transparent map window frame. The idea of a map window is somehow similar to the idea of a MagicLens introduced by Bier et al. [Bier1993] An implementation of the MagicLens idea in context of geospatial data analysis was done in DTMap [Furuichi2005], an application built on top of the Diamond Touch table [Esenther2002]. Figure 3-12 shows a snapshot of DTMap and how the user can change the zoom level of the data inside the MagicLens by stretching it.

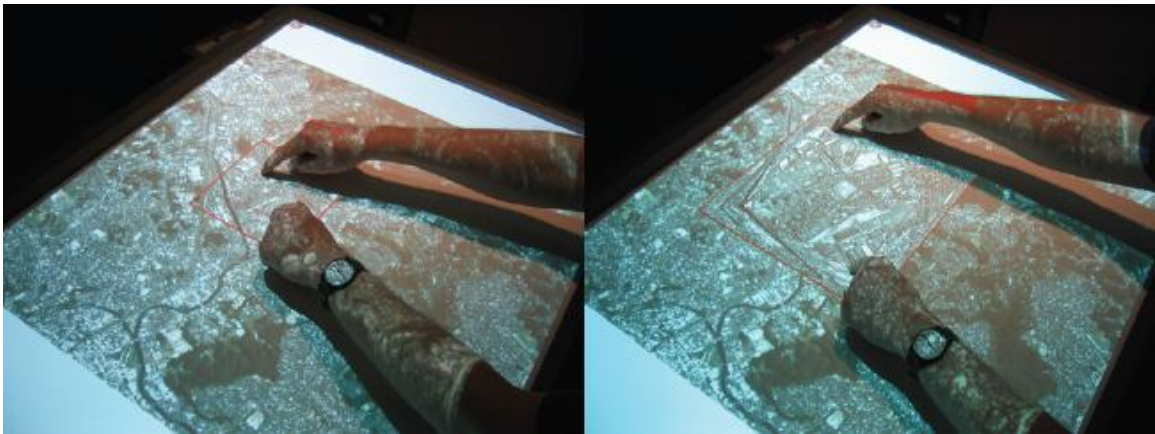


Figure 3-12: A snapshot for DTMap showing how the user changes the zoom level of the data inside the MagicLens using a stretch gesture. [Forlines2005]

In eGrid, the design is different since the underlying hardware architecture is more advanced. In contrast to the user of MagicLens in DTMap, a map window in eGrid can be used by more than one user, the orientation and scale of the window can be changed separately from changing the characteristics of the contents of the map itself and the

frame of the map window has annotation tools which can be used on the map, as will be discussed in another section shortly. In DTMap, the MagicLens can be locked after defining the zoom level allowing the user to use annotation, resizing and magnification tools as in Figure 3-13.



Figure 3-13: Snapshot for a MagicLens from DTMap showing the different tools available when the lens is locked [Forlines2005].

A map window can be created in a number of different ways. The first option attempted is to use a gesture composed of two consecutive finger touches that define the corners of the rectangular part of the background map that will appear in the map window. This gesture is easy to use but it turned out to be not intuitive. Based on the feedback from

different users in demos, users were not able to predict that this gesture is supposed to be used to create the map window. A description had to be given to the users in order for them to use this gesture. A better option was used instead: a lasso gesture. The lasso gesture is a closed circle or rectangle drawn by the user on the background map to encompass the area of focus. This gesture received more positive feedback from the users and they were able to use it without requiring any preliminary description since it resembles a common pointing gesture. Map windows can also be created by opening a specific trouble report either from the trouble report info boxes or from the list of trouble reports which can be accessed from corner panels; both of these user interface elements will be described in following sections.

Figure 3-14 presents a snapshot of eGrid featuring the background map and two map windows each having its own map layer settings. Inside one of the map windows, a color wheel and some annotations are displayed. Two trouble report list windows are also visible on the background map plus the trouble report pins and service crews.

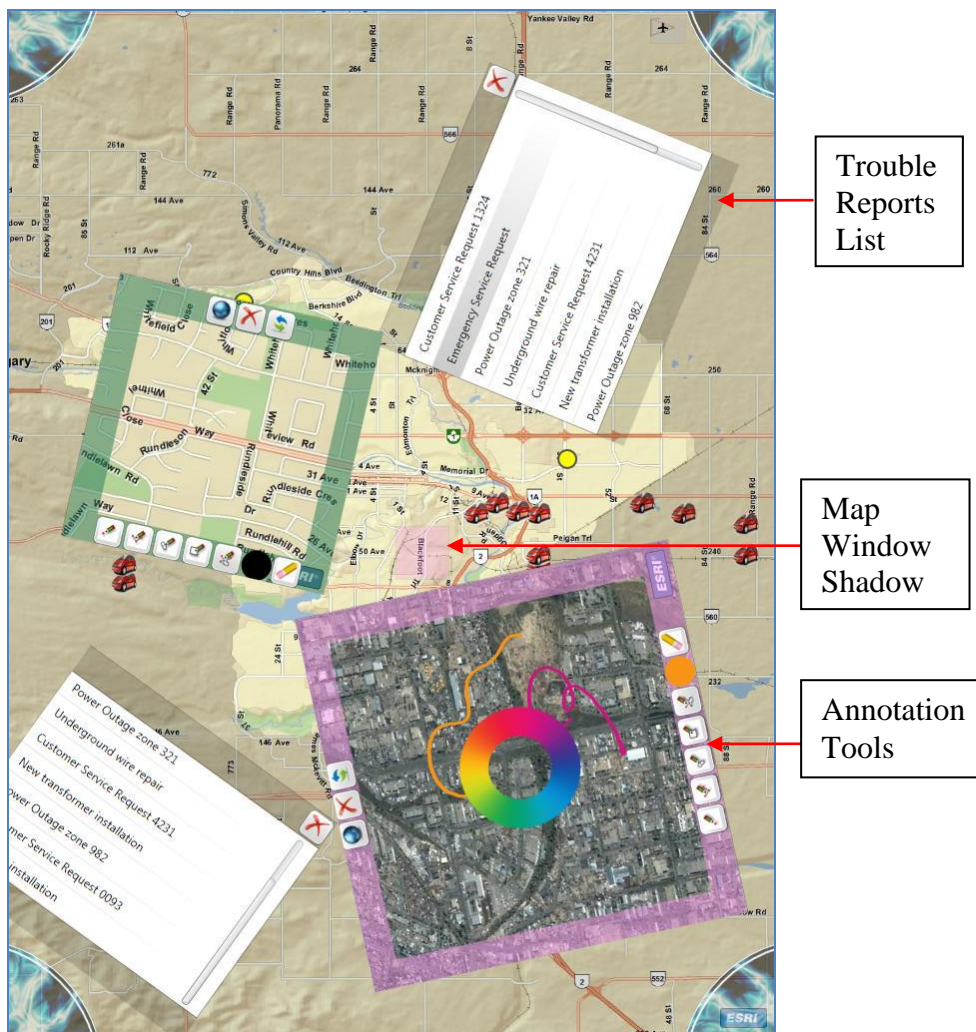


Figure 3-14: Snapshot 1 showing parts of the design of eGrid.

3.4.1.4 Window Shadows

Allowing users to create multiple map windows at the same time gives them the freedom and flexibility according to their problem solving style. However, having too many map windows can confuse the users as to which map window shows which area on the map.

One option to solve this problem is to describe the area shown inside the map window in

a map tip but this solution gets complicated since the design of the map window allows users to resize them to any size and zoom in and out to any area freely. Getting information about the area displayed inside the map window from the database can be tricky in this case and finding out the relationship between different opened map windows is not straight forward. Another solution is to display on the background map indicators of the location of each opened map window to increase user awareness of the location of all the map windows with respect to the background map and in relation to each other. This solution was achieved through the design of Window Shadows. Each map window has a semi-transparent frame and a map area in the center. To help the users keep track of the opened map windows and the location of the data with respect to the background map, each map window is created such that the frame has a different color and shadows for all opened map windows are displayed on top of the background map. Each shadow for a map window is a semi-transparent rectangle that covers the area of the background map that appears inside of its corresponding map window. The color of the window shadow is the same as the color of the frame of the corresponding map window. Thus the link between the shadow and the actual map window is the color.

This design increases user awareness of the locations inside all of the map windows opened and the relationships between the map windows. For example, if the areas inside two map windows overlap, the shadows will overlap as well. If the user pans the map inside a specific map window, the corresponding shadow will move on top of the

background map as the user pans the map. If the user changes the scale of the map inside a map window by panning and zooming, the size of the corresponding window shadow will change depending on user changes to cover more or less of the background map. Furthermore, users are allowed to drag the window shadows on the background map as well and the area inside of the corresponding map window changes accordingly to display the new location. This feature is very handy when the user wants to pan to a location which is far away from the current location in the map window since moving the shadow on the background map until it reaches the new location is easier and less confusing than panning the map inside the map window.

3.4.1.5 Annotation Tools

Operators in the control center usually write their notes on top of the paper maps and these notes are important since they represent the current state of the electrical circuit after changes done by the service crews. Thus, allowing the users to annotate the maps and saving/persisting the annotations to be accessed later on is a key requirement of the system. Rather than giving users the ability to write annotations on the background map which has an inappropriate scale for annotations, allowing them to write inside of map windows seemed like the better choice. Annotation tools were added to the frame of each individual map window and they determine the state of editing inside the window. And for the purpose of allowing users to work together without requiring one user to edit at a time, each map window had to have its own tools rather than creating one set of tools to

be used on one map window at a time. The tools allow the user to select the color of the annotations from a color wheel, change the mode of interaction with the map from browsing to editing and vice versa. In the editing mode users can add annotations to the map using finger touch or pen/stylus if the technology of the table supports pens. Users are allowed to draw freehand, draw points, straight lines, rectangles or polygons on the map. The user is also able to erase the annotations. The changes are saved instantly in a graphics layer in the database and can be seen in all map windows as well as on the background map if the graphics layer is changed to be visible from the global layer configuration manager, which will be described in a following section.

Each opened map window maintains its own state and mode of interaction, which means that the annotating and editing tools are local to the map window and not to eGrid as an environment. The drawback of this design is that if two users are both working on the same map window, they will have to use one tool at a time, for example they can both write or both erase at the same time. Users can annotate the map inside a specific map window at the same time but they have to use the same editing tool since the current tool setting is maintained in the state of the window itself. Even though this design is restrictive in terms of type of concurrent annotations, the advantage is to minimize having conflicting actions by concurrent users. In the environment of the control center, this is not an important issue since the changes are usually critical and users usually take their time and put some thought into the changes they wish to do before starting to edit.

3.4.1.6 Map Window Layers List

To allow users to have separate work spaces represented in different map windows, the design should enable different map windows to have separate settings. One of these settings is the set of map layers visible inside the map window. Each map window should be able to have a different set of layers visible. To achieve this requirement, the frame of the map window has a button that allows the user to change the map layer settings for this specific map window. The control is a scrollable list of layers with checkboxes to control the visibility of each layer and sliders to control the transparency of the layers. The layers setting of each map window is local to the map window itself such that each instance can show different layers. Figure 3-15 displays a snapshot of eGrid, in which the layers list appears inside one of the map windows and two different map windows display different layer settings. The default layers in this control are derived from the environment layers settings for the whole application. In the prototype of eGrid, this list includes some GIS map services published by ESRI on their sample servers, including street maps, satellite images, and shaded relief map services. In addition to those layers, the list includes the trouble reports layer, the service crew vehicles layer and the annotations layer. If the user changes the global layer settings of eGrid, using a dialog which will be described in a later section, any new map window will have the new layer settings, while the already opened map windows will preserve their previous layer settings. For example, in Figure 3-15, the layers control box has an extra layer displayed at the end of the list. The name

of the new layer is NewMapLayer and it was added to the environment layers list through the Environment Configuration Manager using the URL of this published map service.

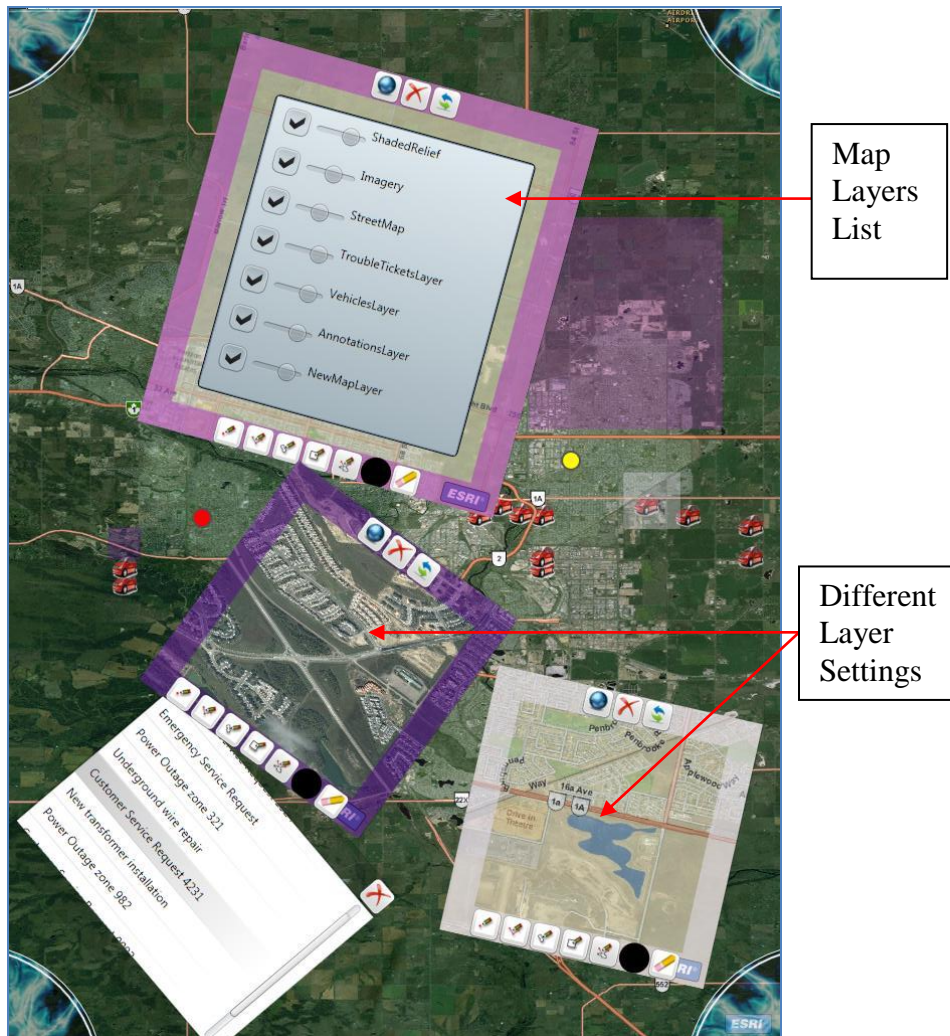


Figure 3-15: Snapshot 2 showing parts of the design of eGrid.

3.4.1.7 Environment Configuration Manager

The default layers displayed in newly created map windows is retrieved from a layers environment setting and an interface was created to allow users to change this setting by

adding or removing accessible layers. This interface is called the Environment Configuration Manager, which can be accessed from any corner menu. It allows the user to change the visibility of the different layers that is immediately reflected on the background map settings. The user can also delete layers from the list and add new layers to the set of available layers by naming the layer and specifying the URL of the published map service. Having this configuration manager causes the environment of eGrid to be very flexible and the users can migrate to new map services without changing the code base. In addition, the users are able to view map services that are published over the Internet from any source, such as government published map services and any data published by GIS providers. Suggested by the participants of the interviews as a future enhancement, the configuration manager could also allow adding secured map services which require user credentials, i.e. username and password. The already added layers such as the trouble reports layer and the service crew layer will be editable as well.

Figure 3-16 features the environment configuration manager widget. The new layer added in this dialog is displayed on the background map. It highlights the major highways on the map. Any new map window opened after the settings were changed in the configuration manager will use the new settings.

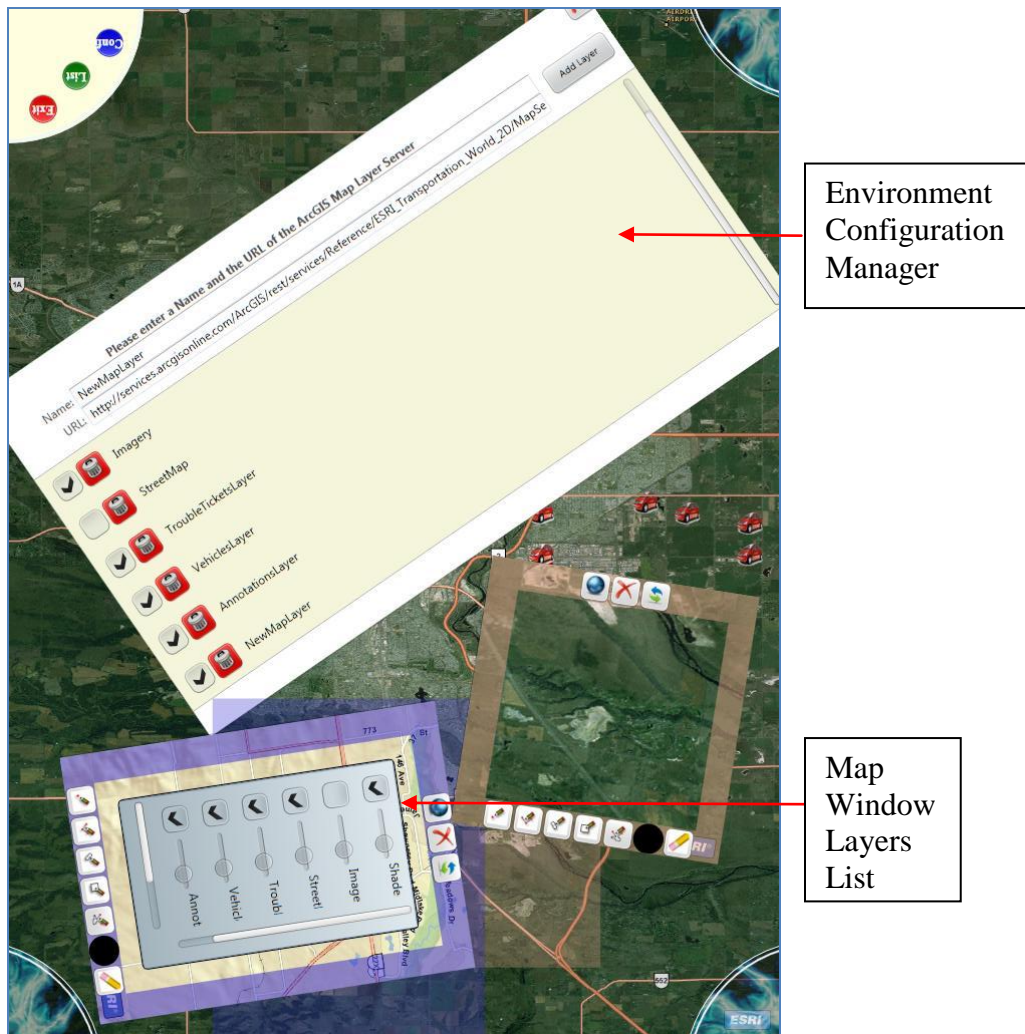


Figure 3-16: Snapshot 3 showing parts of the design of eGrid.

3.4.1.8 Trouble Report Pins

In the control center of the electricity company, they receive trouble reports that define service outage locations and the details of the outage situation. Trouble reports are geo-referenced which means that each report has an exact map location. They also have different severities and/or priorities similar to IT service requests. The design of eGrid

represents these reports as circular pins/dots placed in their locations on the background map and having different colors for different severities and/or priorities. On the background map of Figure 3-14 and Figure 3-15, trouble report pins are scattered in their correct locations. The data of these trouble reports is stored on a GIS layer on a local server in our lab. On the event of deployment of the application inside of the electricity company, the application should be linked to a GIS layer on their server having all the information associated with Trouble Reports, a project that is still under development in the electricity company. A similar situation should happen with the service crew vehicles layer displaying the vehicles of service crews on the map.

Placing trouble report pins on the background map was chosen to provide a general view of all of the existing problems and the relationships between these problems, which can potentially help the decision making process in the control center. The list of trouble reports is also displayed in a list box for easy access to specific trouble reports but this option does not provide the same overall view of the state of the electricity grid.

3.4.1.9 Trouble Report Info Boxes

Based on the information provided by the operators in the control center during the observation sessions, a database table exists containing all the trouble reports handled by the control center, the status of these reports, the crews working on the fix and so on. Operators sometimes need to check this table to view more information about specific

trouble reports especially if they are related to or nearby the location of a current trouble report they are working on. Since it would be beneficial for the user to display the information associated with any trouble report, trouble report info boxes were designed to view this information by simply tapping the pins on the background map. In eGrid, touching any of the trouble report pins displays a small window with more details about this trouble report such as the trouble report number, severity, location and description. The information window is displayed such that the center point of the window is the actual location of the trouble report pin on the map and the window can be rotated around this center point using a clockwise or anti-clockwise single touch rotation gesture to accommodate different user locations around the table.

The trouble report info box has a viewing mode, which is the default mode and it also has an editing mode. By pressing on the “Edit” button, the user can edit the text in the fields using a regular keyboard. The severity can be edited using a slider. Based upon the feedback received from customer demos, a regular keyboard is preferred over a virtual keyboard. Users tend to prefer the haptic feedback they get from using a regular keyboard and it allows them to make fewer mistakes. Based on the interview results, using a stylus or pen to edit the information in the trouble report box can also be useful for some users, especially users who are accustomed to using touch enabled cell phones and tablets. This feature can be implemented in the future as well as hand writing recognition.

A map window showing the exact location associated with the trouble report pin can be opened by touching a button in the info box labelled “Show Map”. The orientation of the newly created map window is set such that the text on the map is easily read by the user who opened the map window, using the orientation of the info box itself as a guide.

3.4.1.10 Trouble Report List Windows

Users from the electricity company suggested that it would be useful to see all the trouble reports in a list. This option can make it easier to browse through a long sorted list and search for a specific trouble report number if the user does not know where this trouble report exists on the map. In eGrid, a window can be opened from any of the corner menus showing a scrollable list of trouble reports. Touching any of the items in the list creates a new map window pointing to the location of the trouble report touched.

3.4.1.11 Service Crews Layer

One of the pieces of information that is useful in analyzing trouble reports and taking appropriate actions to fix them is the location of the service crew handling this trouble report. Currently in the control center, service crew vehicle locations are saved in a map layer and updated from GPS signals received from the vehicles on the road. Operators currently do not use this map layer to check the location of the service crew. Instead, they usually call the crews on the phone to get updates on their location and the current situation. Showing these vehicles on the background map may be beneficial in helping

the operators determine the location of the vehicles in a glance. In eGrid, the vehicles move on the background map according to the updates in the database which will be linked in the future to the coordinates received from the GPS devices in those vehicles.

3.4.2 Interactions and Multi-touch support

This section describes the gestures that are used in eGrid to manipulate the map and other widgets using touch interactions. Since eGrid is built on top of Windows 7, any of the user interface elements which accept taps as input would also work automatically using mouse clicks or stylus, such as trouble report lists, trouble report info boxes, environment configuration manager...etc. Multi-touch interactions such as map manipulation gestures can also be done using multiple mice if necessary for testing in an environment that does not accept touch as input. However, the environment was designed to be used on a multi-touch device, in which case using the environment is much easier.

3.4.2.1 Creating Map Windows

To create a map window showing a certain area from the background map, a couple of gestures were attempted and tested informally by different users during prototype demos.

- **Rectangle Corners Gesture:** Assume that there is a virtual bounding rectangle surrounding the area of the map that will be shown in the map window. The first attempt was a gesture composed of two consecutive touch points where the two touch points represent the two corners of the bounding rectangle. This was a

simple gesture to implement from scratch, yet it did not seem to be intuitive because the gesture had to be described to the user before attempting to use it. The users were never able to predict that this is the gesture they need to use to create the map window.

- Lasso Gesture: The second gesture attempt was made upon the use of the Gesture Toolkit that defines a number of simple gestures to be utilized by any application. The gesture chosen was the lasso gesture. A lot of traditional GIS viewers allow the users to select features on the map using the concept of a lasso or a rubber band. Most GIS users are familiar with this selection technique that allows them to circle an area to be selected by clicking the mouse button and dragging it to draw a rectangle or any other shape surrounding the area of interest. Therefore, the lasso gesture was used in eGrid to allow the user to draw a closed boundary/shape surrounding the area the user wishes to see inside the newly created map window. The lasso shape can either be a freeform closed shape or a rectangle and the area inside of the shape will appear inside of the map window.

3.4.2.2 Manipulating the Map Window

The design of the gestures used to manipulate the map window or navigate the map inside the frame is based on the work of Wobbrock et al in their paper about user defined gestures for surface computing [Wobbrock2009]. Figure 3-17 includes representations for some of the gestures defined by users in the study by Wobbrock et al and used in the

design of eGrid. All of the participants of the interviews for eGrid have agreed that these gestures have recently become very familiar and intuitive to a large number of users since they are also used in many of the multi-touch hand-held devices, such as the iPhone and iPad.

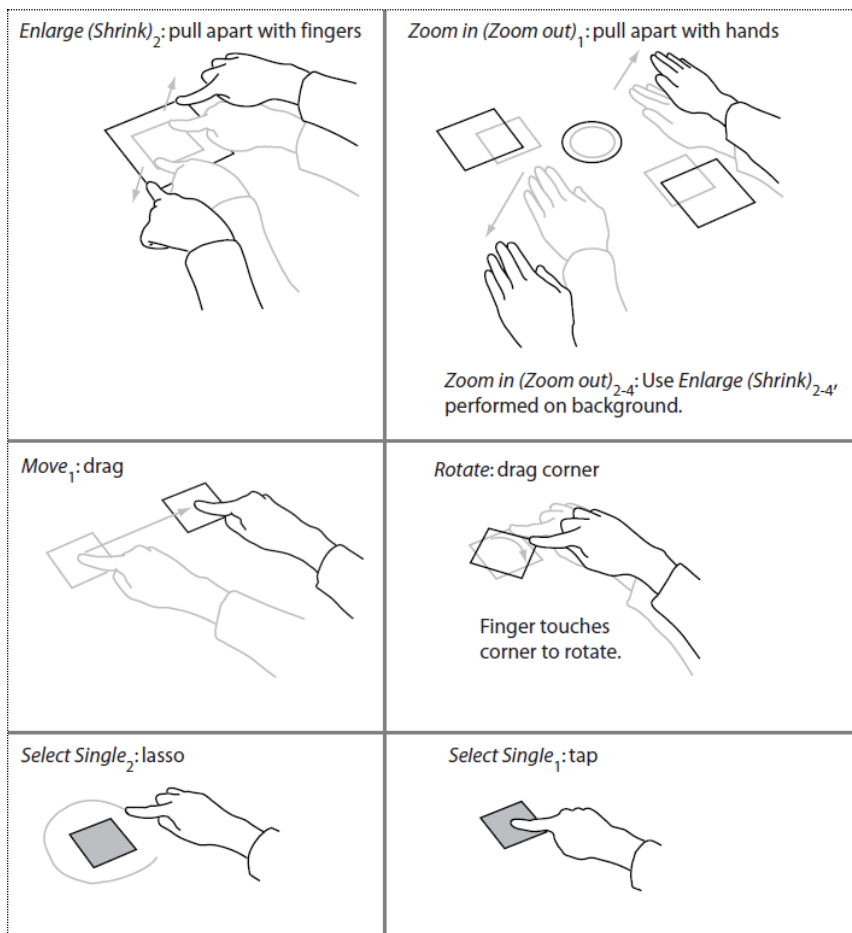


Figure 3-17: Representations for the gestures defined by the users in the study of Wobbrock et al [Wobbrock2009].

Users can change the location of the map window on the table, the size of the map windows and the orientation using touch interactions and simple hand gestures as follows:

- Resizing the map window can be done using two touch points on the map window frame and moving the touch points towards each other for decreasing the size or away from each other to increase the size.
- Rotating the map window can be done using a single touch point on the frame and moving it clockwise or anti-clockwise.
- Touching the frame and dragging it to a different location on the screen can change the position of the map window.

3.4.2.3 Panning and Zooming the Map

Inside the map window, the map itself can be navigated using hand gestures as follows:

- Panning the map can be done using a simple drag and release gesture with one hand
- Zooming in and out of the map can be done using a pinch gesture.

These gestures have become commonly understood by a lot of people since their introduction in some devices such as the iPhone® and iPad®.

3.4.2.4 User Feedback Regarding Gesture Interactions

Based on the results of the observation sessions and interviews, when asked about the use of multi-touch gestures in interacting with maps, all of the participants agreed that touch is really intuitive and that people are getting used to it more quickly especially with the advent of new touch enabled tablets and screens. Some of the participants even mentioned that some gestures, such as those used to pan and zoom the maps in eGrid, can be as easy if not more intuitive than using a mouse and keyboard. Using multi-touch gestures may not be easy if the user intends to edit the GIS map, however, generally for the control room environment, they do not need any advanced GIS functionality. They only need to navigate the maps, switch between the layers, add simple annotations, search for some basic features and so on. They usually do not perform complicated spatial queries, manipulate the geodatabase or try to create new GIS layers.

3.4.3 Enhancement Ideas

3.4.3.1 Displaying Crew Information

In the control center, the service crew information is stored in a table, which is updated according to communication between the operators and the crews. Typically a crew is a team of three individuals identified by the name of the team leader. The table has columns for storing the crew ID, the status of the crew, the circuit they are working on, their location, their function that describes the changes they will do, and more remarks such as fuse numbers. As described before, eGrid displays the position of the crew

vehicles moving on top of the map. One of the participants of the interviews suggested that it would be nice to be able to touch the vehicle symbol on the map and be able to see the current status of the crew and details of the associated trouble report that they are working on. Similarly, an identify tool can be added such that touching GIS features on the map retrieves and displays the associated attributes from the database. This is a very common and a basic tool available in most GIS viewers.

3.4.3.2 Creating and Editing Features

eGrid has most of the visualization features which are needed by the control center, as one of the participants explained. What makes a GIS more useful is that it is not just for visualization but also for editing and adding new features. Typically control center operators are not responsible for editing the maps except for adding annotations. As an enhancement, one of the participants suggested to explicitly support editing and adding new map features, which can be useful for users who are more GIS oriented.

3.4.3.3 Searching Through Annotations

As one of the participants explained "Searching is a big feature...Since they already have their maps subdivided into rectangles, they can search easily through them." The operators in the control center need to search frequently within their annotations for specific reference numbers and crew names. Currently, operators are going through all the maps manually until they find what they are looking for. They cannot perform any

type of automated search through their annotations since they are actually hand written notes on the paper maps, which is why it would be very beneficial to enable the users to search through the annotations, especially by reference numbers.

3.4.3.4 Searching for Equipment by Type or ID

One of the most important enhancement ideas discovered through this study is the ability to search the maps for equipment by specifying the Type or ID of the equipment. It is the main reason why operators use the Buyer viewer. For example, operators may search for fuses to identify circuits and feeders. In addition, address searching seems like a very beneficial feature to help in identifying the locations of trouble reports.

3.4.3.5 Spatial Analysis

Spatial Analysis features can also be useful in a GIS tabletop environment. Spatial queries are typical GIS features. Example questions are how many power poles exist in a certain buffer area, or how many customers are affected in this community. The user may need to do some statistics, summarization, minimum and maximum functions for this area and so on. These ideas were suggested by one of the interview participants although other participants agreed that keeping the features in eGrid as simple as possible and within the actual needs of the control center operators would probably encourage them to use eGrid without being confused by lots of unnecessary and complicated tools.

3.4.3.6 Textual Input Device

Annotations in the latest version of eGrid are freehand annotations, polygons, lines and points. These are important types of annotations but being able to add textual annotations to the map is also an essential feature. When asking the participants about the best textual input devices to be used for annotating maps, the responses were varied. The options were a regular keyboard, a virtual keyboard, and a stylus. One of the participants stated that “it will be intuitive to use through touch, stylus and keyboard. Better than multiple mice and screens and the paper copies of the maps with the annotations”. Some participants believed that having a pen or stylus is a better option in case of writing a few words on the map. A keyboard is important when trying to write a lot of text. A combination of a stylus and a keyboard seems like the best option for future enhancements of eGrid but a choice has to be made between a regular keyboard and a virtual keyboard. Being exposed to the iPad, a couple of participants expressed that they got used to the virtual keyboard concept and would be comfortable having it in an environment like eGrid. However, most of the other participants explained that having a regular keyboard is better than the virtual option because of the haptic feedback that allows them to write faster with fewer mistakes.

3.5 Summary

This chapter explained the first case study of this research in detail. It started by defining the goal of this case study that is to develop a multi-touch tabletop GIS environment for

the control center of a utility company using agile software engineering methods. Afterwards, an explanation of the techniques used to develop the prototype eGrid is presented and finally the design details are discussed along with reasons for certain design decisions. Overall, the agile techniques that were used to gather requirements and create eGrid were helpful in designing an interactive tabletop application driven by user feedback. The resulting environment was tailored to the needs of the potential users and the design aimed at solving their current data management problems, enabling them to use GIS for collaborative decision making and motivating them to embrace the new available technologies effectively for their day-to-day work. Requirements for a new GIS tabletop application for a different application domain were explored next. The similarities between the idea of the new application and the design of eGrid motivated our team to think about reusing pieces of the design of eGrid to help the development of this new application and potentially other applications as well. The following chapter describes the second case study of this research in which a library of reusable components is extracted from eGrid to benefit similar GIS tabletop applications.

Chapter Four: CASE STUDY 2 – GISforTT

This chapter describes the second case study of this research. The case study presents an agile approach for extracting a library of reusable components from eGrid to support the implementation of other multi-touch GIS tabletop applications. The idea of this study was formed upon receiving the requirements of a new GIS tabletop application for an oil and gas company, which motivated us to seize the opportunity and extract from eGrid the components that may be reused in the second application. The goal of this study is to test the feasibility of using the extraction approach in an agile environment to create a library of reusable components extracted from an existing application. The extraction approach is also described in a referenced peer reviewed paper. [Selim2011]

4.1 Background and Motivation

The lightweight practices of agile software engineering were adopted to help develop software applications more quickly and so that changes during the entire software development life cycle could be accommodated more easily. These practices are the reason why agile software engineering in general has become popular [Cao2004].

Software reuse on the other hand, can provide enormous savings and benefits in software development. Building applications on top of reusable components could potentially reduce the development time and future maintenance efforts of the new application.

[Moser2006] Agile software development methods, such as Extreme Programming (XP), are not focused on building software for future reuse. Being committed to developing

products quickly, time is not spent up-front on extensive requirement gathering research [Beck2000] making it possible for the current system components to be reused in similar systems, because there are no guarantees that similar systems will be needed in the future. These systems are developed simply to satisfy the current customer and what this customer really wants [Moser2006] without thinking much about the needs of future customers or how the systems can be reused.

In the process of designing and developing eGrid, the design was not done such that it could be reused in future applications. However, the concepts behind designing some of the components in eGrid were not specific to the domain of electricity companies and could be generalized and used in other GIS tabletop applications. When the idea of another actual GIS application appeared, the motivation of reusing these components emerged. The value of the effort spent in designing these components can be preserved by extracting them and collecting them in a library that can potentially support other GIS applications. Similar to the domain of electricity and utility companies, other domains benefit from various GIS applications and may also need environments like eGrid to facilitate collaborating around and interacting with GIS data on multi-touch digital devices. Examples of these domains include command and control, traffic control, and agricultural and geological domains. The suggested type of reuse is classified as opportunistic reuse, since resources were not spent during the development of eGrid on planning for reuse. As previously explained in the literature review chapter, research

suggests that using opportunistic reuse methods are suitable and more appropriate for agile software development environments, as opposed to planned reuse methods which include investing in various reuse activities up front [Ghanam2011].

The reuse extraction approach evaluated by this case study is a bottom-up approach that relies on refactoring and testing to extract the useful reusable components from an existing application. The approach is designed to be used on demand when a new application might benefit from components in an existing application. The extraction process is guided and supported by the usage examples found in the first application and the code around these components in the original application is changed as well to use the newly extracted and modified components. Usage examples are also extracted from the original application since they help identify the appropriate use of the extracted components [Washizaki2004].

The approach also relies on acceptance or functional tests which can help the refactoring process by planning the changes needed to be done and making sure that the resulting products match the requirements of the customer and behave as expected. This idea was inspired by research recommendations to use acceptance tests to model variability and features across similar software systems [Ghanam2012]. Acceptance tests are used to determine the components which can be reused in the new application and the extra features and changes that need to be done to these components. Acceptance tests are also

used as a safety net in order to make sure that refactoring will not change the behaviour of the original application and that the modified version will fulfill the user stories of the new application. The tests can grow as the library grows, capturing user requirements as they evolve, and they can be used to make sure the system meets the user requirements [Miller 2001]. The approach is incremental since one component is extracted at a time. It is also iterative such that the requirements from any other new application interested in using the library can actually modify the design of the library and add more options/variability points to the components extracted.

The goal of this case study is to find out whether the suggested extraction approach can be used in an agile software development process such as Extreme Programming (XP) and whether reuse in general can be achieved in XP by extracting reusable assets from existing applications.

4.2 Sky Hunter and Reuse Analysis

The second application is called Sky Hunter. It is an application for the oil and gas industry. Its main target is to visualize oil well sensor data that is captured and stored in a GIS server. The requirements of this second application motivated the creation of the library, since there are some features that are common between both eGrid and Sky Hunter and may actually be useful for other GIS tabletop applications. To start thinking about which features can be reused, three areas of knowledge need to be analyzed as

described by Mili et al [Mili1995]. The first area of analysis is knowledge about the source domain. In this case study, it is the requirements and design of eGrid, which was described in details in the previous chapter. The second area of analysis is knowledge about the target domain; in this case it is the requirements of Sky Hunter. And the third area of analysis is knowledge about how design elements from the source domain map to design elements in the target domain; in other words, the design elements from eGrid that can potentially satisfy some requirements of Sky Hunter and how they can achieve this goal. The target of the analysis was to answer the following questions:

- Which features/user scenarios fall in the intersection of the requirements of eGrid and the requirements of Sky Hunter?
- Which components of eGrid can be extracted into the library to benefit Sky Hunter and potentially other applications?
- How can the design of these components be flexible such that it accommodates different application needs?

Comparing the user stories of eGrid and Sky Hunter was useful for the analysis. In addition, a list of user stories for the library user, i.e. the software developer of GIS tabletop applications, was created based on this comparison. The library user story list was helpful in brainstorming the components to be extracted, the granularity of the interface functions exposed in the library and also the extraction approach itself. The

following list includes the components extracted, refactored for reuse and added to the library:

- Background map; where these options can be modified:
 - Enable or to disable gesture interactions on the map
 - Set the coordinates of the map
 - Set the default map service layers
- Corner Menus; with the added flexibility of changing the contents of the menus.
- Map Windows; where the library user can modify the gestures to create map windows and can add tools to the toolbar.
- Bookmarks, which represent the concept of marking some locations on the map and having more information associated with each location. The concept can be very useful in other application domains as well. An example of how this concept is applied in eGrid is the trouble report pins.
- Bookmark List; similar to the trouble report list window.
- Global Layer Configuration Manager

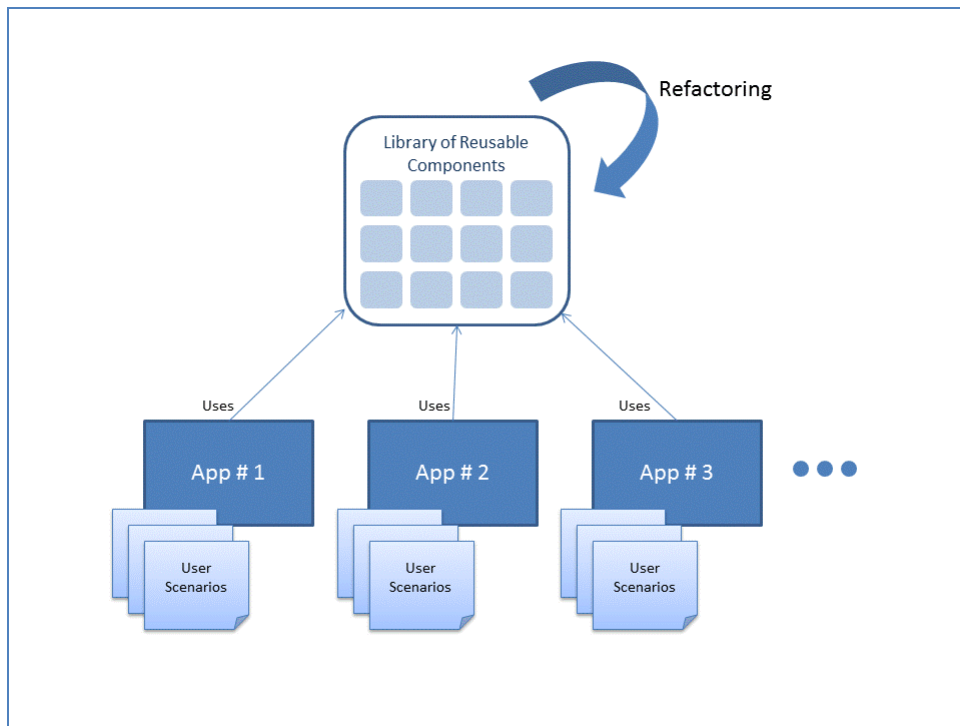


Figure 4-1: A diagram representing how multiple applications can participate in creating and enriching the library by iterative refactoring.

After the extraction is done, the library is created and eGrid is refactored such that it uses the library, other questions need to be answered:

- Which new features are needed in Sky Hunter that were not part of eGrid and, thus, are not supported by the extracted library?
- Which of these features are specific to the environment and domain of Sky Hunter and which of them can be useful for other GIS tabletop applications so they can be added to the library?

By analyzing these questions, an iterative pattern is revealed, where new applications built on top of the library can participate in enriching the library by adding more useful components or options to it, while these applications have still their own flavours. After a few iterations of building products on top of the library, users of the library should be able to select from various options and tailor the components to meet their specific needs.

4.3 Extraction Approach

Figure 4-2 is a flowchart of the extraction process. Based on the design on App#1 and the user stories of App#1 and App#2, components are extracted from App#1, refactored to make them reusable, and collected in a library to be use by both App#1 and App#2.

These components are extracted based on the user stories of both App#1 and App#2 and then they are used in both applications. The library can be modified and enriched by iterative refactoring. Using the library in other applications, such as App#3, can motivate the extraction and modification of more components from both App#1 and App#2.

To describe the approach in more detail, some notations and symbols are defined as follows:

- **S** is a system or an application.
- **AT** stands for a set of Acceptance Tests and **R(AT)** is the result of running this set of acceptance tests.
- **US** stands for User Stories.

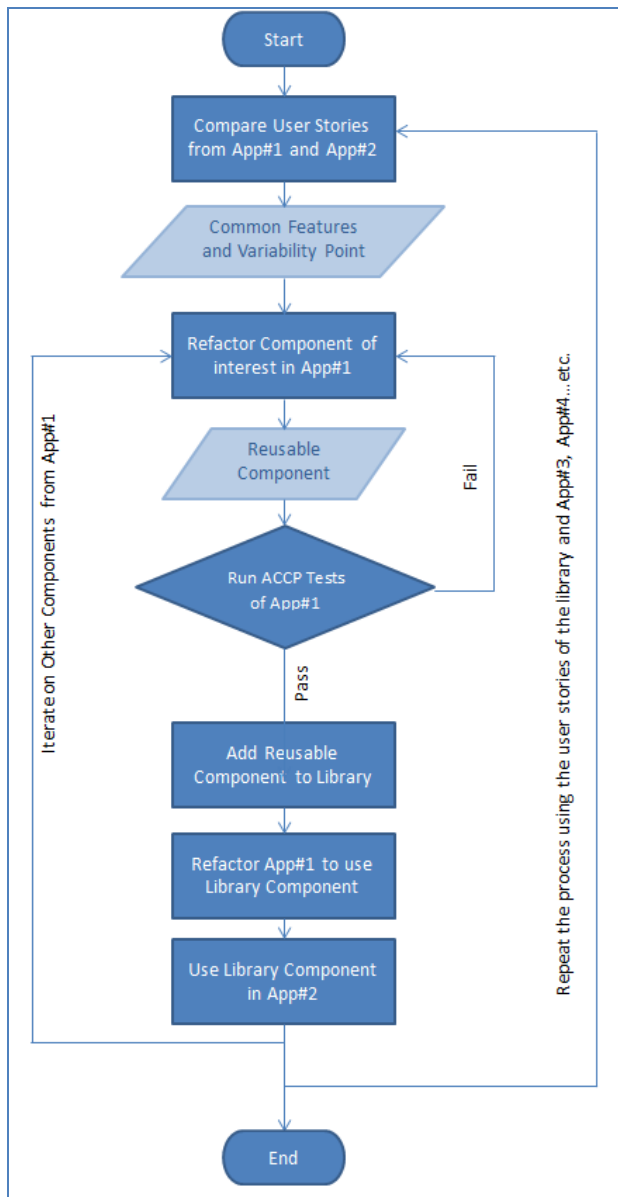


Figure 4-2: A flowchart summarizing the extraction of the library from App#1, App#2

...etc.

Say we have a system named S_1 that has already been developed to satisfy the requirements of a given customer. S_1 has been developed based on a set of user stories US_1 . These user stories are translated to a set of acceptance tests AT_1 . The extraction approach described here has the objective of maximizing reuse of the components (ex: user interface widgets) defined in S_1 to be utilized in a new application S_2 such that:

- a) All AT s are passing for the system S_1 . That is:

$$R(AT_i) = \text{Pass} \forall AT_i \in AT_1 \quad (\text{condition C1})$$

where R is the result of running an AT against the system

C1 should be maintained to be true throughout the whole process. This serves as a safety net for the refactoring tasks done on system S_1 .

- b) The new system S_2 is described using the set US_2 that translates to a set AT_2 .

The underlying functionality in S_2 is initially missing (i.e. classes and methods are empty, not yet implemented), therefore initially in most cases all acceptance tests for S_2 will fail (some of them may coincidentally pass):

$$R(AT_j) = \text{Fail} \exists AT_j \in AT_2$$

where R is the result of running an AT against the system..

- c) When a specific extracted component from S_1 is used in S_2 , there is a non-empty subset of AT_2 called AT_2^* which is equivalent to a non-empty subset of AT_1

called \mathbf{AT}_1 corresponding to this specific extracted component and relying on it to pass. In this case, \mathbf{AT}_2 will pass after using the extracted component in \mathbf{S}_2 , that is:

$$\mathbf{R}(\mathbf{AT}_j) = \text{Pass} \forall \mathbf{AT}_j \in \mathbf{AT}_2 \text{ given } \mathbf{AT}_2 \subseteq \mathbf{AT}_2 \text{ and } \mathbf{AT}_2 \neq \emptyset \quad (\text{condition C2})$$

where \mathbf{R} is the result of running an \mathbf{AT} against the system

C2 should be satisfied for each extracted component that is reused in \mathbf{S}_2 .

This is achieved by executing the following steps:

1. Write the \mathbf{AT} s for the new system \mathbf{S}_2 to verify the satisfaction of its user stories \mathbf{US}_2 .
2. Manually compare \mathbf{US}_1 with \mathbf{US}_2 to find potential reuse opportunities. Say $\mathbf{US}_a \in \mathbf{US}_1$ has been found to be similar to $\mathbf{US}_b \in \mathbf{US}_2$.
3. Compare \mathbf{AT}_a (which tests \mathbf{US}_a) with \mathbf{AT}_b (which tests \mathbf{US}_b). This comparison is useful in understanding:
 - a. Common features between \mathbf{AT}_a and \mathbf{AT}_b . The interface of the library does not need to provide options to configure these common features.
 - b. Variability points between \mathbf{AT}_a and \mathbf{AT}_b . The interface of the library in this case needs to provide some means of configuring the reused artefacts to satisfy these variability points/differences.
4. Refactor the artefacts in \mathbf{S}_1 (mainly the code and possibly also \mathbf{AT} s) that are relevant to the code unit of interest based on the outcome of step 3.

5. Separate the refactored artefact into the library. The objective of this step is to enable reuse of a single source code and avoid having redundant code in the two applications.
6. Refactor the code in S_1 such that it utilizes the new design of the code. This step has to be done observing C1, i.e. **ATs** of S_1 have to pass after refactoring.
7. Use the library component in the implementation of S_2 , observing C2.
8. Repeat the steps 3-7 for all component of S_1 that can be reused in S_2 .

These steps should be gone through for each increment of the extraction project, i.e. for each extracted component. Adding components to the library incrementally and iteratively is very important to:

- (1) Make sure that the extraction process does not have side-effects on the original application,
- (2) Provide a point after each increment to assess the approach itself and the partial result, and also
- (3) Fit this project into the iterations of an agile process such as XP.

After all of the components needed by S_2 have been extracted and added to the library, the whole process can be repeated again for any other new application S_3 , where the extracted components can be enhanced and modified based on the user stories US_3 of S_3 and more components can be extracted from S_1 and S_2 if they are useful to the new

application **S₃**. The library in this case can iteratively and continuously improve when it is used in more new systems, ex: **S₄, S₅, S₆...** etc.

4.4 Extraction Example from the Case Study

This section includes an example of applying the extraction approach on the Map Window interface component from eGrid and reusing it in Sky Hunter. Figure 4-3 is a diagram representing part of the design in eGrid before any extraction is done. All the classes represented in the diagram are in the user interface layer of eGrid. The implementation of the interface in eGrid was done in WPF 4.0 mostly in XAML files. Acceptance tests of eGrid were written using the language of unit tests. Since part of the implementation of eGrid was done using a markup language (i.e. XAML not C# code), it was difficult to implement the acceptance tests for user interface elements without using mock versions of these classes. Using the concept of mock objects for writing unit tests allows the tests to be written for all components even those written in markup languages. The mock objects are written such that domain code is replaced with dummy implementations that emulate real code. They simplify the test structure as described by Mackinnon et al. [Mackinnon2000] In eGrid, mock classes were created to support the creation of acceptance tests such that these classes have no XAML portions and they have mock implementations for the objects defined in the XAML files. The functionality is nearly perfectly preserved between the mock under test and the original user control. They are created only for the purpose of facilitating the creation of acceptance tests.

However, any updates to the original classes need to be reflected on the corresponding mock classes at all times.

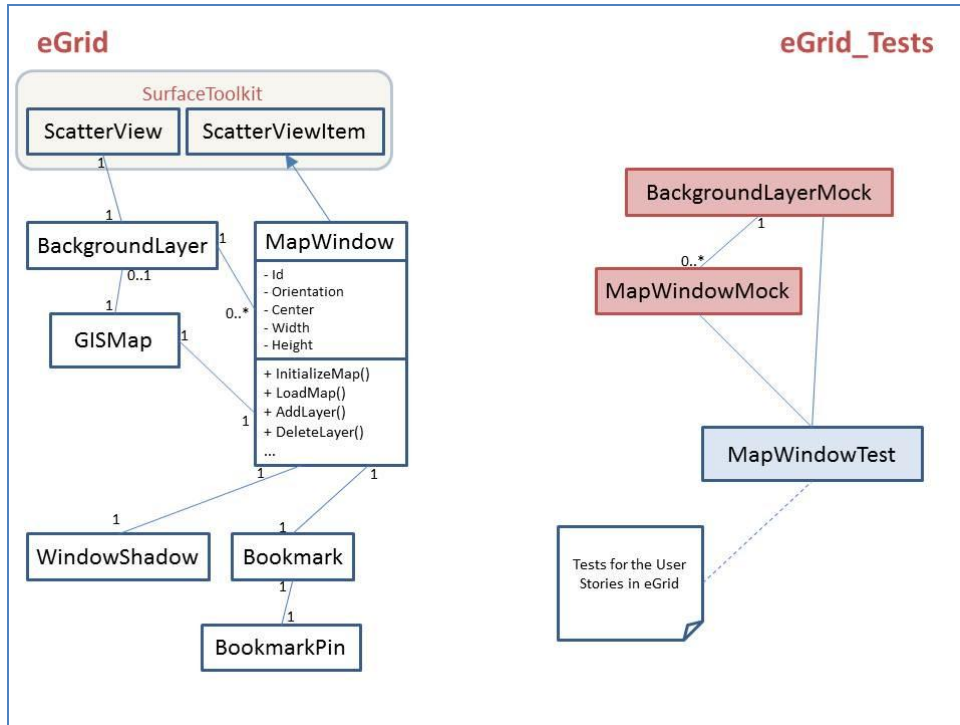


Figure 4-3: A diagram representing part of the design in eGrid before any extraction is done

The following steps are the same as the steps described in the extraction approach section and will be applied iteratively on each component to be extracted.

4.4.1 Writing Acceptance Tests for Sky Hunter

This step includes making sure that the acceptance tests of eGrid and Sky Hunter cover the scenarios related to the map window user interface component. From the user stories

of Sky Hunter, acceptance tests were created. Since there are no implemented classes for Sky Hunter, simple mock classes were created with no logic in them so that they can be attached to the tests. The created tests compile successfully but fail on run against the incomplete mock classes. **Figure 4-4** explains how the acceptance tests of eGrid and Sky Hunter interact with the class and mock classes in the implementation of eGrid.

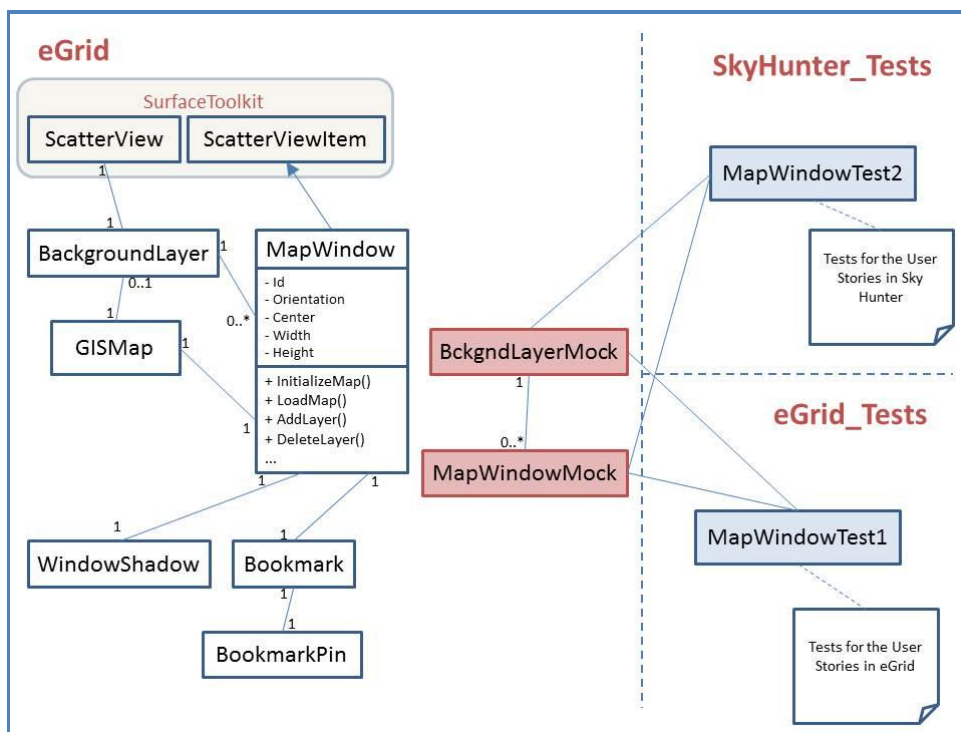


Figure 4-4: This diagram explains how the acceptance tests of eGrid and Sky Hunter interact with the class and mock classes in the implementation of eGrid.

4.4.2 Comparing the User Stories of eGrid and Sky Hunter

User stories explain actions the user would like to be able to do using the system. By comparing the user stories of eGrid and Sky Hunter concerning the map window component, reuse opportunities can be identified.

- In eGrid, the main user story that relates to the map window is: “As a user, I can open a map window using a lasso gesture. The map window is positioned on the screen such that the center of the created window is the same as the center of the lasso drawn and the map inside of the map window shows the area which was surrounded by the lasso”.
- In Sky Hunter, the user story related to the map window is: “As a user, I can create a map window using a single touch point. The map window is positioned on the screen such that its center point is underneath the touch point and the map displayed inside of the map window has the coordinates defined by the center point and a default zoom level”.

```

// eGrid original MapWindowTest
[TestMethod]
public void MapWindowTest()
{
    // Create the map window using a simulated lasso gesture
    Point lassoCenterPoint = new Point(120, 320); // x, y
    double lassoDiameter = 60;
    LassoGestureMock lassoMock = new LassoGestureMock(lassoCenterPoint, lassoDiameter);
    MapWindow mapW = new MapWindow(lassoMock);

    // Make sure the map window is created
    Assert.IsTrue(mapW != null);

    // Make sure the center position of the window relative to the screen is the same as
    // the center of the lasso gesture
    Assert.AreSame(mapW.center, lassoCenterPoint);

    // Calculate the map area (envelope) which should display inside of the map window
    // using a method inside of the LassoGestureMock
    Envelope lassoEnv = lassoMock.getLocationRelativeToBgMap();

    // Make sure the areal extent (coordinates) of the map inside the window show the same
    // area surrounded by the lasso gesture
    Assert.AreSame(mapW.getLocationEnvelope(), lassoEnv);
}

```

Figure 4-5: Example Acceptance test to capture the user story related to using the map window component in eGrid.

Comparing the two user stories reveals that the map window component needs to have different options for creation and accordingly different map contents once the windows are created. These user stories are documented in the related acceptance tests in eGrid and Sky Hunter as in the examples shown in Figure 4-5 and Figure 4-6.


```

// Sky Hunter MapWindowTest
[TestMethod]
public void MapWindowTest()
{
    // Create the map window using a single touch point
    Point singleTouchPoint = new Point(120, 320); // x, y
    MapWindow mapW = new MapWindow(singleTouchPoint);

    // Make sure the map window is created
    Assert.IsTrue(mapW != null);

    // Make sure the center position of the window relative to the screen is the
    // same as the touch point
    Assert.AreSame(mapW.center, singleTouchPoint);

    // Calculate the map area (envelope) which should display inside of the map window
    Envelope mapEnv = new Envelope(singleTouchPoint, MapWindow.getDefaultZoomLevel(),
        MapWindow.getDefaultMapWindowSize()); // touch point, zoom level, size

    // Make sure the areal extent (coordinates) of the map inside the window show the
    // calculated map area
    Assert.AreSame(mapW.getLocationEnvelope(), mapEnv);
}

```

Figure 4-6: Example Acceptance test to capture the user story related to using the map window component in Sky Hunter.

4.4.3 Analyzing the Acceptance Tests of eGrid and Sky Hunter

The next step is to determine the changes to be done to the map window component based on the acceptance tests of eGrid and Sky Hunter. The goal of comparing the acceptance tests is to understand the features which should be provided by the map window interface to serve both applications and to understand the options or variability points that are need to accommodate the differences between the scenarios in eGrid and in Sky Hunter. For example, the map window component needs to be refactored to allow for creating the window using the lasso gesture or creating it with a single touch point

plus performing the necessary position and map coordinate settings depending on the gesture used for creation.

4.4.4 Refactoring the Map Window Based on the Analysis

This step consists of refactoring the map window component that is necessary to support the user stories of both eGrid and Sky Hunter. For example, to handle two different methods for creating the map window, we defined multiple constructors in the refactored map window class and we implemented the logic which sets the position of the window relative to the screen and the coordinates of the inside map depending on the constructor used.

4.4.5 Moving the Map Window to the Library

In this step, the refactored map window classes were moved into the new library. Any application built on top of the library would use the map window component similar to using any third party component. The extraction of the component and the refactoring changes needed are not complete unless the acceptance tests in eGrid and Sky Hunter pass. Since the map window component now resides in the library, the corresponding acceptance tests that define how the map window should behave are moved to the library as well. The tests of both eGrid and Sky Hunter that are related to using this component were combined and slightly modified to run against the refactored map window. The projects of eGrid and Sky Hunter maintain the tests which define the scenarios in which

the MapWindow is used but the tests that define the behavior of the MapWindow itself are moved to the library project. The scenarios tested are semantically identical to the scenarios captured by the original acceptance tests from both applications but the interface of the map window component was modified to allow for the variability points and these changes are reflected on the use of the component inside of the acceptance tests.

4.4.6 Refactoring eGrid Classes that Interact with the Map Window

Since the constructors of the map window were changed to accommodate the extra new scenarios, changes will have to be done to all the code pieces in eGrid that interfaced with the map window component. For example, changing the constructors of the map window necessitates changes in the code parts of eGrid that used these constructors. Other changes also needed to be made in order to make sure that the map window classes absolutely encapsulate the behavior of the widget. At any point of time during the process described above, the acceptance tests of eGrid have to pass in order to make sure that the behavior has not been changed.

4.4.7 Using the Extracted Map Window in Sky Hunter

In this step, the extracted component will be used in the new application. The usage examples found in eGrid serve as a reference on how to use the component in Sky Hunter

as well. Figure 4-7 explains how the MapWindow class and its associated Mock class and test is moved into the library layer and used from there in both eGrid and Sky Hunter.

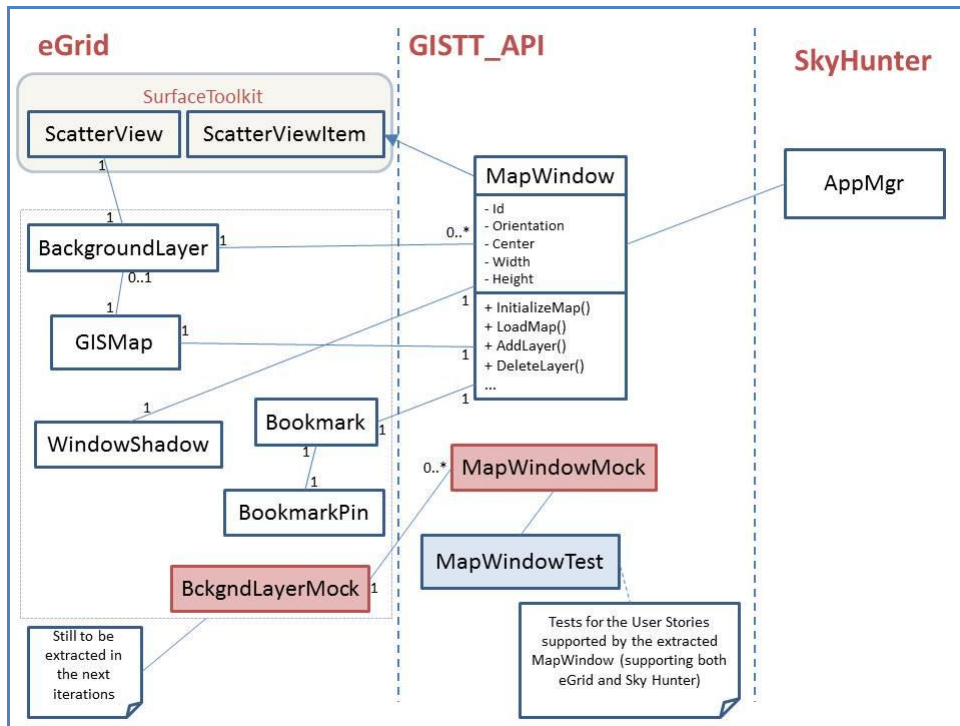


Figure 4-7: This diagram explains how the MapWindow class and its associated Mock class and test is moved into the library layer and used from there in both eGrid and Sky Hunter.

4.5 Description of the Resulting Library

The resulting library was named GISforTT that stands for GIS for Tabletop. It is available under the GNU public license on CodePlex [GISforTT_API2011]

The library supports the creation of WPF 4.0 applications to run on any multi-touch hardware with Windows 7 support and it is built on top of ESRI ArcGIS API for WPF

and Silverlight. The documentation of the library is also available online on the same codeplex site. The iterative nature of the extraction approach allows for more features and options to be added to the library as more applications are built on top of it. It currently provides the following features:

- A background map where the location, orientation and touch manipulation can be changed through the interface.
- Support for multiple pre-defined map layers as well as the ability to add any published map layers through their URLs and also online annotation/graphic layers.
- Marking bookmark locations on the background map.
- Corner menus that give access to the Bookmarks list, a map layer configuration widget and exiting the application. The corner menus can be added to all four corners of the screen, to two opposite corners or not added at all, in which case the developer will need to define different access methods.
- Support for creating and adding custom map Windows.

4.6 Enhancing the Library by Gathering User Feedback

This section describes a study performed on the extracted library. The purpose of the study is to gather user feedback related to the usefulness of the library and subsequently the usefulness of the extraction approach used and also gather initial usability feedback to be able to iteratively enhance the design of the library. This is not a formal usability

study; the goal is not to create statistics representing how useful and usable the library is but rather an approach for preliminary assessment and iterative refinement of the design of the library. The following sections describe the design of this simple study and how the library is enhanced accordingly.

4.6.1 Study Design

4.6.1.1 Participants and Recruitment

The study required a number of participants who are software developers with some experience in developing applications in Microsoft .NET platform using C# and WPF. This requirement is due to the fact that participants were required to build an application on top of the GISforTT library. A recruitment notice was sent to the Agile Software Engineering group of the University of Calgary. The participants who completed the study were five male graduate computer science students and they possessed an average of 3 years of experience in developing applications using C# and WPF. The study was done in the agile software engineering lab in the University of Calgary and each participant took an average of two hours to finish the study.

4.6.1.2 Study Tasks

After each participant signed the consent form, a basic explanation of the research goals and the study process was presented to them. The participant introduction and study explanation can be found in the Appendix. The results of the questionnaires were

analyzed and any critical changes to the code and documentation of the library were done before the next participant started their study. As explained before, the main goal of the study was to get user feedback for the purpose of enhancing the library. If any critical problems were discovered from user feedback, they were fixed or enhanced before the next study participant begins. Enhancing the library based on the feedback received iteratively was crucial in order to get more detailed and comprehensive feedback from the following participants. However, the changes which were introduced to the library between the different participants were only documentation changes or critical bug fixes. The design of the library itself was not changed at all, so this study still provides insight on the usefulness of the library extraction process.

The study participants were asked to use the library to create simple GIS applications on multi-touch surfaces. They were instructed to download the library code from the codeplex website link. This website also has an overview of the features supported by the library and the documentation. The library can be used to develop applications for Windows 7 and .NET framework 4.0. Therefore, the participants were required to use this type of platform for implementation. They were asked to perform a couple of exercises building on top of the library or they could choose their own application specifications. Next, they were asked to fill out a short questionnaire describing their experience throughout the study.

4.6.1.2.1 Exercise 1

The first exercise was to build upon the library to create an application for the government to browse their published demographic layers. The target platform for the application is any tabletop hardware with Windows 7 support. The participants could find some optional map layers on the public ArcGIS servers [ArcGIS_Services2011]. The initial coordinates of the background map were provided. The participants could choose either to allow the background map to be manipulated using finger touches in which case they have to implement a gesture to open individual map windows on top of the background map, or they could keep the default setting which disables the background map from being manipulated and supports the lasso gesture to open individual map windows. Participants could define some bookmark locations and add corner menus allowing the users to exit the application or open the layers configuration manager. They could add a function to allow the user to change the focus of the background map to another location and implement another custom function which opens a map window representing a specific location defined by longitude and latitude.

4.6.1.2.2 Exercise 2

The second exercise was a free exercise where the participants could use the library to develop their own custom multi-touch tabletop applications for any GIS domain. They were asked to come up with their own set of requirements as realistically as possible and

try to accomplish them using the support of the library and their own development capabilities.

4.6.1.3 Questionnaire Content

The questionnaire started by explaining the study and how the results of the questionnaire would be used. The questions prompted the participants to provide the following information:

1. The background experience of the participant regarding .NET framework, WPF 4.0, ESRI ArcGIS API and GIS systems in general.
2. Participants rating for the following statements based on their study experience, i.e. using the library to develop a GIS tabletop application. These statements referred to whether the library is useful to the GIS domain and its potential in supporting multi-touch applications. The rating was a choice between Strongly Disagree, Disagree, Neutral, Agree, Strongly Agree in response to the following statements:
 - a. I found the library to be useful in supporting multi-touch GIS applications.
 - b. I think the library can be used in a lot of GIS domains.
 - c. I thought the library was easy to use in the application.
 - d. The features of the library were relevant to the needs of my application.
 - e. Using the library was not frustrating.
 - f. I think the library has some future value for the industry.

3. Participants rating for the following statements based on their study experience, i.e. using the library to develop a GIS tabletop application. This group of statements referred to the quality of the documentation and code readability. The rating was a choice between Strongly Disagree, Disagree, Neutral, Agree,

Strongly Agree:

- a. The online documentation was useful in getting started with the library.
- b. The online documentation explained clearly how to use the library.
- c. The code of the library is readable.
- d. I could find what I was looking for in the library source code.

4. Participants rating for the following statements based on their study experience, i.e. using the library to develop a GIS tabletop application. This group of statements referred to the actual design of the library and its components. The rating was a choice between Strongly Disagree, Disagree, Neutral, Agree,

Strongly Agree:

- a. The design of the library is easy to understand and follow.
- b. It was easy to customize the user interface elements provided by the library.
- c. The UI elements of the library did what I expected them to do.
- d. The user interface elements provided by the library are flexible and generic.
- e. I think it will be easy to build on top of the library.

5. The topics that are missing from the documentation of the library.
6. Ideas of new customizations that can be added to the prototype of the library.
7. Architectural or design changes that must be done to the library to make it more useful.

4.6.2 Responses of Participants and Enhancements

Regarding the experience of the participants with the technologies of .NET, WPF and GIS, participants had an average of 3 years of experience in .NET and WPF but some participants had absolutely no experience with GIS and ESRI products.

The following list includes the rating statements included in the questionnaire and the responses given by the participants ordered by time on this scale: (Strongly Disagree = 1, Disagree = 2, Neutral = 3, Agree = 4, Strongly Agree = 5). Note that because of the design of the study, the overall experience of the participants could have improved throughout the study because of the changes done to the documentation of the library or the critical bugs resolved. Therefore, an average rating would not be accurate in this case. Instead, the results were stated below sorted by time in ascending order to reflect the gradual improvement, if any.

1. I found the library to be useful in supporting multi-touch GIS applications: 5-5-1-5-5
2. I think the library can be used in a lot of GIS domains: 4-4-3-4-4
3. I thought the library was easy to use in the application: 3-4-1-5-4

4. The features of the library were relevant to the needs of my application: 4-4-2-3-4
5. Using the library was not frustrating: 2-4-1-4-4
6. I think the library has some future value for the industry: 4-5-4-4-5
7. The online documentation was useful in getting started with the library: 1-5-4-5-5
8. The online documentation explained clearly how to use the library: 4-4-3-4-4
9. The code of the library is readable: 4-4-4-4-5
10. I could find what I was looking for in the library source code: 4-3-4-4-4
11. The design of the library is easy to understand and follow: 4-4-4-4-4
12. It was easy to customize the user interface elements provided by the library: 2-4-2-3-4
13. The UI elements of the library did what I expected them to do: 4-4-2-4-4
14. The user interface elements provided by the library are flexible and generic: 3-4-3-4-3
15. I think it will be easy to build on top of the library: 4-4-2-4-4

When asked about the topics that are missing from the documentation of the library, the participants provided a number of different ideas that were very helpful in improving the experience of the following participants. For example, one participant complained that the documentation was not clear on the steps needed to download the latest version of the library API to get started. This participant explained that getting started with the library was the hardest part. However, after being able to setup the environment and build the

solution everything was easy, explaining “I would have liked to see an architecture diagram for the software components for the API. The text description was helpful for the design, but not for how the main components fit together.” Upon receiving the comments from this participant, detailed steps were added to the online documentation of the library to explain how to download the API and setup the development environment such that the following participants would not encounter the same problem.

Another participant asked for a brief introduction on the terminology being used such as map window, types of layers... etc. He also asked for more details on expected method arguments (e.g. are points relative to the screen dimensions or the map dimensions).

Other participants suggested adding step-by-step tutorials to build a simple application on top of the library, which was added to the documentation. Another request was to make sure that all library methods have proper C# documentation such that hover text and auto complete show the description of the parameters needed to make a method call or instantiate an object. This participant explained “As this is my go-to for finding out how to interact with objects, I had to rely on the online documentation, which is slower.”

When asked about ideas of new customizations that can be added to the prototype of the library, the participants gave a number of different suggestions. Some of these suggestions are:

- Giving the user the ability to change the orientation of corner menu buttons and add custom icons to these buttons
- Providing the user with more control on what gestures provide which behaviour.
- Providing the user with the ability to do fine grain customizations to the buttons of the corner menus.

Suggestions like these reflect the fact that more applications need to be built on top of the library in order to add more flexibility to the components extracted. Since these changes include changing the actual design of the library, they were not performed during the study. Instead, they were noted down as ideas for library enhancements and can be thought of in the future.

When asked about architectural or design changes that must be done to the library to make it more useful, some participants could not provide an answer given the limited time they spent during the study. Some other participants described some defects that they encountered while using the library in their applications. For example, one participant explained: “I encountered a number of bugs on the Evoluce One digital tabletop I was trying to use for system development and testing. Some were due to external issues like the map service I selected, some were due to issues with the table itself (never being able to get TouchToolkit to recognize the lasso gesture), and some were due to the application itself (not displaying annotations correctly)”. The main complaint of this participant was not the usability or the documentation of the library but

the issues found during the exercise and attempting to resolve these problems before the following participant was important such that similar technical issues are not faced again. Other participants described their positive experience during the study. One participant stated “I found the API interesting and would like to use in future for my application”. Another participant explained that “the documentation for various features, such as Map Layers, Corner Menus was excellent. I was able to implement each of them with little difficulty and without those 6 or 7 links on the website, I would have been totally lost.”

4.6.3 Limitations

Overall the feedback received from study participants was satisfactory and their comments were extremely helpful in identifying usability issues and important future improvements to the library. However, this study has a number of limitations. First of all, the number of participants is very limited. To be able to receive more valuable feedback and factor out the effects of personal experiences from participant impressions, more participants need to be recruited to perform the study. Since each participant had to have software programming experience in C# and .NET environments and since they had to have access to a Windows 7 development environment and possibly a Windows 7 enabled multi-touch device, gathering an adequate number of participants was a challenge. The second limitation of this study is the necessary critical changes that needed to be done between participants. Even though the changes were done only to the documentation of the library or to fix critical bugs, the results could not be summarized in

appropriate statistics since the environment changed between the iterations. This study evaluated the outcome of the extraction process; i.e. the library, as opposed to evaluating the extraction process itself. However, the study was a first step and further research efforts are needed to assess the extraction process and identify its strengths and weaknesses.

4.7 Summary

In this chapter, a case study is presented in which an extraction approach was used to create a library of reusable components based on the design of eGrid. The goal was to be able to use this library to support the design and implementation of similar GIS multi-touch tabletop applications, thus preserving the effort spent in designing eGrid and gaining the benefits of reuse while creating new applications. The proposed approach was used in an agile software development environment, specifically an Extreme Programming environment. It is an iterative and incremental approach where one component is extracted at a time when it is needed. It is also designed for opportunistic reuse. The approach supports reuse reactively which minimizes investment upfront, and guarantees that component extraction happens only when there is a real need for a component's functionality by other applications. The opposite would be a proactive treatment where possibilities of reuse have to be examined and handled in the beginning facing the risk of losing this investment if the components were not actually used in other applications. Furthermore, this approach makes use of the usage examples and the

acceptance tests, which are existing assets in many agile development processes, to inform the decisions made about exposing and hiding certain features in the API.

A challenge exists, however, when the reusable components that are to be extracted are mostly user interface components. This is due to the fact that testing user interface components is often tricky, especially if they are designed using scripting or mark-up languages. In particular, creating acceptance tests for user interface related stories is far from trivial [Finsterwalder2001]. Usually, in user-interface intensive applications, tests do not cover the entire portion of the user interface layer. This makes refactoring code in this layer risky. A partial solution has been used in this case study where mock classes have been created for user interface classes that have scripting parts. The purpose of having these mock classes is to help create acceptance tests for the user interface classes. This served as a solution for creating acceptance tests for user interface classes, however, these mock classes have to be updated whenever the logic in the original classes change which is extra effort and overhead. Although we emphasized the use of acceptance tests as a safety net throughout the whole process, the proposed approach can leverage any type of tests (e.g. unit tests) to do the same job. The most critical aspect is that these tests need to be automated so that the abstraction process could be done with minimal risk. The process may be repeated by developing other applications based on the new library and enriching the library by adding more components using the same approach. The benefit of repeating the process in the future is to further assess the extraction approach

as well as enhance the library by generalizing it according to the options needed by the new applications.

To gather user feedback on the extracted library and to further enhance the library, a preliminary study was conducted in which participants used the library to build simple GIS tabletop applications and reported their feedback through an online questionnaire. The questionnaire gathered user feedback regarding how effective, valuable and useful the library is for developing GIS tabletop systems and indirectly assess how successful the approach is in designing the library.

Chapter Five: Conclusion

5.1 Summary of Results

This research explores the requirements and design decisions needed to create an interactive multi-touch tabletop environment for geocollaboration in the control center of utility companies and possibly other GIS domains. It also explores agile software development approaches that can be used for designing and developing similar interactive systems or reusing them across different domains.

From the first case study described in this research, agile software engineering approaches were found to be useful in designing an interactive collaborative GIS application to be used by a local utility company. The agile practices described in this case study were used to gather the requirements for this new system and evaluate those requirements. These approaches were also useful in engaging users and collecting feedback regarding better design ideas to make the system more useful and potentially ease its adoption. From the second case study, an opportunistic reuse approach was proposed, in line with agile software engineering, to generalize the requirements of the first system and extract a library of reusable components that can be used to create new interactive GIS tabletop applications.

5.2 Thesis Contribution

The contribution of this research is threefold: First, through the first case study, a design is proposed for eGrid; an advanced prototype for supporting the control center of the utility company. It's a multi-touch tabletop environment for interacting with GIS maps. Second, through the second case study, a library of reusable components was extracted from eGrid to support developing interactive GIS tabletop applications. Last but not least, the research contributed the agile approaches and practises which were used in the first case study to design the environment of eGrid and in the second case study to reactively extract the reusable components from eGrid.

The main conclusion of this research can be summarized as follows: Agile software engineering approaches can be used to gather the requirements and design multi-touch tabletop applications such as applications for interacting with GIS data and these practises can be used to iteratively engage users in providing feedback and driving the enhancements of the application. Agile software engineering approaches can also be used to support opportunistic reuse by extracting reusable components from existing applications to benefit new applications through iterative refactoring.

5.3 Limitations and Future Work

The research described in this thesis has a number of limitations. First of all, in order to better assess the usefulness and suitability of the agile practices used in the first case

study, the scope needs to be broadened and these practises need to be employed in other projects in addition to eGrid, which was described in this research. In addition, to be able to properly assess the extraction approach described in the second case study, the extracted library needs to be used to create multiple different GIS tabletop applications and consequently be able to assess the usefulness of the library and the iterative extraction approach. Formal usability studies with an appropriate number of participants can also be helpful in assessing the quality of the design of eGrid and the usefulness of the extracted library. Overall, the practices and approaches promoted by this research and used in the two case studies described are very promising but need to be further evaluated in the future.

References

[Agile2001] Manifesto for Agile Software Development, <http://www.agilemanifesto.org>

(last accessed December 2010)

[Agrawal2004] Agrawal, P., Rauschert, I., Inochanon, K., Bolelli, L., Fuhrmann, S., Brewer, I., Cai, G., MacEachren, A. and Sharma, R. (2004) Multimodal Interface Platform for Geographical Information Systems (GeoMIP) in Crisis Management. In International Conference on Multimodal Interfaces. State College, PA: ACM

[Apple2012] Apple iPhone: www.apple.com/iphone (last accessed January 2012).

[Arciniegas2009] Arciniegas, G. and Janssen, R. (2009) Using a touch table to support participatory land use planning. In Anderssen, R.S., R.D. Braddock and L.T.H. Newham (eds) 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, pages 2377–2383, 2009.

[ArcGIS_API2011] ArcGIS API for WPF and Silverlight can be found at, <http://resources.esri.com/arcgisserver/apis/silverlight/> (Last Accessed November 2011)

[ArcGIS_Services2011] Published map services from ArcGIS website can be found at, <http://services.arcgisonline.com/ArcGIS/rest/services> (Last Accessed November 2011)

[ArcGIS2011] ArcGIS system description can be found at,

www.esri.com/software/arcgis/index.html (Last Accessed November 2011)

[Beck2000] Beck, K. (2000) Extreme Programming Explained: Embrace Change.

Addison-Wesley, Reading 2000.

[Benko2003] Benko, H., Ishak, E. and Feiner, S. (2003) Collaborative Visualization of an

Archaeological Excavation. In: NSF Lake Tahoe Workshop on Collaborative

Virtual Reality and Visualization (CVRV 2003), October 26-28.

[Benko2004] Benko, H., Ishak, E. W. and Feiner, S. (2004) Collaborative Mixed Reality

Visualization of an Archaeological Excavation. In Proceedings of the 3rd

IEEE/ACM international Symposium on Mixed and Augmented Reality

(November 02 - 05, 2004). Symposium on Mixed and Augmented Reality. IEEE

Computer Society, Washington, DC, 132-140.

[Bieman1995] Bieman, J. and Kang, B. (1995) Cohesion and reuse in an object oriented

system. SIGSOFT Software Engineering Notes, 20,259-262, August, 1995.

[Bier1993] Bier, E. et al. (1993) "Toolglass and Magic Lenses: The See-Through

Interface," SIGGRAPH 1993.

- [Bjørneseth2008] Bjørneseth F., Dunlop M. and Strand, J. (2008) Dynamic positioning systems: usability and interaction styles. In Proceedings of the 5th Nordic Conference on Human-Computer interaction: Building Bridges (Lund, Sweden, October 20 - 22, 2008). NordiCHI '08, ol. 358. ACM, New York, NY, 43-52.
- [Brewer2000] Brewer, I., MacEachren, A.M., Abdo, H., Gundrum, J. and Otto, G. (2000) Collaborative Geographic Visualization: Enabling shared understanding of environmental processes. In IEEE Information Visualization Symposium, Salt Lake City, Utah, 2000.
- [Brown2008] Brown, J., Lindgaard, G. and Biddle, R. (2008) Stories, Sketches, and Lists: Developers and Interaction Designers Interacting Through Artefacts. In Proceedings of the Agile Conference 2008, Toronto, Canada, pp. 39–50.
- [Burd1996] Burd, E., Munro, M. and Wezeman, C. (1996) Extracting reusable modules from legacy code: considering the issues of module granularity. In Proceedings of the Third Working Conference on Reverse Engineering, pp. 189–196.
- [Buxton2010] Buxton, B. (2010) A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future. An invited paper presented in Society for Information Display (SID) Symposium Digest of Technical Papers, May 2010, Volume 41(1), Session 31, 444-448.

[Buxton2012] History of multi-touch surfaces and interaction:

www.billbuxton.com/multitouchOverview.html (Last accessed January 2012)

[Byte_Tags2011] Information on Byte Tags used with Microsoft Surface can be found at:

[http://msdn.microsoft.com/en-us/library/ee804885\(v=surface.10\).aspx](http://msdn.microsoft.com/en-us/library/ee804885(v=surface.10).aspx) (Last accessed November 2011)

[Caballero2002] Caballero, R. and Demurjian, S.A. (2002) Towards the Formalization of a Reusability Framework for Refactoring. In Proceedings of the 7th International Conference Software Reuse: Methods, Techniques, and Tools, ICSR-7, Austin, TX, USA.

[Cao2004] Cao, L., Mohan, K., Xu, P. and Ramesh, B. (2004) How Extreme Does Extreme Programming Have to Be? Adapting XP Practices to Large-Scale Projects. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004) - Track 3, p. 30083c.

[Clements2001] Clements, P. and Northrop, L. (2001) Software Product Lines: Practice and Patterns. Addison-Wesley, US.

[Coltekin2003] Coltekin, A. (2003) Virtual Reality as an Interface to GIS: Focus on WWW. In 21st International Cartographic Conference (ICC), Durban, South Africa.

- [Daiber2009] Daiber, F., Schöning, J. and Krüger, A. (2009) Whole Body Interaction with Geospatial Data. Book Chapter in: Smart Graphics, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009.
- [DeAntonellis1994] De Antonellis, V., Castano, S. and Vandoni, L. (1994) Building reusable components through project evolution analysis. Information Systems 19(3), 259–274.
- [Epps2006] Epps, J., Lichman, S. and Wu, M. (2006) A study of hand shape use in tabletop gesture interaction. In CHI '06 Extended Abstracts on Human Factors in Computing Systems (Montréal, Québec, Canada, April 22 - 27, 2006). CHI '06. ACM, New York, NY, 748-753.
- [Esenther2002] Esenther, A., Forlines, C., Ryall, K., Shipman, S. (2002) DiamondTouch SDK: Support for Multi- User Multi-Touch Applications, ACM Conference on Computer Supported Cooperative Work (CSCW), November 2002.
- [ESRI] Information about ESRI Products is available at: <http://www.esri.com/> (last accessed June 2011).
- [Evoluce] Evoluce Table Technology available at: <http://www.evoluce.com/en/> (last accessed November 2010).

- [Finsterwalder2001] Finsterwalder, M. (2001) Automating acceptance tests for GUI applications in an extreme programming environment. In: Proceedings of the 2nd International Conference on eXtreme Programming and Flexible Processes in Software Engineering, Villasimius, Sardinia, Italy, pp. 114–117. Addison-Wesley, Reading, 2001.
- [Forlines2005] Forlines, C. and Shen, C. (2005) DTLens: multi-user tabletop spatial data exploration. In Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology (Seattle, WA, USA, October 23 - 26, 2005). UIST '05. ACM, New York, NY, 119-122.
- [Fowler1999] Fowler, M. (1999) Refactoring: Improving the Design of Existing Code. Addison-Wesley, Longman Publishing Co., Inc., Boston, MA, USA.
- [Fowler2000] Fowler, M. (2000) Refactoring Improving the Design of Existing Code. Addison-Wesley, 2000.
- [Frakes1994] Frakes, W. and Pole, T. (1994) An empirical study of representation methods for reusable software components, IEEE Transactions on Software Engineering, 20(8), 617-630, 1994.
- [Furuichi2005] Furuichi, M., Mihori, Y., Muraoaka, F., Estenther, A., Ryall, K. (2005) DTmap demo: Interactive tabletop maps for ubiquitous computing. In UbiComp 2005.

- [Geller2006] Geller, T. (2006) Interactive Tabletop Exhibits in Museums and Galleries. IEEE Comput. Graph. Appl. 26, 5, 6-11.
- [Ghanam2009] Ghanam, Y., and Maurer, F. (2009) Extreme Product Line Engineering: Managing Variability & Traceability via Executable Specifications, Proc. of the international conference on Agile methods in software development, Agile 2009, Chicago.
- [Ghanam2010] Ghanam, Y., Andreychuk, D. and Maurer, F. (2010) Reactive variability management in agile software development. In: Agile '10: Proceedings of International Conference on Agile Methods in Software Development, IEEE Computer Society, 2010.
- [Ghanam2011] Ghanam, Y. (2011) An Agile Framework for Variability Management in Software Product Line Engineering. PhD. Thesis, University of Calgary, Department of Computer Science, August, 2011.
- [GISforTT_API2011] Codeplex site for the API GISforTT:
<http://gisfortabletopsapi.codeplex.com/> (Last accessed February 2012)
- [GoogleEarth2011] Google Earth can be found at:
<http://www.google.com/earth/index.html> (Last accessed November 2011)

- [Han2005] Han, J. (2005) Low-cost Multi-touch Sensing through Frustrated Total Internal Reflection. In Proc. of UIST '05, ACM Press, (2005), 115–118.
- [Hancock2009] Hancock, M., Hilliges, O., Collins, C., Baur, D., and Carpendale, S. (2009) Exploring tangible and direct touch interfaces for manipulating 2D and 3D information on a digital table. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces Banff, Alberta, Canada, November 23 -25, 2009). ITS '09. ACM, New York, NY, 77-84.
- [Henninger1997] Henninger, S. (1997) An evolutionary approach to constructing effective software reuse repositories. *ACM Transaction on Software Engineering*, 6(2), 111-140, 1997.
- [Hevner2004] Hevner, A., March, S., Park, J., and Ram, S. (2004) Design Science in Information Systems Research, *MIS Quarterly* (28:1) 2004, pp. 75-105.
- [Hesselmann2009] Hesselmann, T., Flöring, S. and Schmitt, M. (2009) Stacked Half-Pie menus: navigating nested menus on interactive tabletops. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Banff, Alberta, Canada, November 23 - 25, 2009). ITS '09. ACM, New York, NY, 173-180.

- [Hosseini-Khayat2010] Hosseini-Khayat, A. (2010) Distributed Wizard of Oz Usability Testing for Agile Teams, M.Sc. Thesis, University of Calgary, Department of Computer Science, January (2010).
- [Hummel2007] Hummel, O. and Atkinson, C. (2007) Supporting Agile Reuse through Extreme Harvesting. In Proc. Int'l Conf. Agile Processes in Software Eng. and Extreme Programming, Springer, 2007, pp. 28–37.
- [Jacobson1997] Jacobson, I., Griss, M., and Johnson, P. (1997) Software Reuse: Architecture, Process and Organization for Business success, Addison-Wesley.
- [Jones1984] Jones, T. C. (1984) Reusability in Programming: A survey of the state of the art. In IEEE Trans. Software Engineering, vol. 10, no. 5, pp. 488-494, Sept. 1984.
- [Khandkar2010] Khandkar, S.H. and Maurer, F. (2010) A Domain Specific Language to Define Gestures for Multi-Touch Applications. The 10th SPLASH Workshop on Domain-Specific Modeling, October 17-18, (2010), Reno/Tahoe, Nevada, USA.
- [Khandkar2010b] Khandkar, S. H. (2010) A Domain-Specific Language for Multi-Touch Gestures, M.Sc. Thesis, University of Calgary, Department of Computer Science, December (2010).
- [Koskinen2008] Koskinen, H., Laarni, J., and Honkamaa, P. (2008) Hands-on the process control: users preferences and associations on hand movements. In CHI '08

Extended Abstracts on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 3063-3068.

[Krahnstoever2002] Krahnstoever, N., Kettebekov, S., Yeasin, M., and Sharma, R. (2002) A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays, Proc. of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI 2002), Pittsburgh, USA, 2002.

[Krueger2006] Krueger, C. (2006) New methods in software product line practice. Commun. ACM 49(12), 37–40.

[Lanubile] Lanubile, F. and Visaggio, G. (1997) Extracting reusable functions by flow graph-based program slicing, IEEE Transactions on Software Engineering, vol. 23, no. 4, 1997, pp. 246-259.

[Lemos2009] Lemos, O., Bajracharya, SK, Ossher, J., Masiero, PC, and Lopes, C. (2009) Applying test- Driven code search to the reuse of auxiliary functionality. In 24th Annual ACM Symposium on Applied Computing (SAC 2009), 2009.

[MacEachren2003] MacEachren A.M., Brewer I., Cai G. and Chen J. (2003) Visually-Enabled Geocollaboration to Support Data Exploration and Decision-Making International Cartographic Conference, Durban, South Africa, pp. 394-400.

- [Mackinnon2000] Mackinnon, T., Freeman, S., Craig, P. (2000) Endo-Testing: Unit Testing with Mock Objects, eXtreme Programming and Flexible Processes in Software Engineering - XP2000, Cagliari, Sardinia, Italy, June 21-23, 2000.
- [Marakas1998] Marakas, G. and Elam, J. (1998) Semantic Structuring in Analyst Acquisition and Representation of Facts in Requirements Analysis, *Information Systems Research* (9:1), March 1998, pp. 37-63.
- [McCarey2005] McCarey, F., Ó Cinnéide, M. and Kushmerick, N. (2005) An Eclipse Plugin to Support Agile Reuse. In: *Proc. of the 6th Int. Conf. on Extreme Progr. and Agile Processes*, Sheffield, 2005.
- [Mommel2008] Mommel, T. and Reiterer, H. (2008) Model-Based and Prototyping-Driven User Interface Specification to Support Collaboration and Creativity. *Journal of Universal Computer Science* 14(19), 2008.
- [Mens2004] Mens, T. and Tourwe, T. (2004) A survey of software refactoring. *IEEE Transactions on Software Engineering* 30(2), 126–139, 2004.
- [Micire2009] Micire, M., Desai, M., Courtemanche, A., Tsui, K. and Yanco, H. (2009) Analysis of natural gestures for controlling robot teams on multitouch tabletop surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, New York, NY, USA, 41-48.

[Microstation] Microstation software products are available at:

<http://www.bentley.com/en-US/Products/microstation+product+line/> (Last accessed November 2011)

[Mili1995] Mili, H., Mili, F., and Mili, A. (1995) Reusing software: issues and research directions. *IEEE Transactions on Software Engineering*, 21(6), 1995, pp. 528–562.

[Miller2001] Miller, R. and Collins, C. (2001) Acceptance Testing. *Proc. XPUniverse 2001*, July, 2001.

[Mohagheghi2004] Mohagheghi, P. (2004) *The Impact of Software Reuse and Incremental Development on the Quality of Large Systems*, Doctoral Thesis, Department of Computer and Information Science, Norwegian University of Science and Technology, 2004.

[Moser2006] Moser, R., Sillitti, A., Abrahamsson, P., and Succi, G. (2006) Does refactoring improve reusability? In *Proc. 9th International Conference on Software Reuse (ICSR'06)*, 2006, pp. 287- 297.

[Nasehi2010] Nasehi, S.M. and Maurer, F. (2010) Unit Tests as API Usage Examples, *Proc. of 26th IEEE International Conference on Software Maintenance (ICSM 2010)*, Timișoara, Romania, 2010.

- [Neves1997] Neves, N., Silva, J., Goncalves, P., Muchaxo, J., Silva, J. M., and Camara, A. (1997) Cognitive spaces and metaphors: A solution for interacting with spatial data. *Computers and Geosciences*, vol. 23, pp. 483-488, 1997.
- [Nóbrega2008] Nóbrega, R., Sabino, A., Rodrigues, A., and Correia, N. (2008) Flood Emergency Interaction and Visualization System. In *Proceedings of the 10th international Conference on Visual information Systems: Web-Based Visual information Search and Management*, Salerno, Italy, September 11 - 12, 2008.
- [Ning1993] Ning, J.Q., Engberts, A. and Kozaczynski, W. (1993) Recovering reusable components from legacy systems by program segmentation. In *Proceedings of Working Conference on Reverse Engineering*, pp. 64–72, 1993.
- [Pohl 2005] Pohl, K., Böckle, G. and Linden, F. (2005) *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer, Germany, 2005.
- [Prieto-Diaz1996] Prieto-Díaz, R. (1996) Reuse as a New Paradigm for Software Development. In: *Proceedings of the International Workshop on Systematic Reuse*, London (1996).
- [Rauschert2002a] Rauschert, P., Agrawal, S., Fuhrmann, I., Brewer, H., Wang, R., Sharma, G., and MacEachren, A. (2002) Designing a human-centered, multimodal GIS interface to support emergency management. Presented at ACM

GIS'02, 10th ACM Symposium on Advances in Geographic Information Systems, Washington, DC, USA, 2002.

[Rauschert2002b] Rauschert, I., Sharma, R., Fuhrmann, S., Brewer, I. and MacEachren, A. (2002) Approaching a New Multimodal GIS-Interface. Proceeding of the 2nd International Conference on GIS (GIScience), CO, USA (2002)

[Refactoring2011] <http://martinfowler.com/refactoring/> (Last accessed November 2011)

[SaturateApp] Saturate is available at: <http://www.saturateapp.com/> (Last accessed July 2011).

[Schöning2008] Schöning, J., Hecht, B., Raubal, M., Krüger, A., Marsh, M., and Rohs, M. (2008) Improving interaction with virtual globes through spatial thinking: helping users ask "why?". In Proceedings of the 13th international Conference on intelligent User interfaces (Gran Canaria, Spain, January 13 - 16, 2008). IUI '08. ACM, New York, NY, 129-138.

[Schöning2008b] Schöning, J. and Krüger, A. (2008) Multi-Modal Navigation through Spatial Information. In Proc. Int. Conf. on GIScience (2008).

[Schöning2008c] Schöning, J., Rohs, M. and Krüger, A. (2008) Spatial Authentication on Large Interactive Multi-Touch Surfaces. In: Proc. TABLETOP 2008. IEEE

- [Schöning2009] Schöning, J., Daiber, F., Rohs, M. and Krüger, A. (2009) Using hands and feet to navigate and manipulate spatial data. In: CHI 2009: CHI 2009 extended abstracts on Human factors in computing systems. ACM, New York (2009)
- [Selim2010] Selim, E. and Maurer, F. (2010) eGrid: Supporting the Control Room Operation of a Utility Company with Multi-Touch Tables. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Saarbrücken, Germany, November 7-10, 2010). ITS'10, ACM, New York, NY.
- [Selim2011] Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and eXtreme Programming (XP 2011), Madrid, Spain, 2011.
- [Shen2006] Shen, C., Ryall, K., Forlines, C., Esenther, A., Vernier, F.D., Everitt, K., Wu, M., Wigdor, D., Morris, M.R., Hancock, M., and Tse, E. (2006) Interfaces and Interactions for Direct-Touch Horizontal Surfaces. IEEE Computer Graphics and Applications, vol. 26, no. 5, pp. 36-46, (2006).
- [Slyngstad2008] Slyngstad, O., Li, J. , Conradi, R., Rønneberg, H., Landre, E. and Wesenberg, .H (2008) The Impact of Test Driven Development on the Evolution of a Reusable Framework of Components - An Industrial Case Study, Proceedings

of the 2008 The Third International Conference on Software Engineering Advances, p.214-223, October 26-31, 2008.

[Spindler2009] Spindler, M., Stellmach, S., and Dachsel, R. (2009) PaperLens: advanced magic lens interaction above the tabletop. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Banff, Alberta, Canada, November 23 - 25, 2009). ITS '09. ACM, New York, NY, 69-76.

[Sugumaran2008] Sugumaran, V., Tanniru, M. and Storey, V. (2008) A knowledge-based framework for extracting components in agile systems development. Inf. Technol. and Management 9(1), 37–53, 2008.

[Surface2012] Information about the Microsoft Surface table can be found at:

<http://www.microsoft.com/surface/en/us/default.aspx> (Last Accessed February 2012)

[SurfaceToolkit] Microsoft® Surface® Toolkit for Windows Touch Beta:

<http://www.microsoft.com/download/en/details.aspx?id=13787> (Last Accessed June 2011).

[Tuddenham2009] Tuddenham, P., Davies, I., and Robinson, P. (2009) WebSurface: an interface for co-located collaborative information gathering. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Banff,

Alberta, Canada, November 23 - 25, 009). ITS '09. ACM, New York, NY, 181-188.

[EuropeanCommission1998] European Commission (UNIGIS). (1998) Guidelines for Best Practice in User Interface for GIS. ESPRIT/ESSI project no. 21580.

[Voida2009] Voida, S., Tobiasz, M., Stromer, J., Isenberg, P., and Carpendale, S. (2009) Getting practical with interactive tabletop displays: designing for dense data, "fat fingers," diverse interactions, and face-to-face collaboration. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Banff, Alberta, Canada, November 23 - 25, 2009). ITS '09. ACM, New York, NY, 109-116.

[Wartik1992] Wartik, S., Prieto-Diaz, P. (1992) Criteria for comparing reuse-oriented domain analysis approaches. *International Journal of Software Engineering and Knowledge Engineering* 2(3), 403–431, 1992.

[Washizaki2003] Washizaki, H. and Fukazawa, Y. (2003) Automated extract component refactoring. In: Marchesi, M., Succi, G. (eds.) *XP 2003. LNCS*, vol. 2675, pp. 328–330. Springer, Heidelberg, 2003.

[Washizaki2004a] Washizaki, H. and Fukazawa, Y. (2004) A technique for automatic component extraction from object-oriented programs by refactoring. In: *Science*

of Computer Programming. *New Software Composition Concepts*, vol. 56(1-2), pp. 99–116 (April 2005) ISSN 0167-6423, doi:10.1016/j.scico.2004.11.007

[Washizaki2004b] Washizaki, H., Fukazawa, Y. (2004) A technique for automatic component extraction from object-oriented programs by refactoring. *Sci. Comput. Program.* 56(1-2): 99-116. 2005.

[Wilson2008] Wilson, A., Izadi, S., Hilliges, O., Garcia-Mendoza, A. and Kirk, D. (2008) Bringing physics to the surface. In *Proc. of UIST 2008*, ACM Press, (2008), 67-76.

[Wobbrock2009] Wobbrock, J., Morris, M., and Wilson, A. (2009) User-defined gestures for surface computing. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009)*. CHI '09. ACM, New York, NY, 083-1092.

[WPF2011] Information on Windows Presentation Foundation can be found at:

<http://windowsclient.net/wpf/> (Last accessed November 2011)

[Zhao2000] Zhao, J. (2000) A slicing-based approach to extracting reusable software architectures. In *Proceedings of the 4th European Conference on Software Maintenance and Reengineering*, pages 215–223, 2000.

APPENDIX A: Ethics Board Approval



MEMO

CONJOINT FACULTIES RESEARCH ETHICS BOARD
 c/o Research Services
 Main Floor, Energy Resources Research Building
 3512 - 33 Street N.W., Calgary, Alberta T2L 1Y7
 Telephone: (403) 220-3782
 Fax: (403) 289-0693
 Email: rburrows@ucalgary.ca
 Thursday, August 12, 2010

To: Elaf S. Selim
 Computer Science

From: Dr. Kathleen Oberle, Chair
 Conjoint Faculties Research Ethics Board (CFREB)

Re: Certification of Institutional Ethics Review: eGrid: A Multi-User, Digital Table-Top Environment for Managing Control Room GIS Maps

The above named research protocol has been granted ethical approval by the Conjoint Faculties Research Ethics Board for the University of Calgary.

Enclosed are the original, and one copy, of a signed **Certification of Institutional Ethics Review**. Please make note of the conditions stated on the Certification. A copy has been sent to your supervisor as well as to the Chair of your Department/Faculty Research Ethics Committee. In the event the research is funded, you should notify the sponsor of the research and provide them with a copy for their records. The Conjoint Faculties Research Ethics Board will retain a copy of the clearance on your file.

Please note, an annual/progress/final report must be filed with the CFREB twelve months from the date on your ethics clearance. A form for this purpose has been created, and may be found on the "Ethics" website, <http://www.ucalgary.ca/research/cfreb>

In closing let me take this opportunity to wish you the best of luck in your research endeavor.

Sincerely,

Russell Burrows
 For:
 Kathleen Oberle, PhD
 Chair, Conjoint Faculties Research Ethics Board

Enclosures(2)
 cc: Chair, Department/Faculty Research Ethics Committee
 Supervisor: Frank Maurer



CERTIFICATION OF INSTITUTIONAL ETHICS REVIEW


This is to certify that the Conjoint Faculties Research Ethics Board at the University of Calgary has examined the following research proposal and found the proposed research involving human subjects to be in accordance with University of Calgary Guidelines and the Tri-Council Policy Statement on *"Ethical Conduct in Research Using Human Subjects"*. This form and accompanying letter constitute the Certification of Institutional Ethics Review.

File no: **6570**
 Applicant(s): **Elaf S. Selim**
 Department: **Computer Science**
 Project Title: **eGrid: A Multi-User, Digital Table-Top Environment for Managing Control Room GIS Maps**
 Sponsor (if applicable):

Restrictions:

This Certification is subject to the following conditions:

1. Approval is granted only for the project and purposes described in the application.
2. Any modifications to the authorized protocol must be submitted to the Chair, Conjoint Faculties Research Ethics Board for approval.
3. A progress report must be submitted 12 months from the date of this Certification, and should provide the expected completion date for the project.
4. Written notification must be sent to the Board when the project is complete or terminated.


Kathleen Oberle, PhD
Chair
Conjoint Faculties Research Ethics Board

AUG 12 2010
Date:

Distribution: (1) Applicant, (2) Supervisor (if applicable), (3) Chair, Department/Faculty Research Ethics Committee, (4) Sponsor, (5) Conjoint Faculties Research Ethics Board (6) Research Services.



UNIVERSITY OF
CALGARY

MEMO

Conjoint Faculties Research Ethics Board (CFREB)
Research Services Office
Main Floor, Energy Resources Research Building
Research Park
 Telephone: (403) 220-3782 or (403) 210-9863
 Fax: (403) 289-0693
 Email: csjahrau@ucalgary.ca or rburrows@ucalgary.ca

To: Elaf S. Selim
 Department of Computer Science
 Faculty of Science

Date: June 20, 2011

From: Dr. Kathleen Oberle, Chair
 Conjoint Faculties Research Ethics Board

Re: Approval of Modification for: "eGrid: A Multi-User, Digital Table-Top Environment for Managing Control Room GIS Maps"
 Original Approval Date: August 12th 2010
 File No: 6570

The Certificate of Institutional Ethics Review issued on August 12th 2010 continues in force and extends to the modifications as set out in your email/memo dated June 16th 2011. Your request to revise the study consent and recruitment materials to each more accurately reflect an additional research focus on the testing of the usability of the application programming interface (API), rather than solely on the testing of the usability of eGrid itself, is approved as described.

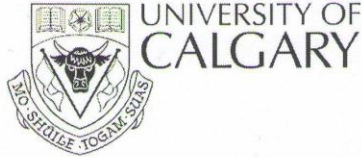
You should attach a copy of the documentation you provided in order to request the modification, together with a copy of this memorandum, to the original Certification in your files.

Sincerely,

Kathleen Oberle, PhD
 Chair, Conjoint Faculties Research Ethics Board

Cc: Dr. Frank Maurer (Supervisor)

APPENDIX B: Co-Authors Authorization



Co-Author Permission

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

April 20th 2012

I, **Frank Maurer**, grant Elaf Selim full permission to use the content of the following co-authored publications in her MSc dissertation:

- Selim, E. and Maurer, F. (2010) eGrid: Supporting the Control Room Operation of a Utility Company with Multi-Touch Tables. In Proceedings of the ACM international Conference on interactive Tabletops and Surfaces (Saarbrucken, Germany, November 7-10, 2010). ITS'10, ACM, New York, NY.
- Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and eXtreme Programming (XP 2011), Madrid, Spain, 2011.

A handwritten signature in black ink, consisting of a stylized 'F' followed by a long horizontal stroke.

Frank Maurer



Co-Author Permission

University of Calgary
2500 University Drive NW
Calgary, Alberta
T2N 1N4

April 20th 2012

I, **Yaser Ghanam**, grant Elaf Selim full permission to use the content of the following co-authored publications in her MSc dissertation:

- Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and eXtreme Programming (XP 2011), Madrid, Spain, 2011.

Yaser Ghanam

A handwritten signature in black ink, appearing to read "Y. Ghanam", written over a horizontal line.



Co-Author

Permission

University of Calgary

2500 University Drive NW

Calgary, Alberta

T2N 1N4

April 20th 2012

I, **Chris Burns**, grant Elaf Selim full permission to use the content of the following co-authored publications in her MSc dissertation:

- Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and eXtreme Programming (XP 2011), Madrid, Spain, 2011.

A handwritten signature in blue ink, appearing to read "Chris Burns".

Chris Burns



Permission

Co-Author

University of Calgary

2500 University Drive NW

Calgary, Alberta

T2N 1N4

April 20th 2012

I, **Teddy Seyed**, grant Elaf Selim full permission to use the content of the following co-authored publications in her MSc dissertation:

- Selim, E., Ghanam, Y., Burns, C., Seyed, T. and Maurer, F. (2011) A Test-Driven Approach for Extracting Libraries of Reusable Components from Existing Applications. Proc. of 12th International Conference on Agile Processes and eXtreme Programming (XP 2011), Madrid, Spain, 2011.

Teddy Seyed

A handwritten signature in blue ink that reads "Teddy Seyed". The signature is written in a cursive, flowing style.

APPENDIX C: Case Study 1 Prototype Demos Timeline

Nov 4, 2009	Initial requirement gathering meeting with electricity company Business Analysts and IT
Nov 18, 2009	Brainstorming and User interface Sketches
Dec 02, 2009	Initial mock demo on Microsoft Surface
Jan 22, 2010	Demo to industrial partners
Jan 27, 2010	Demo to TR Labs
Feb 19, 2010	Creating first demo video of eGrid and posting it on youtube
Mar 22, 2010	SurfNet media launch event, demos, discussions and feedback
Apr 15, 2010	Demos for electricity company business analysts and team leaders
May 09, 2010	eGrid prototype version 1.0 running on Surface table
May 25, 2010	CAMUG presentation and demo to industrial users
May 26, 2010	Demos and discussions with electricity company technical architect and ESRI analyst, feedback about new features
Jun 02, 2010	Working on prototype version 1.1 using Gesture Toolkit to support gesture design in eGrid on Windows 7 platforms
Jul 06, 2010	Demo to TR Labs and DexPro and discussions about requirements
Jul 08, 2010	Recruitment notice for eGrid interviews
Aug 11, 2010	TR Labs demos and presentation
Sep 06, 2010	Demo of eGrid to industrial users from other companies
Sep 30, 2010	Demo of eGrid to Sky Hunter company
Oct 10, 2010	Creating second video of eGrid in HD and posting on youtube
Oct 15, 2010	Demo to electricity company users
Oct 27, 2010	Starting interviews with control center operators and GIS analysts
Nov 01, 2010	Demo to electricity company and ESRI users and more feedback
May 18, 2011	Creating third video for eGrid prototype 1.2 for ESRI user conference demos
May 25, 2011	Presenting eGrid in ESRI user conference in Calgary

APPENDIX D: Library Evaluation Study

Recruitment Notice:

Elaf Sameh Selim – June 2011 – [Recruitment Notice](#)

GIS for Tabletop: An API for creating Multi-touch, Digital Table-Top GIS Map Applications

The following information is in regards to a study being conducted at the University of Calgary, Department of Computer Science, as a part of MSc Research. The study will be conducted by Elaf Selim, a Masters student at the Agile Software Engineering Lab. For further details and inquiries please see the contact information at the end of this email.

The general purpose of the research is to study the usability and software design guidelines for creating gesture-based, multi-touch applications that utilize digital tables for accessing and manipulating GIS maps in collaborative team environments. Such applications are useful in a lot of domains including Utility Companies, Command and Control centers, Transportation ... etc. We are developing an API to facilitate the creation of multi-touch digital tabletop GIS applications, where the users will be utilizing multi-touch digital table hardware for viewing and annotating sets of maps and different users will have concurrent access to the table. To assess the usefulness of this new API, a usability study will be performed and the feedback of the participants will be captured through short interviews.

You are invited to participate in the usability study. We are looking for participants who have experience with .NET framework and WPF 4.0 and would welcome trying to use the prototype version of the API to develop simple multi-touch tabletop GIS applications. After experimenting with the API to create a sample application for 2-3 hours, participants will be asked to provide their feedback by filling in a survey. We will not collect any identifying information or personal data about them.

If you would like to participate in this study please email Elaf Selim at esselim@ucalgary.ca you're your preferred timings. If you have any further questions or want clarification regarding this research and/or your participation, please contact: Elaf Selim, Department of Computer Science, University of Calgary, esselim@ucalgary.ca.

Participant Guide:

GIS for Tabletop API Usability Study – Participant Guide

The general purpose of the research is to study the usability and software design guidelines for creating gesture-based, multi-touch applications that utilize digital tables for accessing and manipulating GIS maps in collaborative team environments. Such applications are useful in a lot of domains including Utility Companies, Command and Control centers, Transportation ... etc. We are developing an API to facilitate the creation of multi-touch digital tabletop GIS applications, where the users will be utilizing multi-touch digital table hardware for viewing and annotating sets of maps and different users will have concurrent access to the table. To assess the usefulness of this new API, a usability study will be performed and the feedback of the participants will be captured through short interviews.

In this study, you will have the opportunity to use the GIS API for tabletops to create simple GIS applications on multi-touch tables. You can download the API from the codeplex link below, which also has an overview of the features supported by the API and the documentation. The API runs on Windows 7, .NET framework 4.0. Try to do the two exercises described below building on top of the API or choose your own application specifications. The next step is to fill out a short questionnaire about your experience using the API.

API Codeplex Site:

<http://gisfortabletopsapi.codeplex.com/>

Exercise 1:

Build upon the GIS tabletop API to create an application for the government to browse their published demographic layers. The target platform for the application is any tabletop hardware with windows 7 support. You can find some optional map layers here:

<http://services.arcgisonline.com/ArcGIS/rest/services>

The initial location of the background map is North America (60.196156,-40.517578) and (9.535749,-154.248047). You can choose either to allow the background map to be manipulated using finger touches in which case you have to implement a gesture to open individual map frames on top of the background map, OR you can keep the default setting which disables the background map from being manipulated and supports the lasso gesture to open individual map windows.

Define some bookmark locations and add corner menus allowing the users to exit the application or open the layers configuration manager. Add a function to allow the user to change the focus of the background map to another location. Implement another custom function which opens a map frame representing a specific location defined by longitude and latitude.


Exercise 2:

Use the API to develop your own custom multi-touch tabletop application for any GIS domain. Design your own set of realistic requirements and try to accomplish them using the support of the API and your own development capabilities.

Post-study Questionnaire:

<http://www.surveymonkey.com/s/TVJ9NWD>

Questionnaire:

GIS for Tabletops API Usability Study Feedback		Exit this survey
1. Informed Consent		
Title of the Survey		
		
<p>Welcome to the "Title".</p> <p>Before you consent to participate and proceed to the survey, it is important that you understand your rights as a participant.</p> <p>If you want more details about something mentioned here, or information not included here, you should feel free to ask by contacting the investigator esselim@ucalgary.ca. Please take the time to read this page carefully before you proceed.</p> <p>The University of Calgary Conjoint Faculties Research Ethics Board has approved this research study.</p> <p>*Purpose of the Study</p> <p>The purpose of the research is to study the usability and software design guidelines for gesture-based, multi-touch applications that utilize digital tables for accessing and manipulating GIS maps data in collaborative team environments. The goal of the case study is to develop a table-based prototype – eGrid – for supporting the operation of the utility company control centre. eGrid, along with a GIS server, can be used in the control rooms to replace the tedious and asynchronous handling of large paper maps. The users of the environment will be utilizing digital table hardware and our eGrid prototype for viewing and annotating sets of maps and different users will have concurrent access to the system. From eGrid, an API is extracted to help in creating similar multi-touch GIS applications.</p> <p>When a basic prototype for the API is ready, a usability study will be performed to assess the usefulness of this API in supporting similar applications. The feedback of the participants in usability studies will be captured through this post study survey. The results of the usability study will be used to assess the development process of eGrid and the extracted API.</p> <p>*What Will I Be Asked To Do?</p> <p>We ask that you complete the questionnaire by following the link below.</p> <p>The questionnaire consists of a few questions and should take approximately 15 minutes to complete.</p> <p>Participation is voluntary and completely anonymous. You may withdraw from the project at any time by not completing the questionnaire, but the investigators will keep the data collected to that point.</p> <p>You may be requested to participate in a short 15 minute interview with the researcher after you complete the questionnaire if your answers need more explanation.</p> <p>*What Type of Personal Information Will Be Collected?</p> <p>No personal identifying information will be collected in this study, and all participants shall remain anonymous.</p>		

***Are there Risks or Benefits if I Participate?**
 There are no risks with participating in this project.
 There is no remuneration or compensation for participation in this project.

***What Happens to the Information I Provide?**
 Only the investigators will have access to the answers you provide. However, they will publish the results of their analysis of your answers in aggregate form, in academic journals and conferences.

The data from all questionnaires will be kept by the investigators indefinitely, stored on a password-protected computer system.

***To Participate**
 By following the link below you indicate that you
 1. understand to your satisfaction the information provided to you on this page about your participation in this research project, and
 2. agree to participate as a research subject.

In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from this research project at any time. You should feel free to ask for clarification or new information throughout your participation.

***Questions/Concerns**
 If you have any further questions or want clarification regarding this research and/or your participation, please contact:

Ms. Elaf Selim, Department of Computer Science, esselim@ucalgary.ca
 And Dr. Frank Maurer, Department of Computer Science, fmaurer@ucalgary.ca

If you have any concerns about the way you've been treated as a participant, please contact

Bonni

Powered by **SurveyMonkey**
 Create your own [free online survey](#) now!

2. Survey Questions

***1. Please describe your background experience regarding the following: .NET framework, WPF 4.0, ESRI ArcGIS API, GIS systems in general...etc**

***2. Based on your experience of using the API to develop a GIS tabletop application, how do you rate the following statements:**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I found the API to be useful in supporting multi-touch GIS applications.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think the API can be used in a lot of GIS domains.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I thought the API was easy to use in the application.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The features of the API were relevant to the needs of my application.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Using the API was not frustrating.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think the API has some future value for the industry.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

***3. Based on your experience of using the API to develop a GIS tabletop application, how do you rate the following statements:**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The online documentation was useful in getting started with the API.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The online documentation explained clearly how to use the API.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The code of the API is readable.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I could find what I was looking for in the API source code.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

***4. Based on your experience of using the API to develop a GIS tabletop application, how do you rate the following statements:**

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The design of the API is easy to understand and follow.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It was easy to customize the user interface elements provided by the API.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The UI elements of the API did what I expected them to do.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The user interface elements provided by the API are flexible and generic.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think it will be easy to build on top of the API.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. In your opinion, what are the topics which are missing from the documentation of the API?

6. The API is in prototype version, so what other customizations do you think the API needs to provide?

7. Do you think there are architectural or design changes that must be done to the API to make it more useful?

8. Further Comments:

Powered by **SurveyMonkey**
Create your own [free online survey](#) now!