A TOOL FOR MAPPING IMAGES TO GEOGRAPHIC AREAS

by

Omar Valenzuela Agosto

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
OCTOBER 2008

Approved by:

________________________________ __________________
Néstor J. Rodríguez, PhD Date
President, Graduate Committee

________________________________ __________________
José A. Borges, PhD Date
Member, Graduate Committee

________________________________ __________________
Manuel Rodríguez, PhD Date
Member, Graduate Committee

________________________________ __________________
Ismael Pagán-Trinidad, MSCE Date
Representative of Graduate Studies

________________________________ __________________
Isidoro Couvertier, PhD Date
Chairperson of the Department

ABSTRACT

A TOOL FOR MAPPING IMAGES TO GEOGRAPHIC AREAS

By

Omar Valenzuela Agosto

October 2008

Chair: Néstor J. Rodríguez
Department: Electrical and Computer Engineering

This work presents a web-based application with the capability of creating and editing georeferenced boundary maps (polygons) of geographical areas using georeferenced images. The application is platform-independent. It can run on any computer with a compatible browser and an Internet connection. The application provides the capability of locally storing boundary maps attaching spatial information from the images that facilitates image searching. The boundaries can be imported from other sources and they can also be exported to make them available for other GIS applications. A usability study conducted with the application demonstrated that it is easy to learn and to use. The participants of the study were able to complete 96% of the task with just a fifteen minutes tutorial. I addition they found all the interaction actions tested to be easy to use.

RESUMEN

UNA HERRAMIENTA PARA LA DEMARCACIÓN DE IMÁGENES A ÁREAS GEOGRÁFICAS

Por

Omar Valenzuela Agosto

Octubre 2008

Consejero: Néstor J. Rodríguez
Departamento: Ingeniería Eléctrica y Computadoras

Este trabajo presenta una aplicación basada en el Web con la capacidad de crear y editar mapas de demarcaciones (polígonos) georeferenciados de áreas geográficas utilizando imágenes georeferenciadas. La aplicación es independiente de plataformas. Esta puede correr en cualquier computadora con un navegador compatible y una conexión a Internet. La aplicación provee la capacidad de guardar mapas de demarcaciones localmente, atando información de entorno de las imágenes para facilitar la búsqueda de las mismas. Las demarcaciones se pueden importar de otras fuentes y se pueden exportar para uso en otras aplicaciones de GIS. Se condujo un estudio de usabilidad con la aplicación que demostró que ésta es fácil de aprender y de utilizar. Los participantes del estudio pudieron realizar el 96% de las tareas con sólo un entrenamiento de quince minutos. Ellos encontraron además, que todas las acciones de interacción que se probaron fueron fáciles de realizar.

To God and my family.

ACKNOWLEDGEMENTS

During the progress of my graduate studies at the University of Puerto Rico at Mayagüez Campus, several people and institutions collaborated directly and indirectly with my research. Without their support, it would be impossible for me to finish my work. That is why I wish to dedicate this section to recognize their support.

First, I would like to thank God for giving me the opportunity to continue studying and finish my graduate studies and for all the great experiences that came with it. Also, I would like to thank my parents and family for their support, help, and encouragement to finish.

Thanks to my friends at the University of Puerto Rico at Mayagüez and the ones that are outside of the university, that gave me help and words of wisdom through the graduate studies.

This research and thesis were done with the help of the several people that without their counseling, knowledge, and direction this research would have never been finished. Dr. Néstor J. Rodríguez, my advisor, for his advice and for always being available when I needed help with my research. Dr. José A. Borges for being part of my graduate committee and for his advice in this research. Dr. Manuel Rodríguez for being part of my graduate committee and Ismael Pagán-Trinidad for being the representative of graduate studies. Prof. Linda L.

Vélez for taking the time to help me with this research and Neysa López for her support. Prof. Domingo Rodríguez for his financial support and for letting me become part of the WALSAIP project.

Student Research under WALSAIP Project NSF-CISE-CNS Grant No. 0424546 provided funding and resources for the development of this research.

TABLE OF CONTENTS

TABLE LIST

Tables **Page**

FIGURE LIST

Tables **Page**

Chapter 1

INTRODUCTION

1.1 Motivation

Over the years, there have been numerous methods to organize geospatial data. Before 1962, the tasks of finding the necessary information was accomplished by organizing hundreds or even thousands papers maps in a specified order and browsing manually over them. The overall performance of the computers at the time did not make the Geographical Information Systems (GIS) tasks viable with a software application because of their lack of storage space, memory, and processing power. The first commercial vendors of GIS software emerged in 1980. These were Environmental Systems Research Institute (ESRI) and Computer Aided Resource Information System (CARIS). Faster computers with more capacity were being developed that promised to make tasks of searching and organizing information easier and faster. Nevertheless, even if all that computer power, there still was a lack of bandwidth for sharing large amounts of data contained on satellite and aerial images, with other researchers and/or colleges around the world.

Most of the limitations of the computers systems have been overcome in recent years providing developers more powerful tools to develop more interactive GIS applications. This new era of information is making Internet Protocol (IP) transfer speeds fast enough for the implementation of new GIS applications that use network topologies to carry great amounts of information. Today we have the capacity of having hardware storage systems, small in size, but with vast amounts of storage and at the same time being able to communicate that data across any type of network for world access to the data.

Our proposed tool was motivated by the need of researchers of the WALSAIP (Wide Area Large Scale Automated Information Processing) project at the University of Puerto Rico in Mayagüez to have a flexible image mapping tool useful for GIS applications. The tool had to be capable of being customized for future developments. Our tool also needed to be accessible via the World Wide Web (WWW) in order to make the application easy to use at different sites without the need to install any type of software in different types of computers and operating systems.

Our tool is able to open a georeferenced image in a couple of seconds and display the tools necessary to create a georeferenced polygon on top of the image. The user marks the boundary area with a couple of clicks building a polygon shape surrounding it. Once the polygon and/or boundary is complete, the user will be able to fill out the spatial data (e.g., country, city) and store it inside the local database. This boundary will be accessible by other WALSAIP applications for searching different types of geographical content referenced to multiple map images. The user will have the option of exporting a boundary in order to perform additional tasks with other applications that supports KML files, an open standard by the Open Geospatial Consortium (OGC) [Google08].

Older tools did not have the flexibility of performing these tasks through the Internet with just a couple clicks of the mouse. People needed to find a paper map in order to search for an image area. Using older tools, researchers needed to be physically on the lab in order to find, for example, a river or a lake. Geographical Information Systems (GIS) introduce methods and environments to visualize, manipulate, and analyze geospatial data. [Sayar05] A web-based GIS application can reach many more users at low cost by using the Internet. [Athan07] Now with the dynamic interaction of Web 2.0 applications this tool has the feel of a standard off the web application with flexibility of letting the user work at any place where there is Internet access and share the data with other colleges using KML files.

1.2 Objectives

The goal of this project to create a web-based GIS tool capable of creating boundaries of geographical areas with vector-based tools (polygons), store that information, and make it available for later use. One of the requirements of the tool is that it runs on the web without the need of any API controlled by a third party. The tool should also manage high-resolution images without using many system resources and/or unnecessary bandwidth in the transmission of the images data. Another objective of the project is that the usability of the user interface be as simple as possible with a dynamic clean interaction to ease the user's actions and improve performance. Another objective was to develop the tool as a platform independent tool allowing it to run in any computer with any operating system in any place of the world.

1.3 Contributions

The main contributions of this work are: (1) the creation of a new user friendly, web-based, and platform independent GIS tool for the creation of boundaries (polygons) of geographical areas; (2) the creation and edition of multiple boundaries on top of an image view, letting the user view and compare boundaries and/or create new boundaries using previous boundaries as reference; and (3) the creation and management of KML files to support data sharing of the boundaries between different GIS applications available on the web.

1.4 Thesis Outline

The organization of the rest of this thesis is as follows. Chapter 2 presents a review of the literature related to the project presented in this thesis. Chapter 3 presents the description of the application. Chapter 4 presents a usability study conducted for the application and the discussion of the results. Chapter 5 presents the conclusion and future work of our GIS tool.

Chapter 2

RELATED WORK

2.1 Web-based GIS Applications

In an article by Tara Athan [Athan07], she writes about Web-based GIS applications. She points out that a web-based Geographical Information Systems (GIS) application can reach many more users at low cost by using the Internet. Contrary to the traditional GIS shop where large format printers are available for hard-copy output, allowing a larger area to be viewed at high resolution, a web-based GIS application uses a monitor to display high-resolution images by using zoom and pan controls. This allowed the user to view a larger area at low resolution and zoom in for high resolution details. A big difference from the old GIS shop to the new one is the vast availability of geographic data on the Internet. Another important difference is the price aspect. A web-based application is cheaper to maintain economically than commercial software.

Athan also mentions that KML files are gaining popularity for the publishing of geographic data [Athan07]. GIS introduce methods and environments to visualize, manipulate, and analyze geospatial data. The nature of the geographical applications requires seamless integration and sharing of spatial data from a variety of providers [Sayar05].

Our Web-based GIS application currently exports and imports KML files in order to export the spatial data created within the application and is able to import data from other applications using the same format. KML files, an open standard by the Open Geospatial Consortium (OGC) officially named the OpenGIS® KML Encoding Standard (OGC KML) is used in many other applications, including Google Earth, Google Maps, Google Maps for mobile, NASA WorldWind, ESRI ArcGIS Explorer, Adobe PhotoShop, AutoCAD, and Yahoo! Pipes [Google08].

2.2 Usability of GIS Applications

In a research by Traynor and Williams [Traynor95] to find out why are geographic information systems hard to use, GIS experts gave a group of social scientists a tutorial in order to use a GIS application with residents in inner-city neighborhoods of Lowell, MA. The application had data about demographics, schools, crime, hospital emergency use, and other data. Despite the tutorial, the social scientists found the software very difficult to use.

On two additional projects in the same research, people with more computing experience were chosen for a test with a GIS application. Computer science graduate students were chosen to use various off-the-shelf GIS software packages as part of their graduate HCI (Human Computer Interaction) class work. Using two GIS packages Macintosh-based and five PC-based they watched how computer science graduate students spent as much as an hour trying to figure out how to open a map in one particular GUI-driven GIS, and gave up in frustration. These tests show us the importance of usability testing in the development of Graphical User Interfaces. This short paper showed the need for designing new Graphical User Interfaces improves user interaction with GIS applications [Traynor95].

2.3 Implementation of GIS Applications

The use of JavaScript in the implementation of GIS applications is necessary. JavaScript is already integrated into all the big Internet browsers like Firefox®, Safari, Opera, and even Internet Explorer that has his own version of JavaScript called Jscript, just to mention a few examples. Users do not need to download any additional software in order to start using one of the most important aspects of it, which is that it is fast and it is becoming faster every year with new improvements. Because JavaScript code can run locally in a user's browser (rather than on a remote server), it can respond to user actions quickly, making an application feel more responsive. Furthermore, JavaScript code can detect user actions, which HTML alone cannot, such as individual keystrokes. Applications such as Gmail take advantage of this. Much of the user-interface logic of this application is written in JavaScript, and JavaScript dispatches requests for information (such as the content of an e-mail message) to the server [Wiki08].

On a report by Stephen Shankland of CNET News Shankland says, in his first reason, why it matters to use JavaScript is that "Internet companies have found that even small improvements in Web page responsiveness increases user interaction with their sites." His second reason states that JavaScript powers many sophisticated Web sites and Web-based applications, endowing them with features such as drag-and-drop, and pop-up dialog boxes. Faster JavaScript means companies can build features into the Web applications and that users will find those applications easier to use. His last reason is that JavaScript is standing strong beside their competition, Microsoft®'s Silverlight, Adobe® Systems Flash® and Flex® technology, and applications that run directly on the PC, such as Microsoft Office. [Shankland08] Fast interaction, easy access, and current and future speed and feature enhancement with the use of JavaScript are enough justification to improve the usability of any type of Web-based application today and in the future to come. The GIS application Graphical User Interface described in Chapter 3 was coded using JavaScript.

AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) is another important development tool for Web-based applications. AJAX is a style of web application development that uses a mix of modern web technologies to provide a more interactive user experience. It uses several technologies, which come together and incorporate to create a powerful new model like JavaScript, HTML, Cascading Style Sheets (CSS), Document Object Model (DOM), XML, and XSLT, and XMLHttpRequest as messaging protocol. Web Services also use widely-used and well-known technologies such as XML and HTTP as AJAX does. Since

AJAX and Web Services are XML based structures they are able to leverage each other's strength [Sayar06].

These core technologies forming AJAX are mature, well known and used in web applications widely. AJAX became so popular because it has a couple of advantages for the browser-based web applications developers. AJAX eliminates the stop-start nature of interactions, user interactions with the server happen asynchronously, data can be manipulated without having to render the entire page multiple times in the web browser, and requests and responses over the XMLHttpRequest protocol are structured XML documents. This enables developers to easily integrate AJAX applications into Web Services [Sayar06]. The application described in the following chapter was developed using AJAX.

GIS services will be implemented more extensively by using the Web service approach. A spatial data infrastructure lets many GIS vendors share data stores and applications in a distributed net environments, making it easier for application developers to integrate geospatial functionality and data into their custom applications. They describe how the Open Geospatial Consortium (OGC) defines a number of standards, both for data models and for online services that have been widely adopted in the GIS community. [Sayar06] As mentioned previously, our web application is using an Open Geospatial Consortium (OGC) standard called KML.

Tezuka, Kurashima, and Tanaka [Tezuka06] discuss the directions available for tighter integration of Web search with a GIS, in terms of extraction, knowledge discovery, and presentation. They also describe implementations to support their argument, which is that the integration must go beyond the simple map-and-hyperlink architecture.

They discuss the integration of a Web search with geographic information and mention that there are a number of local Web search systems enabling users to find location-specific Web content, but in a superficial level by linking Web content to a map interface. By integrating Web search with GIS we can increase the amount of data involved. The most significant integration methods are Linking, Extraction, Knowledge Discovery, and Presentation [Tezuka06].

Linking is the simplest integration of Web search with a GIS by linking a Web page to a geographic point or region. Such integration is already implemented in many existing local Web Search systems, that is why the focus of their discussion is on the other three integration methods. Extraction is the process of getting relevant pieces of geographic information from a Web page increasing user convenience by eliminating unrelated parts. This extracted data can be later used for various applications. Knowledge discovery is the extracted information from Web and GIS knowledge discovery into knowledge on geographic space. Aggregation created knowledge that is not available in a conventional GIS [Tezuka06].

Presentation is when URLs and snippets are presented along with a map interface, which is the most common presentation scheme on local Web search systems [Tezuka06].

Our boundary tool is part of the extraction method were another application can get boundary data from our tool and serve as a source of information for other types of applications. Our tool can map areas that could be use in other applications. For example, a researcher maps all the bays of an area and those bays could be information for another research involving lake locations or even sensor location in lakes.

There are two major Internet GIS applications: server-side applications and client-side applications. Server-side applications rely on a GIS server (usually reside on a remote server) to perform all GIS analysis, while client-side applications perform GIS analysis and processing on the Web browser on the user's local machine [Peng97].

In server-side Web-Based GIS, because the process is focused on the server side, the system load increases rapidly while the client user increases, thereby performance of system will fall rapidly. In addition, it is difficult to provide high interactive geographic information such as space analysis information. Considering these complication, it is meaningful to propose the client-side Web-Based GIS [Wei99].

In the paper they propose a method for efficient spatial data transmission using tiles in order to make data transmissions more efficient using a client-side Web-Based GIS. They mention that due to the huge size of data transmission, enormous diversity in the transmission speed of data on the Internet and local bus of computer, the Web-Based GIS has a main drawback, which is the low performance of data transmission. Usually this problem has not been considered in traditional desktop GIS. It takes a long time to initially download GIS plug-ins, ActiveX controls, JAVA applets and map data [Wei99].

Generally, the low performance of Web-Based GIS can be solved in two ways: 1) increasing the speed of Internet connection and 2) developing more efficient program [2] [Wei99]. While designing our boundaries tool we took this bandwidth and speed problems into consideration, and tried to build a more efficient web application. Our tool manages the boundary creation and editing within the client-side's user interface, inside the browser using JavaScript, freeing bandwidth by reducing communications with the server and processing time. Some tasks need to be processed in the server-side using PHP5 and its results are sent back to the JavaScript User Interface by using AJAX, reducing bandwidth use and time.

In order to improve bandwidth usage even more, we decided to use a tile method to download images to the user's browser. Only the tiles of the image that are currently on the user's screen are the ones that download to the browser. This technique uses bandwidth more efficiently than downloading the entire image in order to work using only part of the entire high resolution image.

Chapter 3

A TOOL FOR MAPPING IMAGES TO GEOGRAPHIC AREAS

This chapter describes the operation and details of the development of a web-based application capable of the creation of boundaries using spatial data contained in aerial and/or satellite images. Section 3.1 describes the functional requirements of the application. Section 3.2 describes the operation of the application and its user interface. Finally, section 3.3 describes the implementation details of the application.

3.1 Functional Requirements

This application is a Web-based GIS tool capable of displaying a georeferenced image. The tool uses the georeferenced data from the image to georeferenced an image view on the JavaScript GUI. At this point, the user is able to use multiple tools to create and customize a new boundary created by the user or an imported boundary. The user can use older boundaries as reference to create new ones.

When the user finishes the creation and/or editing of the boundary, he/she is given the option of saving the boundary to the local database for use locally in the WALSAIP project and/or later export the boundary for use in another GIS tool. The resulting boundary is georeferenced and will be able to display on any GIS application that accepts KML files.

3.2 Application Description

The first stage of the development of the web tool was to create a design of the user interface. Multiple designs came to the drawing board and even the last one experienced multiple changes. The image below shows the final design after multiple iterations to the user interface, as shown in Figure 3.2.1. A web user interface capable of creating boundaries, while at the same time, displaying boundaries and storing its descriptive information in a database. All this was accomplished without requiring the reloading of the websites' URL *(Uniform Resource Locator)* to perform user's actions like edit, move, or delete a point inside the boundary.

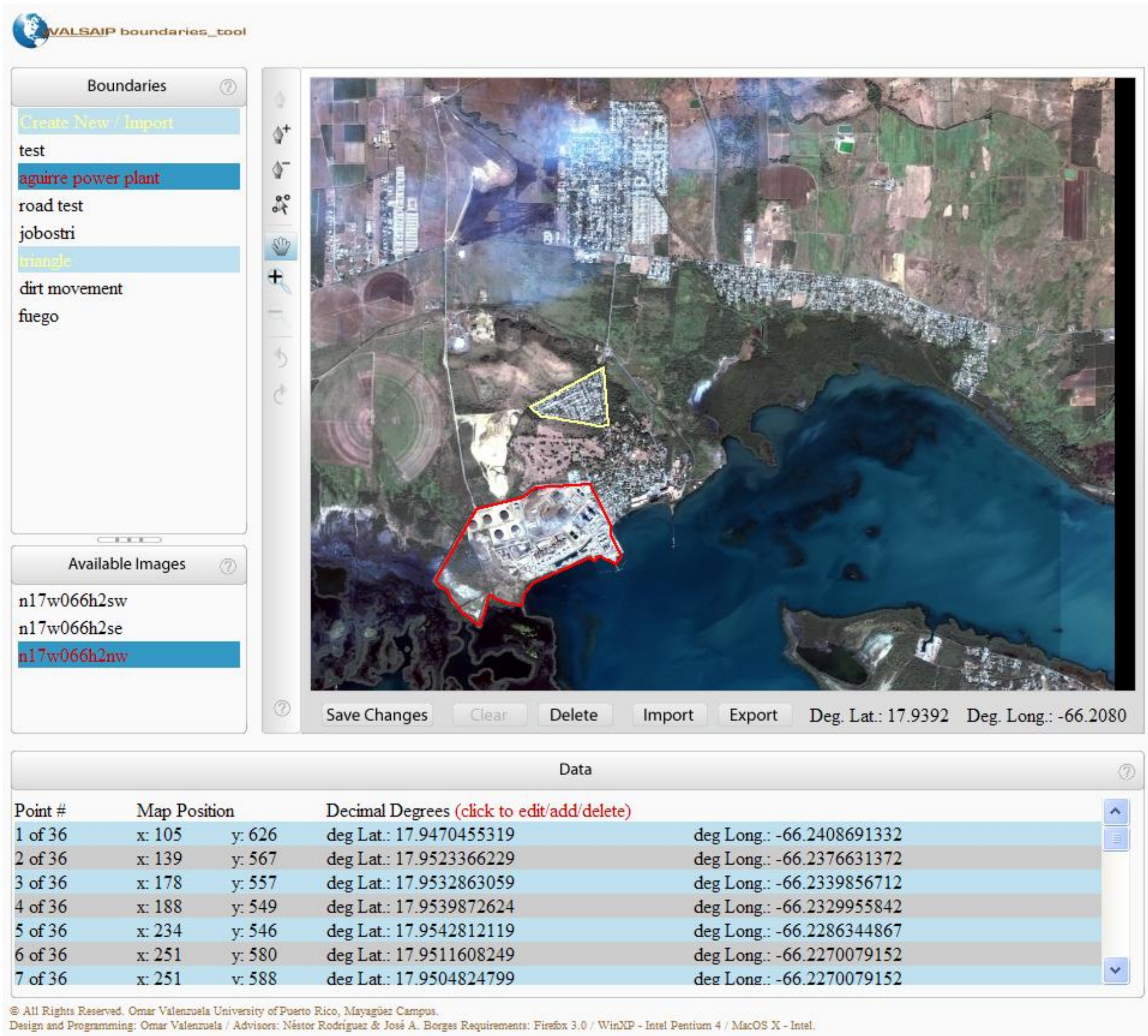


Figure 3.2.1 Complete view of the Boundaries Tool's web-based user interface

The window of the image area shown in Figure 3.2.1 is where most of the user's actions take place. This section manages the user's interaction when he/she starts to create a new boundary from scratch, modify a boundary, save, and import and export of boundaries. All the events that take place in this area are constantly being georeferenced in real-time by following the images' World File data.

The tools at the left of the image area interact with the image area by changing the behavior of the mouse actions according to the intention of the user. All the tools have tool tips to recall the user about the actions that each button performs.

As you can see on Figure 3.2.2, the user can create a new boundary using the Create a boundary tool and later modify that boundary by using one of the tools to add, remove, or move a point (Figure 3.2.3). At the same time, the user is capable of dragging the image currently on the screen by using the hand tool for dragging the image. The user also will be able to zoom in or zoom out the image. Also, for the mistakes there are options to undo and redo the actions made to the boundary points. The actions of these buttons are presented dynamically to the user, making the events easy to follow giving the user useful feedback of his/hers actions.

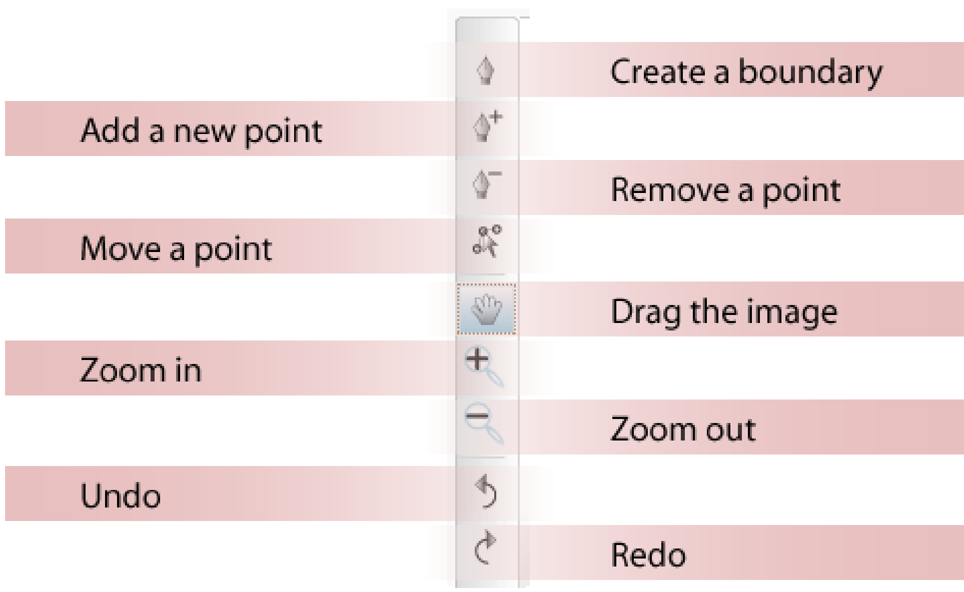


Figure 3.2.2 Tools

Figure 3.2.3 Editing a Boundary

At the bottom of the image area, there are a number of buttons shown in Figure 3.2.4. The first button from left-to-right controls the saving and/or editing of new and current boundaries. The button changes according to boundaries status. If the user is creating a new boundary or is importing a new boundary to the database, this button takes the action of saving the boundary and changes its text name to "Save Boundary". If the user is editing an existing boundary, the button takes the action of saving the changes made to the boundary by changing its name to “Save Changes”.

The second button called “Clear”, clears all the changes made to the current boundary being shown inside the image area. If a new boundary is created, a click on the clear button clears all the changes made to the boundary starting on the creation to the last point made in this new boundary finishing with the complete deletion of the new boundary. If the user is editing an existing boundary from the local database, only the changes made starting from the time the boundary was active are deleted from the screen’s data.

The third button called “Delete” only works with boundaries that are currently stored inside the local database. A click on this button takes you to a confirmation screen in order to delete the boundary from the database completely.

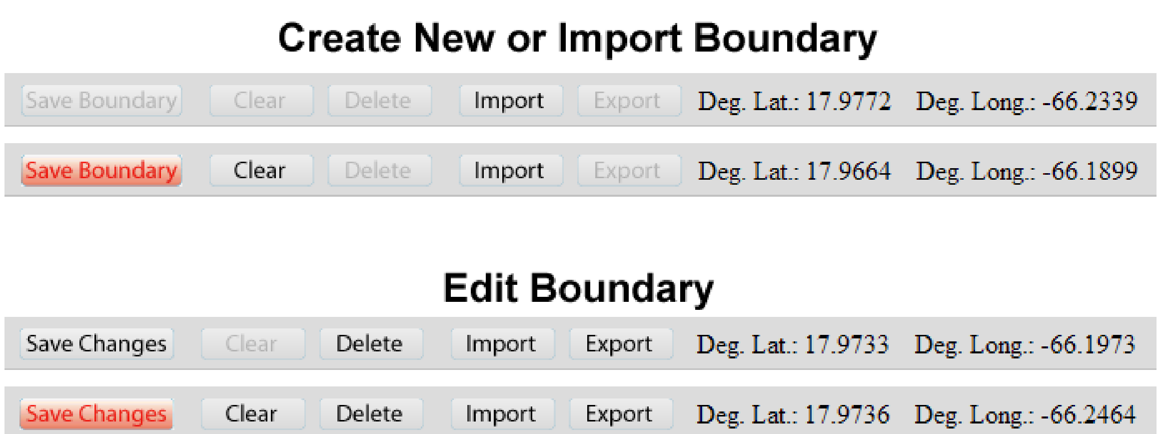


Figure 3.2.4 Image Area Action Buttons

The fourth button called "Import" works by showing the import window to the user when clicked, as shown in Figure 3.2.5. The "Create new/import" boundary name will activate to receive the new imported boundary when it is uploaded to the browser. The imported boundary is not saved into the local database until the Save Boundary option is used. The "Import" window reads XML files using the KML standard. If the KML is composed of multiples boundaries, the software will ask the user to choose one of the multiple boundaries to be imported into the application and the user can import each boundary separately as shown in Figure 3.2.6.

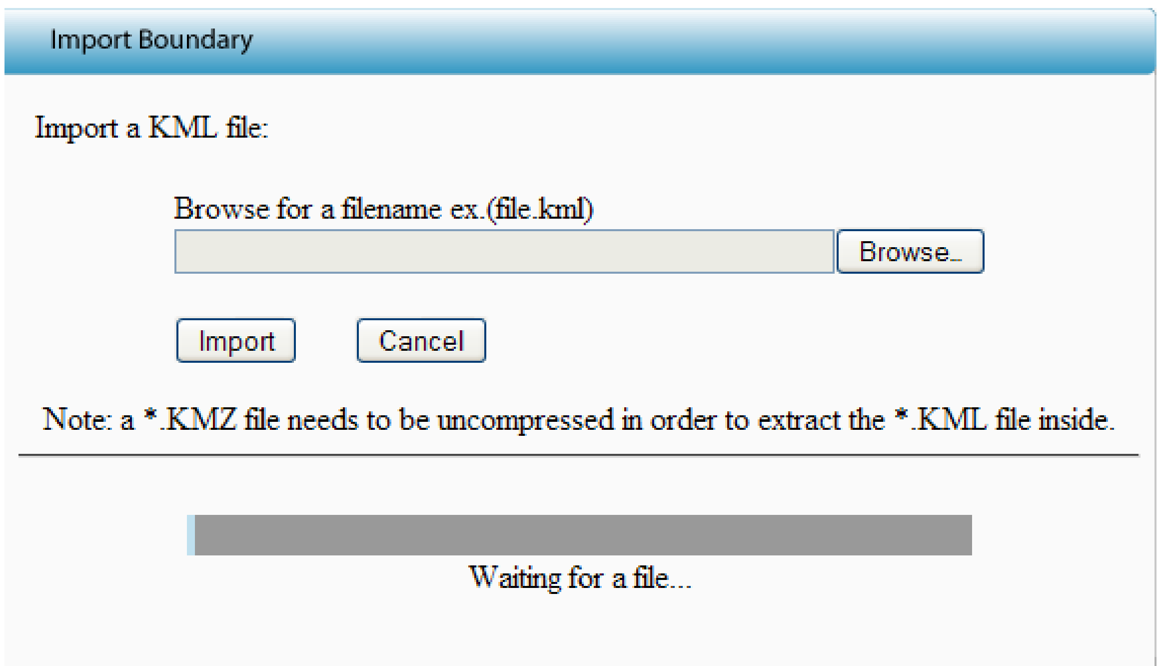


Figure 3.2.5 Import Window

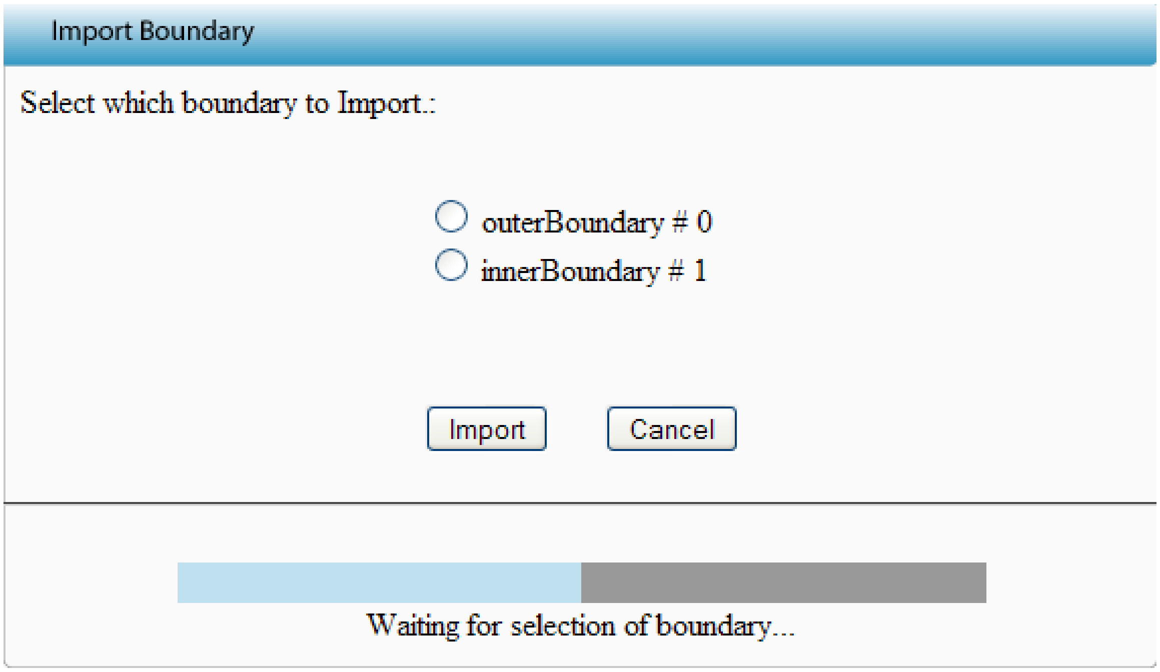


Figure 3.2.6 XML with Multiple Boundaries

The fifth button called “Export” only works when editing a boundary. A new window for exporting a boundary shows up. This window takes all the points that conform the boundary using the database information stored and creates a XML file using the KML standard so that the file is compatible with other software, like for example, Google Earth.

The “Data” window shows all the points currently inside the local database as shown in Figure 3.2.7. They are organized in ascending order, starting with the first point created, their X (horizontal pixel), Y (vertical pixel) position on the computer’s screen map, and the georeferenced position in decimal degrees (Latitude in degrees and Longitude in degrees).

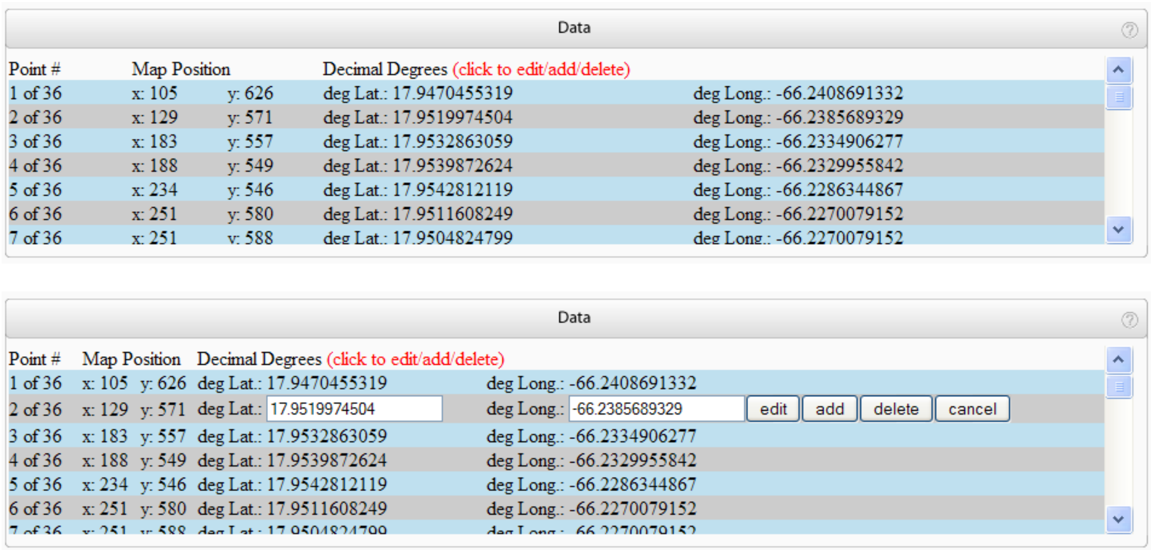


Figure 3.2.7 Data Window

The data of only one boundary is shown in the Data window. All the points shown on the screen that corresponds to the boundary activated in dark blue on the "Boundaries" window. All the points displayed in the Data window are clickable to show the edit, add, and delete options. You can edit, add, or delete only one point at a time. This feature is available for users that want to enter data manually to the database.

Changes made to a point are not stored in the database in real-time. Changes are shown inside the map window and the user has the option of saving all the changes made to the current boundary by using the "Save Changes" button located at the bottom of the image area window. A red "Save Changes" button alerts the user of changed data inside the boundary waiting to be saved permanently into the database.

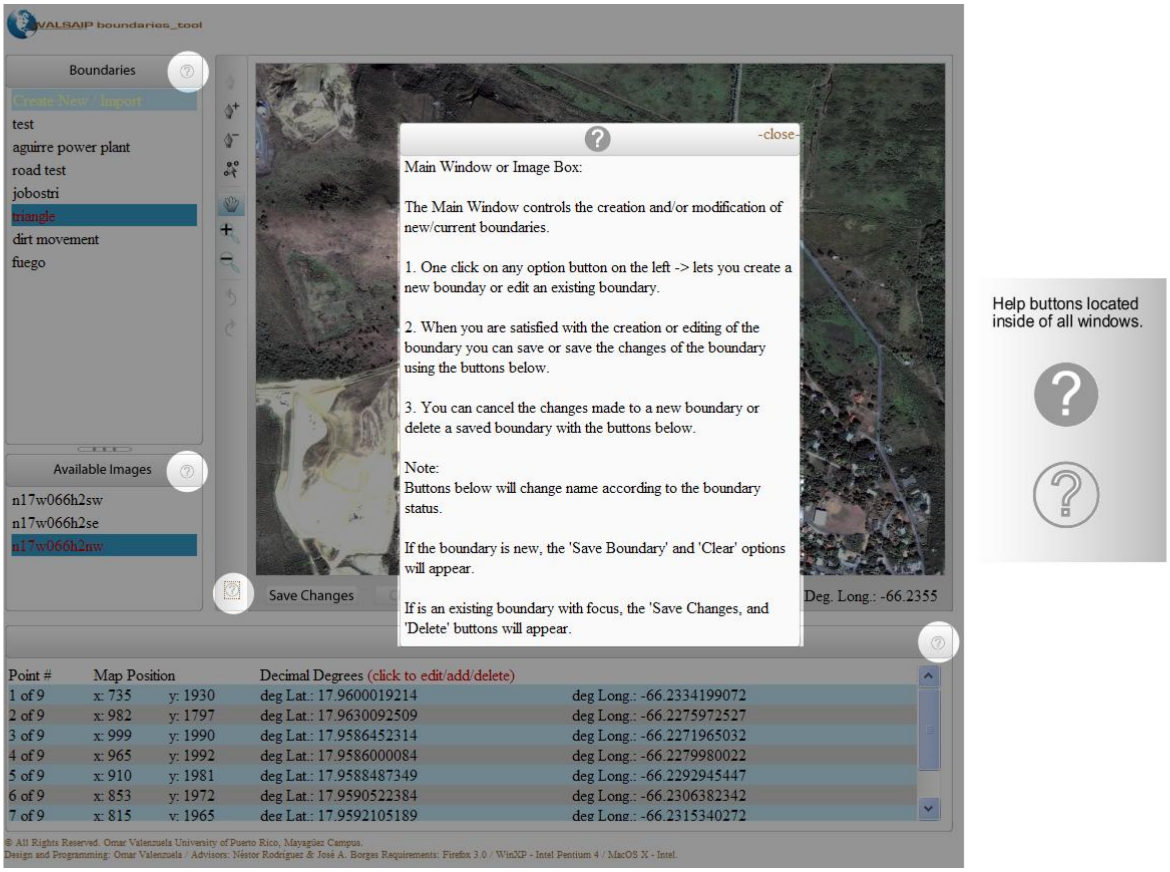


Figure 3.2.8 Help Buttons

All the windows inside the tool have a help button located at the upper right corner of the corresponding window. A question mark image like the one shown on Figure 3.2.8 shows where the help buttons are located. A click in one of the help buttons will bring up a new window with instructions explaining how the tools inside the specific window work.

3.3 Implementation Details

The goal of this research was to create a completely new user interface capable of creating spatial data by using georeferenced images as backgrounds. All the boundary information was stored inside a local database and made accessible through a local website to avoid the use of proprietary APIs (*application programming interface)* that we do not have control over its modifications and could limit future development of our tool. The first step in the creation of this tool was to find new ways of presenting the information to the user via a user interface and create boundaries using a JavaScript run interface in which all the information created by the tool could be easily stored and accessible by existing and new future software applications in the market.

A user interface capable of managing images in the background and boundaries was created. The creation of the boundaries performed inside our web tool is capable of interacting with the user's actions in real-time, without refreshing/reloading or switching from the current web page to perform different tasks. This was done by using AJAX techniques being used more frequently in the web these days. The information gathered from the user's work needed to be in an open format available to anyone and that is where KML, an XML-based language became part of the web tool.

With AJAX, we can merge the dynamic content obtained from dynamic HTML and CSS, with the coding support of PHP 5 and manage the user interface's dynamic information display with the use of JavaScript. In this case, we are using SAJAX (**S**imple **AJAX**

Toolkit) [SAJAX06] as the AJAX tool to carry the communications between the JavaScript and PHP 5.

JSON (**J**ava**S**cript **O**bject **N**otation) which is now included in version 5 of PHP is used to carry the variable data from/to JavaScript that is sent over to the PHP 5 to be displayed in the user interface. **JSON** is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language [JSON07].

This combination of applications (JavaScript/HTML/CSS, SAJAX, PHP, MySQL®, Apache), also called Web 2.0, allows us to create dynamic content, similar to what users experience when using desktop application, but in a web environment together with all its flexibility and more advantages like being able to work from any computer as shown in Figure 3.3.1 [Paulson05].

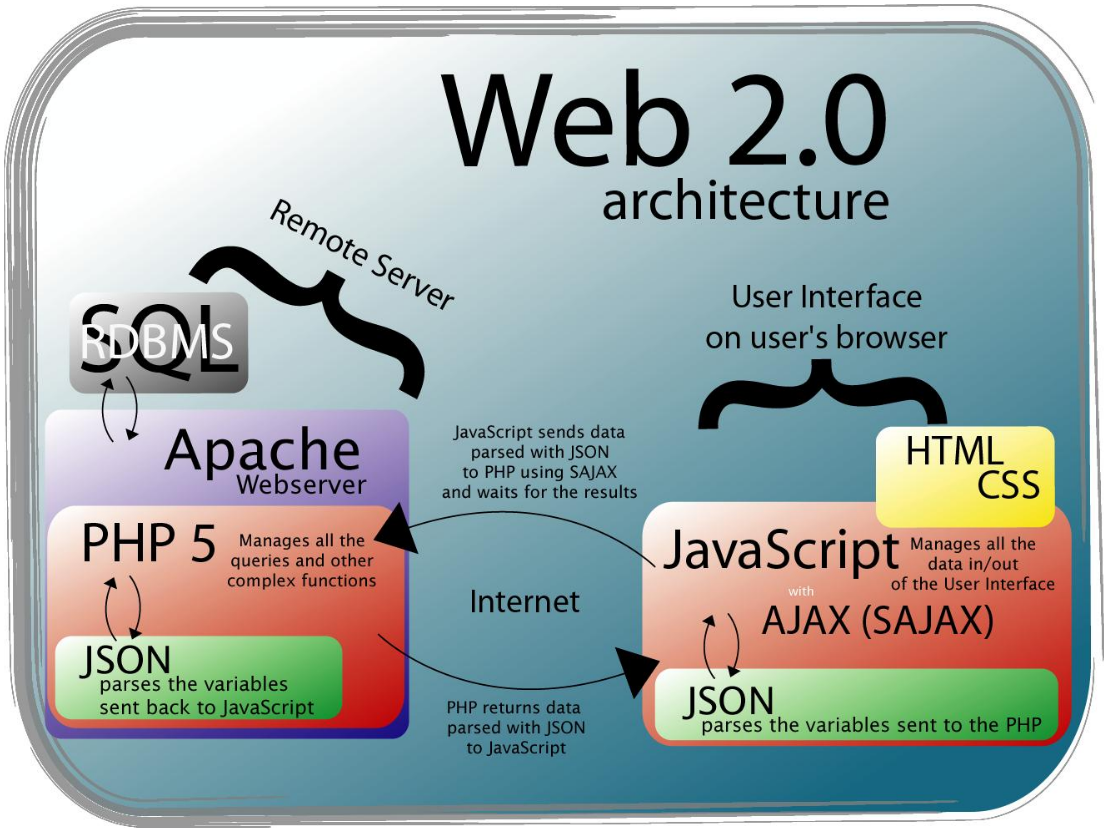


Figure 3.3.1 Web 2.0 Architecture

Our application was developed for Mozilla Firefox version 2 and future version 3 because of its stability, standard compliance, and performance of JavaScript at the time of the development of the application. Programming for only one browser makes the process of developing a web application a lot faster. This decision turns out to be a good one, because when Firefox 3 was released in summer 2008 our tool's actions improved dramatically in performance. Fortunately, the tool also performed and rendered correctly when using Safari's Apple Internet browser.

In the server side of the tool, we are using Apache, MySQL®, and PHP 5 to manage queries, image files, and store information inside the database. All these server side tools are available on the Internet, making the implementation of applications very easy in any type of work environment.

3.4 Interaction with Other Application

As part of the WALSAIP project, our boundary tool is capable of accepting GET links from other web pages, which is encoded browser data inside the URL.

The boundary tool accepts information from other web pages that target information currently stored inside a local database of boundaries and images. Another WALSAIP tool called TerraServer [DeLaPaz08], developed by William De La Paz, uploads new georeferenced and not-georeferenced images to the database, and performs dynamic searches of them organizing the image data. The result of a search can contain images from the local database or from other GIS databases available for public access on the Internet and the boundaries currently stored in the local database.

In the search results, the users have the option of sending an image or a boundary to the Boundaries Tool with just a click of the mouse and start to work with it as soon as it loads up. Here are two scenarios explaining the process with an image and with a boundary.

The first scenario is that a user wants to use an image that he/she found on the local database using the TerraServer tool and wants to create a boundary using that image as a background. In this case, the user only clicks on the image and later clicks on the boundary tool button. The TerraServer application sends the data using the GET method to the boundary tool with data that targets the corresponding image selected by the user. Later, as the boundary tool loads up in a browser window detects the GET method being active and finds the image requested by the TerraServer tool, displaying it as the background image, all ready and set for the creation of a new boundary.

The second scenario is that a user wants to use a boundary that he/she found using the TerraServer tool and wants to take a close look at it or edit the boundary. In this case, the user only clicks on the boundary link shown in the TerraServer results and the tool sends data using the GET method to the boundary tool containing the boundary code number corresponding to the boundary selected by the user before. Later, as the boundary tool loads up in a new browser window it finds the boundary requested and the corresponding background image used in the original creation of the boundary.

All the scenarios run automatically without the intervention of the user. The user only has to select an image or a boundary and both tools interact with each other to accomplish the task.

Chapter 4

USABILITY STUDY

4.1 Methodology

The purpose of this usability study is to determine the ease or difficulty of use of the boundaries tool application by potential real users as well the satisfaction of the users with the application. The study was performed using a local web server to prevent any network related variability in the results of the study. The computer used to run the test was a PC with an Intel Core 2 Duo @ 2.16 GHz processor, 2 GB RAM and 7200 RPM HardDisk. The software used to run the application was Apache 2.2, PHP 5.2.6, MySQL® Server 5.0. The browser used for the test was FireFox® version 3.0.1

The users selected for the usability test were ten Civil Engineering and Surveying students from the University of Puerto Rico, Mayagüez Campus, with experience in the creation of polygons and/or boundaries with any type of GIS tool. Four of the students were females and six were males. Their ages were between 21 to 23 years. They were approximately in the middle of their bachelor programs. All of them had more than 6 years of experience using computers and on a typical day, they used computers for more than three hours.

Prior to the test, the users were asked to fill a questionnaire about their experience using computers and their professional background. Then they were given a short tutorial on how to use the application. After the tutorial, the users were asked to perform 20 tasks using the boundary tool. The tasks were the following:

1. Switch to the image called **n17w066h2nw** using the "**Available Images**" window.
2. Look for a small residential area that has the shape of an inverted triangle at the lower left section of the image. Center the current image view using the hand tool. Ask the tester for help, if necessary to find the area.
3. Zoom-in and get a comfortable view of the residential area.
4. Create a boundary using the tools provided at the left of the map area.
5. Add a new point within the residential area points already created.
6. Undo the last action.
7. Redo the last action.
8. Remove one point within the points already created.
9. Move the position of one of the points created.
10. Save the boundary with the following information:
    a. Boundaries Name: **residential triangle**
    b. Country: **Puerto Rico**
    c. State: "leave empty"
    d. City: **Salinas**
    e. Type: **Map**
    f. Description: **near aguirre power plant**
11. Activate the boundary called "**aguirre power plant**".
12. Press the export button to export the boundary to your computer's Desktop by choosing "Save File" when prompted. Close any download confirmation pop-up window that shows up.
13. Import the boundary called "circle" from the computer's Desktop.

14. Use the “Data” window to find point #2, edit its Latitude value to 17.97, and press edit.
15. Clear the actions made to the current boundary.
16. Save the boundary called “circle”, with the following information:
    a. Boundaries Name: **circle**
    b. Country: **Puerto Rico**
    c. State: “leave empty”
    d. City: **Salinas**
    e. Type: **Map**
    f. Description: **near aguirre power plant**
    g. Press Save Boundary when finished.
17. Deactivate the boundary called “aguirre power plant”.
18. Activate the boundary called “residential triangle”.
19. Delete the boundary called “residential triangle”.
20. Delete the boundary called “circle”.

The first three tasks were simple tasks intended for the user to get familiarized with the tool and develop confidence using it. The intention of task 4 was to let the user create a boundary for the first time using the tools located at the left of the map. When the boundary was completed, the participant was asked to edit the boundary. The participant was asked to add a point (Task 5) to the current boundary, undo the action (Task 6), redo the action (Task 7), remove a point (Task 8), move a point (Task 9), and save a new boundary (Task 10). By performing these tasks, we tested the buttons that controlled the creation of a new boundary creation.

The purpose of the next tasks was to test the management of the boundaries already stored in the database. The users activated a boundary already stored in the database (Task 11) and exported its content to the computer's desktop (Task 12). Later it was asked to import a different boundary already saved on the computer's desktop into our boundary tool (Task 13). This set of tasks would test if the participant was able to find a boundary already stored inside the database, activate it on the screen, export a boundary to the local computer for use by another application, and import a new boundary for use in our application.

On task 14, we tested the data window that contained all the points that composed a complete boundary. Here the participant was able to edit the boundary by entering numeric data inside the data window instead of using the mouse to move, edit, and/or delete a point's boundary.

The following two tasks were intended to test the **clear changes** and **save boundary** buttons. We asked the participant to clear the changes (Task 15) made to the boundary on task 14 and then save the boundary called "circle" (Task 16) currently on the screen. On task 17, the participant needed to find the boundary called "circle" already saved on the database and deactivate it so that the boundary no longer appears on the screen.

The final tasks tested the deletion of boundaries from the database. Task 18 asked the participant to activate a boundary called "residential triangle" and task 19 asked the participant to delete that boundary. Then the participant was asked to delete another boundary called "circle".

After performing the tasks, the users were asked to fill a questionnaire about their satisfaction with various aspects of their interaction with the tool. The questionnaire used was the following:

1. Rate the following set of tools actions by their ease or difficulty in their usage.

Circle one number in each line.

	Very Difficult				Very Easy
a. **Switch** an image	1	2	3	4	5
b. **Activate** a boundary	1	2	3	4	5
c. **Deactivate** a boundary	1	2	3	4	5
d. **Create** a boundary	1	2	3	4	5
e. **Add** a new point	1	2	3	4	5
f. **Zoom** in/out tool	1	2	3	4	5
g. **Remove** a point	1	2	3	4	5
h. **Move** a Point	1	2	3	4	5
i. **Hand** tool	1	2	3	4	5
j. **Clear** a boundary	1	2	3	4	5
k. **Save** a boundary	1	2	3	4	5
l. **Import** a boundary	1	2	3	4	5
m. **Export** a boundary	1	2	3	4	5
n. **Delete** a boundary	1	2	3	4	5

2. Overall, how do you rate the performance of the application you've just tested?

Very Hard — Very Easy

1	2	3	4	5

3. How useful do you find the application you've just tested?

Not Useful — Very Useful

1	2	3	4	5

4. Comments and/or Suggestions about the boundaries tool:

____

The dependent variables of the study were the time to complete the tasks, the number of tasks completed and the user subjective satisfaction.

The participant's interaction with the computer was recorded using TechSmith® Morae® software. This software was able to capture the real participant's interaction with our mapping tool in our usability study. The software consists of three applications: Morae Recorder, Morae Observer, and Morae Manager. Morae Recorder records the onscreen actions of the users performing the tasks in the study, including actions from their keyboard and mouse. Later, the software saves the recording in a file format that can be opened in Morae Manager. Morae Observer allows an investigator to connect to Recorder over a network and watch, log markers and tasks using another computer while the test is in process. Morae Observer was not used in this study. Morae Manager allows the creation of projects, opening and analyzing recordings logging markers and tasks, creating graphs and video to present the results.

4.2 Results

This section presents the analysis and summary of the usability study. Table graphics for the complete time results task-by-task is available in Appendix A:. The following Table 4.2.1 has the time in seconds to complete the tasks performed in the study.

Table 4.2.1 Time to Complete the Tasks

	Time to Complete a Task (seconds)											
	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	Mean	Std. Deviation
Task 1 Switch an Image	7.7	3.5	11.6	13.0	9.0	10.8	8.1	8.0	3.1	2.8	7.8	3.6
Task 2 Hand tool	7.1	31.6	11.5	56.7	36.1	56.7	5.6	15.4	4.8	4.6	23.0	20.9
Task 3 Zoom-in	46.5	44.1	13.3	24.5	2.8	22.8	10.5	16.8	7.8	10.7	20.0	14.8
Task 4 Create boundary	33.9	61.0	55.0	15.9	21.5	22.1	12.7	25.7	34.0	37.6	31.9	15.9
Task 5 Add point	17.7	14.7	5.1	34.5	13.0	14.5	8.3	11.7	12.9	10.4	14.3	7.9
Task 6 Undo	5.1	5.5	7.2	8.9	2.0	8.9	2.7	8.9	3.2	2.6	5.5	2.8
Task 7 Redo	2.7	5.3	2.3	5.3	2.0	5.3	3.1	3.7	2.7	1.5	3.4	1.4
Task 8 Remove point	6.4	5.3	6.5	9.7	6.2	5.6	4.5	7.3	5.6	5.5	6.3	1.4
Task 9 Move a point	13.2	10.4	10.4	35.8	11.2	11.7	12.8	11.1	6.8	13.7	13.7	8.0
Task 10 Save boundary	49.9	68.8	51.2	48.8	39.8	83.5	65.2	64.1	58.5	63.5	59.3	12.4
Task 11 Activate boundary	5.8	9.7	10.9	3.5	2.5	5.4	3.7	3.5	4.2	4.4	5.3	2.8
Task 12 Export boundary	37.1	43.8	36.6	16.4	44.5	32.7	8.7	49.8	23.9	16.4	31.0	13.9
Task 13 Import boundary	30.2	27.6	27.9	30.7	25.8	25.9	30.0	20.6	25.3	17.7	26.2	4.2
Task 14 Edit data window	29.6	29.8	36.9	32.2	33.2	30.3	51.2	32.7	28.9	21.6	32.6	7.6
Task 15 Clear button	19.4	7.7	19.8	4.3	12.2	15.4	19.8	6.2	7.6	5.5	11.8	6.3
Task 16 Save boundary circle	47.0	43.1	39.6	38.8	44.6	52.7	52.7	32.1	46.1	43.3	44.0	6.3
Task 17 Deactivate boundary aguirre power plant	5.7	53.1	5.0	10.0	5.8	8.9	9.0	3.8	8.0	53.1	16.2	19.5
Task 18 Activate boundary residential triangle	4.0	17.1	3.6	2.7	3.3	6.3	2.4	10.8	6.2	6.2	6.2	4.6
Task 19 Delete boundary residential triangle	11.2	9.7	7.2	6.1	7.1	8.8	5.1	5.7	6.7	11.4	7.9	2.3
Task 20 Delete boundary circle	8.1	7.5	8.1	6.7	7.9	7.8	8.3	8.2	9.2	6.7	7.8	0.7
Total Time	388.2	499.2	369.5	404.4	330.5	435.9	324.3	345.9	305.5	339.3	374.3	59.4

Figure 4.2.1 presents the mean and the standard deviation of the tasks completion times. Overall, it took the users and average of 374.3 seconds to complete all the tasks with a standard deviation of 59.4 seconds.

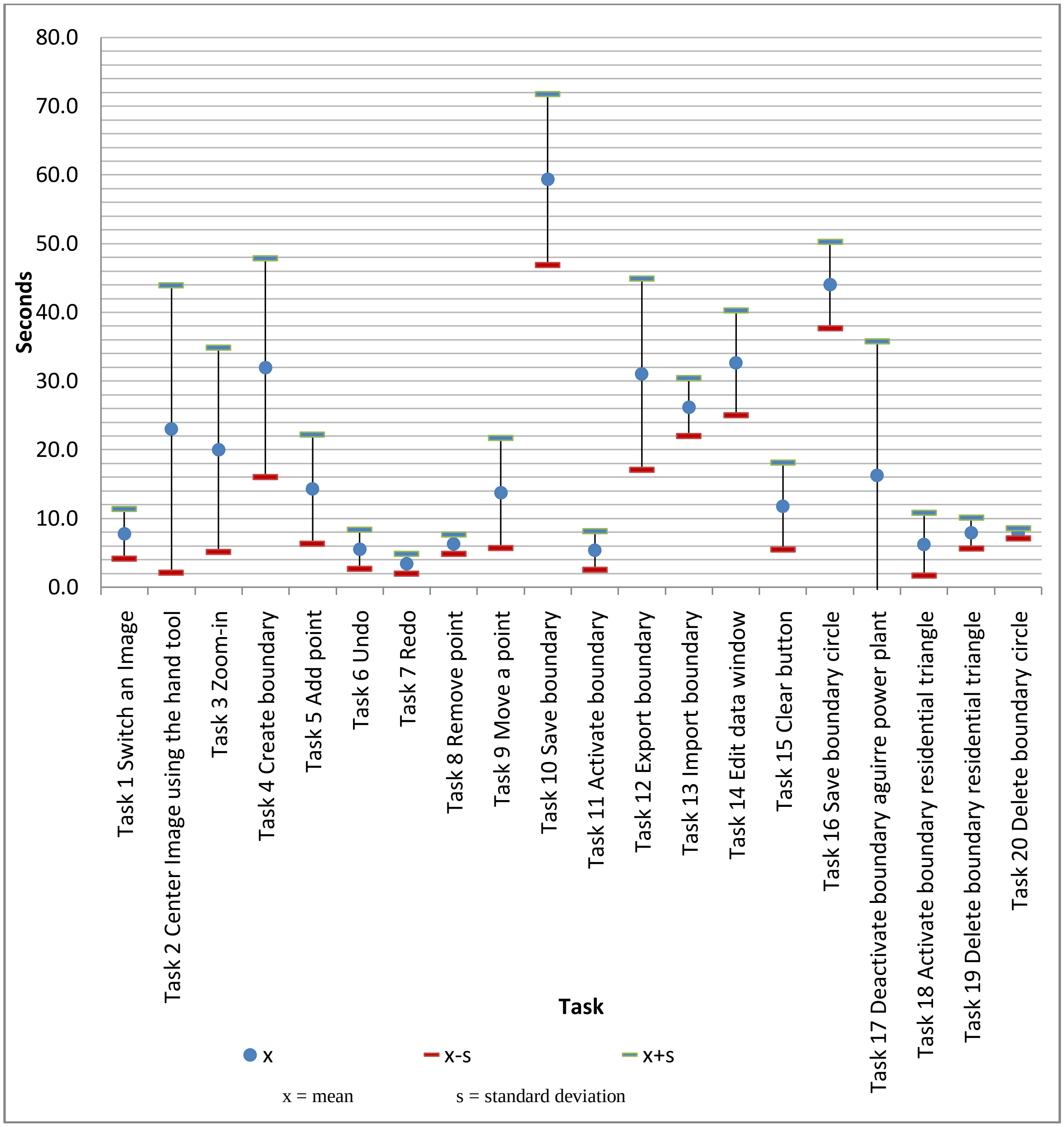


Figure 4.2.1 Task Completion Times

Saving a boundary was the most time consuming task. The first boundary saving task (Task 10) took an average of 59.3 seconds while the second (task 16) took 44.0 seconds. These results show a learning effect on the boundary saving tasks. The least time consuming tasks were: switching an image (Task 2), undo (Task 6), redo (task 7), remove point (Task 8), activate boundary (Task 11 and Task 18), and delete a boundary (Task 19 and Task 20). In general, standard deviations for average completion time of individual tasks were less than the magnitude of the average completion time of the task.

Table 4.2.2 summarizes the number of tasks that were not completed by the users. Only eight out of 200 were not completed, this constitutes a 96% task completion time rate. One of the users was not able to complete three tasks, another was not able to complete two while three were not able to complete one task.

Table 4.2.2 Tasks Not Completed

	Tasks Not Completed (in red)									
	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
Task 1 Switch an Image										
Task 2 Hand tool				■						
Task 3 Zoom-in										
Task 4 Create boundary										
Task 5 Add point										
Task 6 Undo				■		■				

	(continued) Tasks Not Completed (in red)									
	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
Task 7 Redo				■		■				
Task 8 Remove point										
Task 9 Move a point										
Task 10 Save boundary										
Task 11 Activate boundary										
Task 12 Export boundary										
Task 13 Import boundary										
Task 14 Edit data window										
Task 15 Clear button			■							
Task 16 Save boundary circle							■			
Task 17 Deactivate boundary aguirre power plant		■								
Task 18 Activate boundary residential triangle										
Task 19 Delete boundary residential triangle										
Task 20 Delete boundary circle										

The following table presents the results for the Post-Test questionnaire. Graphs for individual aspects of the questionnaire are presented in Appendix B:. A summary of the average rating given to each of the items of the questionnaire is presented in Figure 4.2.2. The results indicate a very high rating for all the items of the post test questionnaire. All the average ratings are on or above 4.6.

Table 4.2.3 Post-Test Questionnaire Results

	Post-Tests Questionnaire (1 = hard and 5 = very easy)															
	Switch an Image	Activate a boundary	Deactivate a boundary	Create a boundary	Add a new point	Zoom in/out tool	Remove a point	Move a point	Hand tool	Clear a boundary	Save a boundary	Import a boundary	Export a boundary	Delete a boundary	Rate the overall performance	How useful did you find the application
User 1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
User 2	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
User 3	5	5	5	4	5	5	4	5	5	5	5	5	5	5	5	5
User 4	5	5	4	4	5	5	4	4	5	5	5	5	5	5	4	4
User 5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
User 6	4	4	3	3	5	5	5	5	5	4	3	3	4	4	4	5
User 7	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5
User 8	5	5	5	5	4	4	5	5	4	5	5	5	5	5	4	5
User 9	5	5	5	5	5	5	5	5	5	5	5	5	3	5	5	4
User 10	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Mean	4.90	4.90	4.70	4.60	4.90	4.90	4.80	4.90	4.90	4.80	4.80	4.80	4.70	4.90	4.70	4.80
Standard Dev.	0.32	0.32	0.67	0.70	0.32	0.32	0.42	0.32	0.32	0.42	0.63	0.63	0.67	0.32	0.48	0.42
Minimum	4	4	3	3	4	4	4	4	4	4	3	3	3	4	4	4
Maximum	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

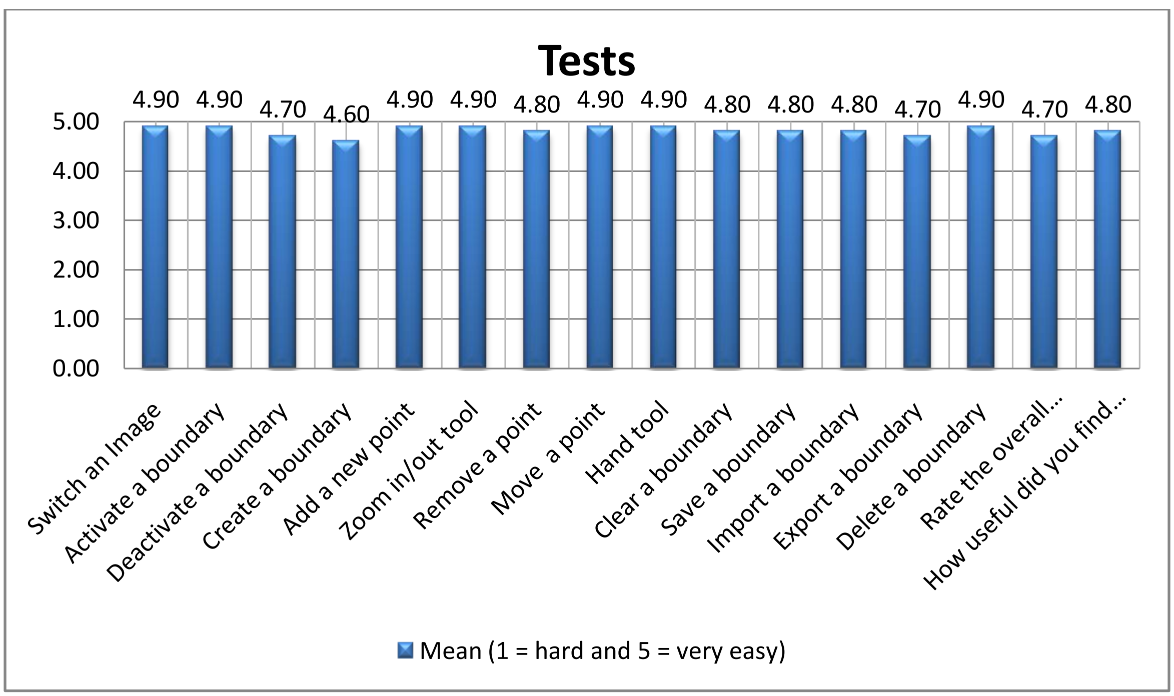


Figure 4.2.2 Average Ratings of Interaction Actions

Overall, the comments included in the Post-test Questionnaire were positive. One user said that boundary creation was easier than a program that he/she used for creating boundaries called Survcad from AutoCAD. Another user mentioned that the application was very useful either for Surveying or for Civil Engineering. With the data obtained from our boundary tool, they could get triangulations, and the application is easy to use.

4.3 Discussion

The less time-consuming tasks were task that in general require a click on a button or a point and click action. Saving a boundary was the most time-consuming tasks. This result is expected because this task required text entry in order to specify the necessary information of the boundary to be stored. Text entry usually consumes time and in addition, it is dependent on the user's typing skills.

The Undo and Redo tasks (Tasks 6 and 7) were the only two tasks that were not completed by two users. It seems that for these two users the funtionallity of these two buttons was not well understood. One of the users tried to undo the actions by manually reversing the last task done instead of using the undo button. In task 7 this user attempted to recreate the actions mannually instead of using the redo button. The second user used the undo button to do the redo a task.

The results of the Post-Test Questionnaire revealed that the boundary tool application resulted in general very easy to use. The results also revealed that the users found each the actions tested was easy to perform with the boundary tool application.

Chapter 5

CONCLUSION AND FUTURE WORK

The results of the study support the conclusion that the boundary tool is very easy to learn and to use. With only a fifteen-minute tutorial, the users were able to complete 96% of the tasks. The users perceive the application as one that is very easy to use in general. They found all individual interaction actions tested as very easy to execute.

The dynamic interaction of our Web 2.0 application proved to be a very efficient technique for building a simple user interfaces while at the same time the loading of only parts of the image when needed, made a more efficient runtime with less bandwidth usage while working with high-resolution images.

The most important contribution of this study is the creation of a very easy to use and very easy to learn boundary tool (polygon creator) able to run as web-based application without using APIs to work, that could be access anywhere in the world.

The boundary application uses only decimal degrees (latitude and longitude geographic coordinates) to display georeferenced data about the boundaries. Other types of coordinates available throughout the world could be supported in the future.

Another capability that should be considered in the future is to allow users to set just one point location to a map instead of a boundary. This single coordinate could be used to represent things such as a sensor’s location.

The boundary tool could be modified to allow the user to georeferenciate an image without georeferenciation by assigning multiple points to the new image and create a new World File data file for the image.

Mathematical calculations can be added to the GUI using data available from the geo-referenced position of points on the map. For example it could be possible to determine the distance between outer lines or calculate the area inside a boundary.

The map area’s dynamic interactions that let the user move the map with the mouse hand tool, could be modified to support JPEG 2000 standard. JPEG 2000's flexibility with respect to random access, codestream reordering and incremental decoding is highly exploitable in a networked environment [JPEG07].

References

[Athan07] Athan, T. (2007) Introduction to Web-based GIS, In alt2is.com. Retrieved October 16, 2008, from www.alt2is.com

[DeLaPaz08] DeLaPaz, W. (2008). *AN ADAPTIVE USER INTERFACE FOR SATELLITE IMAGERY AND DATA CLASSIFICATION*, Master's Thesis 2008, University of Puerto Rico Mayagüez.

[Google08] KML Documentation Introduction. (2008, October 16). In *Google Code*. Retrieved October 16, 2008, from http://code.google.com/apis/kml/documentation/

[JPEG07] JPEG 2000 Interactive Protocol. (2007). In *The JPEG committee home page*. Retrieved, October 21, 2008, from http://www.jpeg.org/jpeg2000/j2kpart9.html

[JSON07] "Introducing JSON". Retrieved April 14, 2007, from http://www.json.org/

[Paulson05] Paulson, L. D. (2005). *Building Rich Web Applications with Ajax* (pp. 14-17). IEEE Computer Society, 38, (0).

[Peng97] Peng, Z. (1997) *An Assessment of the Development of Internet GIS* Proceedings of the ESRI User Conference.

[SAJAX06] Simple Ajax Toolkit in *MODERNMETHOD.* (2006) Retrieved April 29, 2006, from http://www.modernmethod.com/sajax/

[Sayar05] Sayar, A., Pierce, M., Fox, G. C. (2005) *DEVELOPING GIS VISUALIZATION WEB SERVICES FOR GEOPHYSICAL APPLICATIONS* ISPRS International Society for Photogrammetry and Remote Sensing Workshop Commission II WG/2 METU, Ankara, Turkey, November, 24-25

[Sayar06] Ahmet S., Pierce M., and Fox G. (2006) *Integrating AJAX Approach into GIS Visualization Web Services* Proceedings of IEEE International Conference on Internet and Web Applications and Services ICIW'06 February 23-25 Guadeloupe, French Caribbean.

[Shankland08] Shankland, S. (2008, October 17). Newer Chrome, Firefox show speed improvements | Business Tech – CNET News. *CNET News.* Retrieved, October 20, 2008, from http://news.cnet.com

[Tezuka06] Tezuka, T., Kurashima, T., Tanaka, K. (2006) *TOWARD TIGHTER INTEGRATION OF WEB SEARCH WITH A GEOGRAPHIC INFORMATION SYSTEM* International World Wide Web Conference, Proceedings of the 15th international conference, Edinburgh, Scotland, 277-286

[Traynor95] Traynor, C. and Williams, M. G. (1995). Why are geographic information systems hard to use? *Conference Companion on Human Factors in Computing Systems* (pp. 288-289). Denver, Colorado, United States, May 07 - 11, 1995. I. Katz, R. Mack, and L. Marks, Eds. CHI '95. ACM, New York, NY. DOI= http://doi.acm.org/10.1145/223355.223678

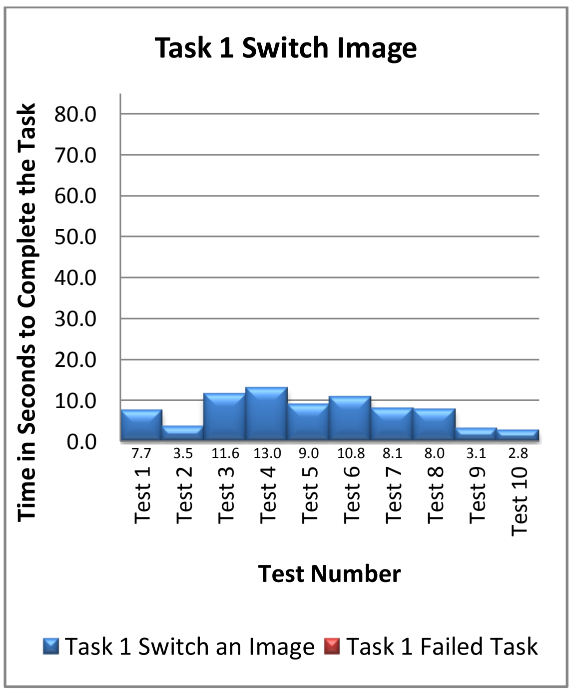
[Wei99] Wei, Z., Oh, Y., Lee, J., Kim, J., Park, D., Lee, Y., and Bae, H. (1999) *Efficient spatial data transmission in Web-based GIS* Proceedings of the 2nd international Workshop on Web information and Data Management (Kansas City, Missouri, United States, November 02 - 06, 1999). C. Shahabi, Ed. WIDM '99. ACM, New York, NY, 38-42.

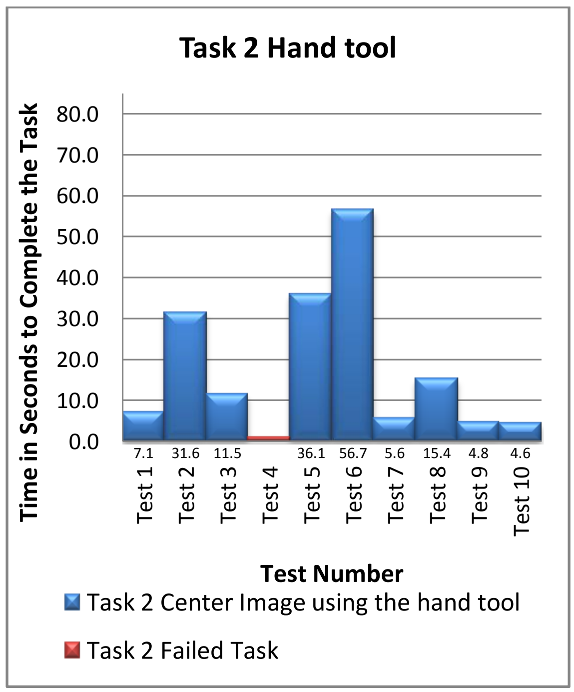
[Wiki08] JavaScript. (2008). In *Wikipedia, The Free Encyclopedia*. Retrieved, October 19, 2008, from http://en.wikipedia.org/

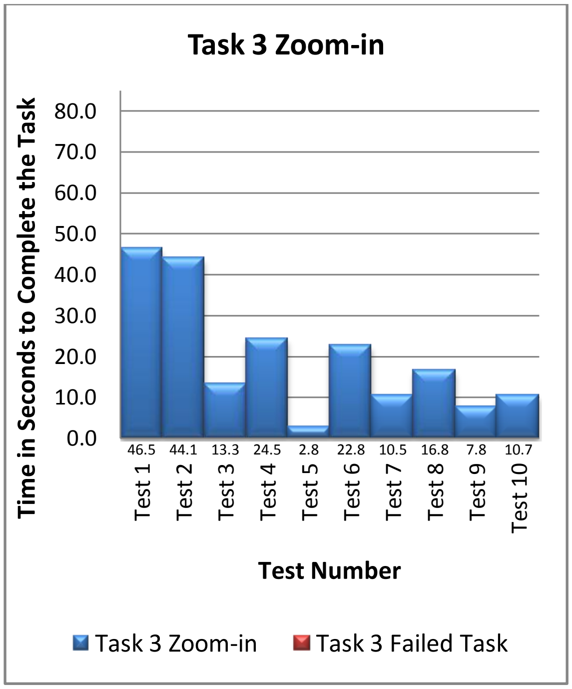
APPENDIX A:

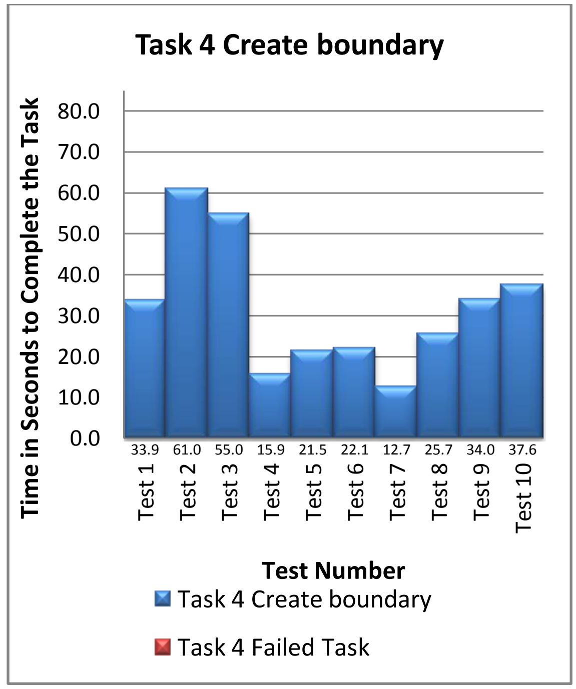
Data results Task-by-task

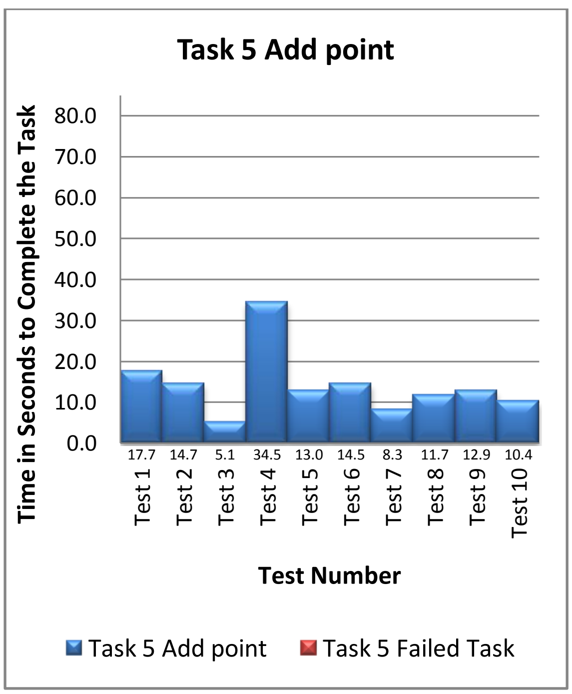
The following table graphics are the task-by-task results of the usability test performed with the participants while using the WALSAIP boundary tool.

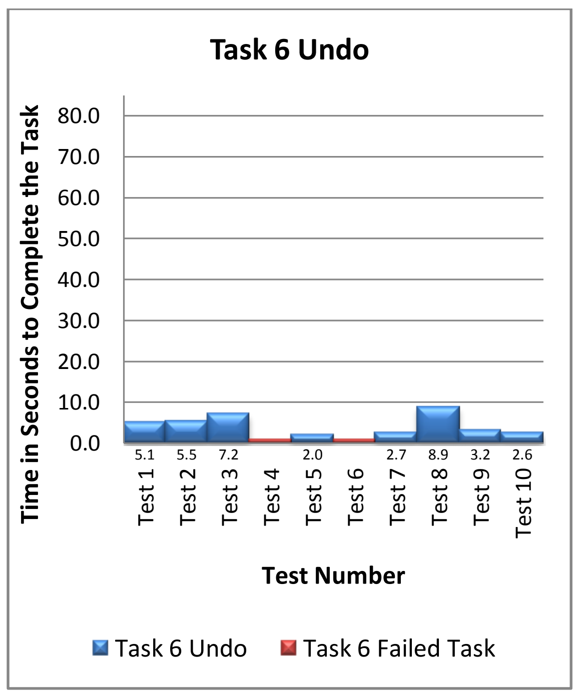


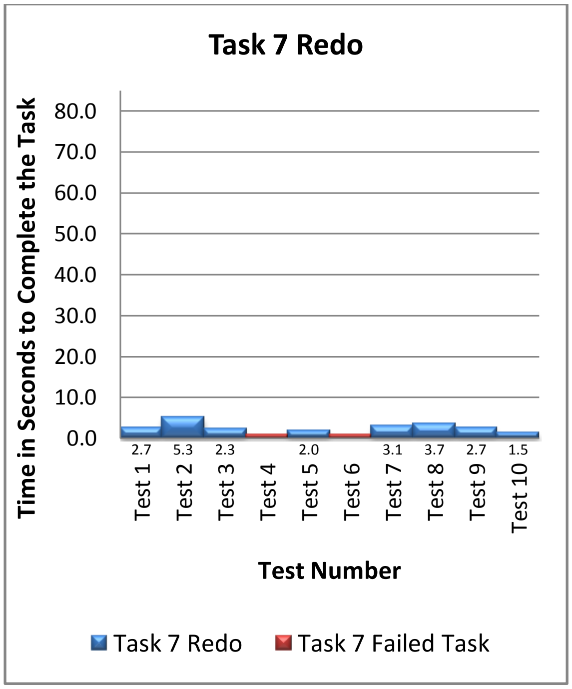


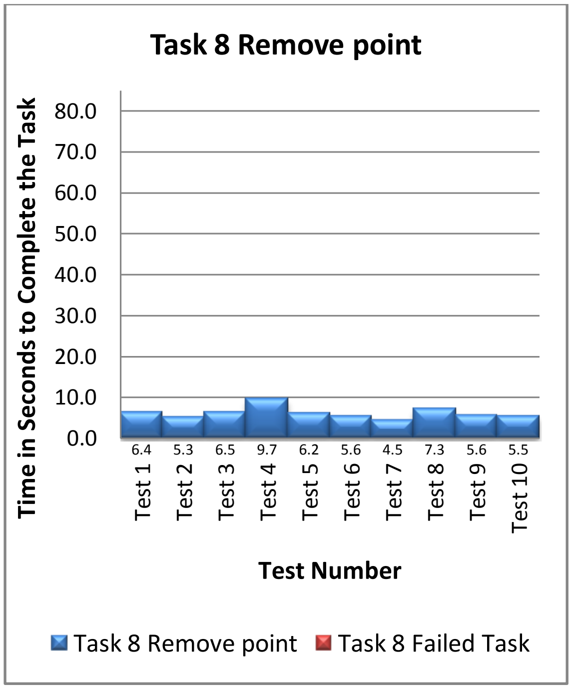

Task 3 Zoom-in
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
46.5 44.1 13.3 24.5 2.8 22.8 10.5 16.8 7.8 10.7
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 3 Zoom-in
Task 3 Failed Task

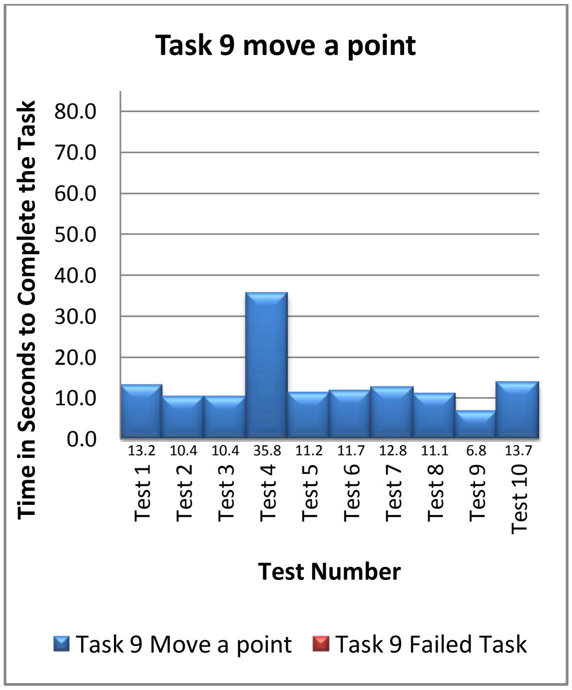

Task 4 Create boundary
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
33.9 61.0 55.0 15.9 21.5 22.1 12.7 25.7 34.0 37.6
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 4 Create boundary
Task 4 Failed Task

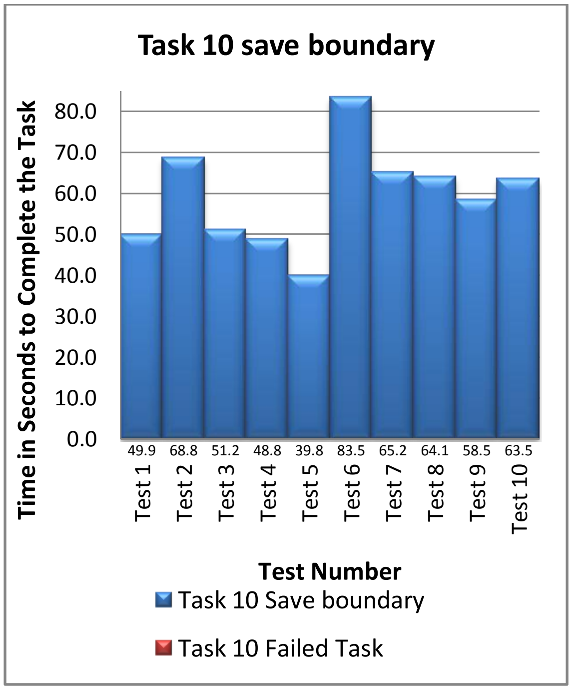

Task 5 Add point
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
17.7 14.7 5.1 34.5 13.0 14.5 8.3 11.7 12.9 10.4
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 5 Add point
Task 5 Failed Task

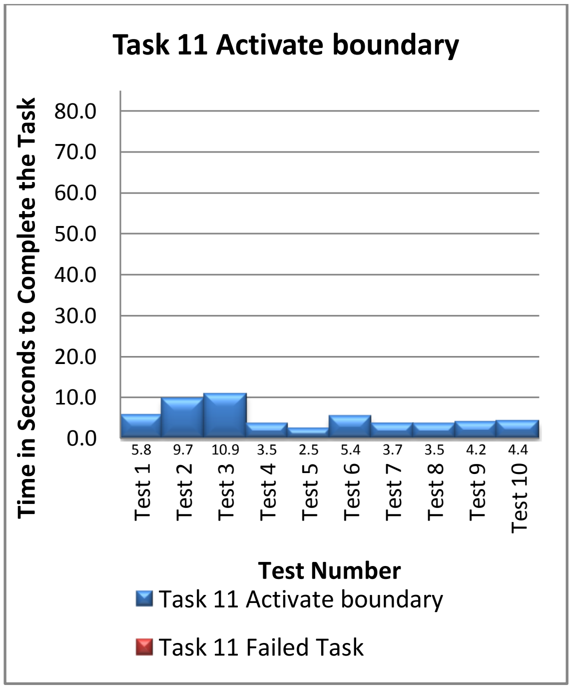

Task 6 Undo
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
5.1 5.5 7.2 2.0 2.7 8.9 3.2 2.6
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 6 Undo
Task 6 Failed Task

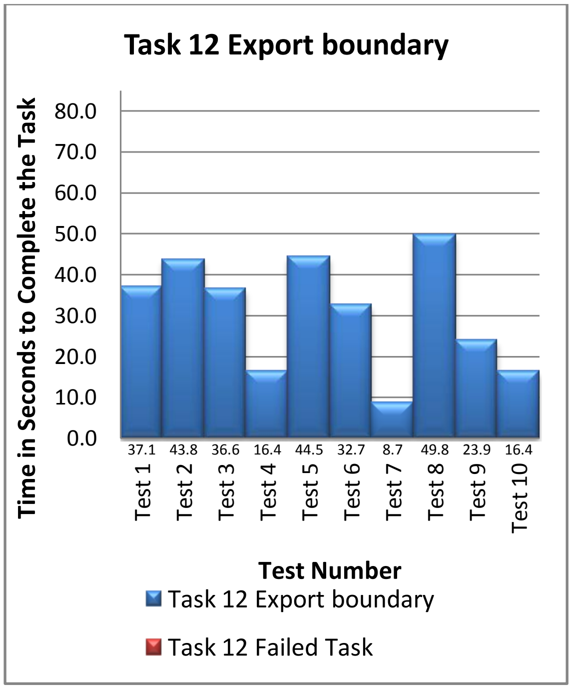

Task 7 Redo
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
2.7 5.3 2.3 2.0 3.1 3.7 2.7 1.5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 7 Redo
Task 7 Failed Task

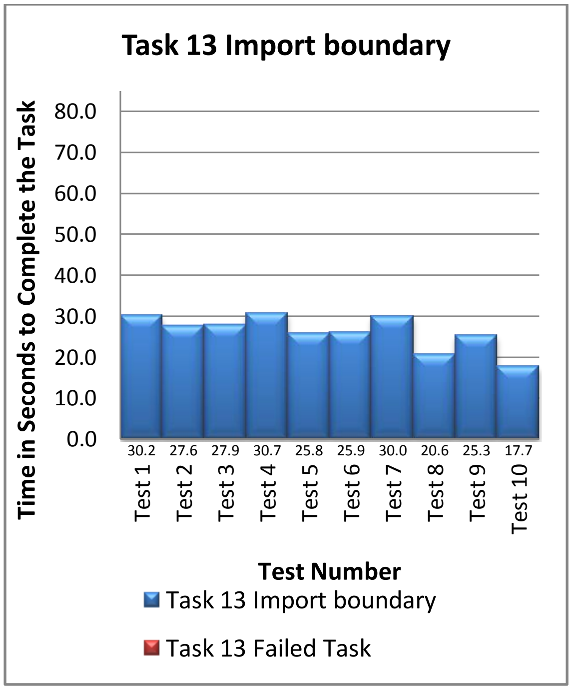

Task 8 Remove point
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
6.4 5.3 6.5 9.7 6.2 5.6 4.5 7.3 5.6 5.5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 8 Remove point
Task 8 Failed Task

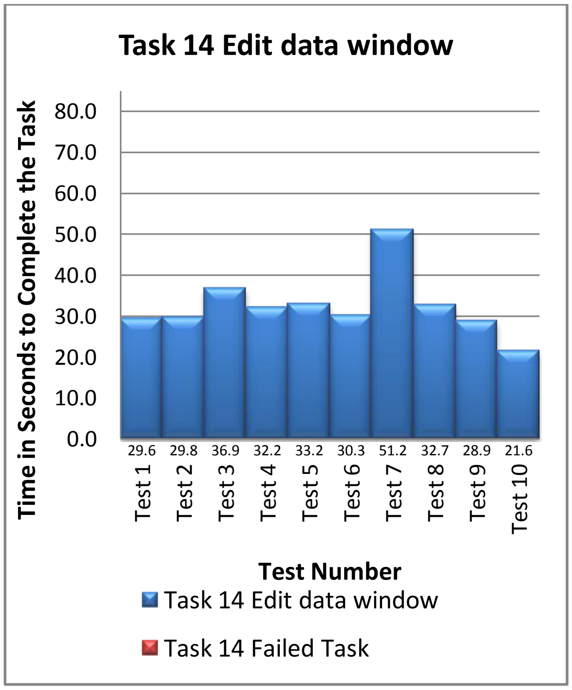

Task 9 move a point
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
13.2 10.4 10.4 35.8 11.2 11.7 12.8 11.1 6.8 13.7
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 9 Move a point
Task 9 Failed Task

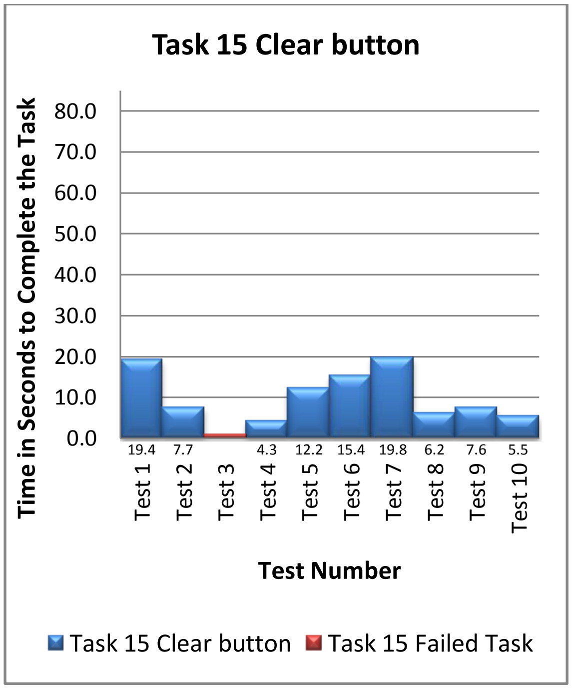

Task 10 save boundary
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
49.9 68.8 51.2 48.8 39.8 83.5 65.2 64.1 58.5 63.5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 10 Save boundary
Task 10 Failed Task

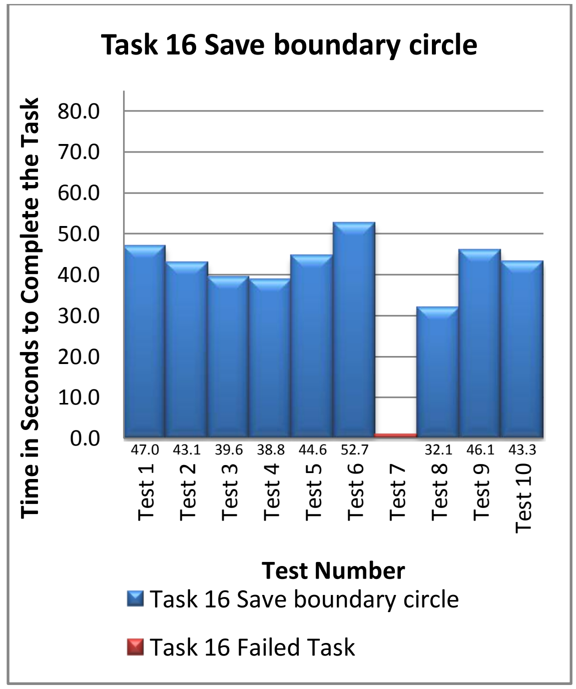

Task 11 Activate boundary
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
5.8 9.7 10.9 3.5 2.5 5.4 3.7 3.5 4.2 4.4
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 11 Activate boundary
Task 11 Failed Task

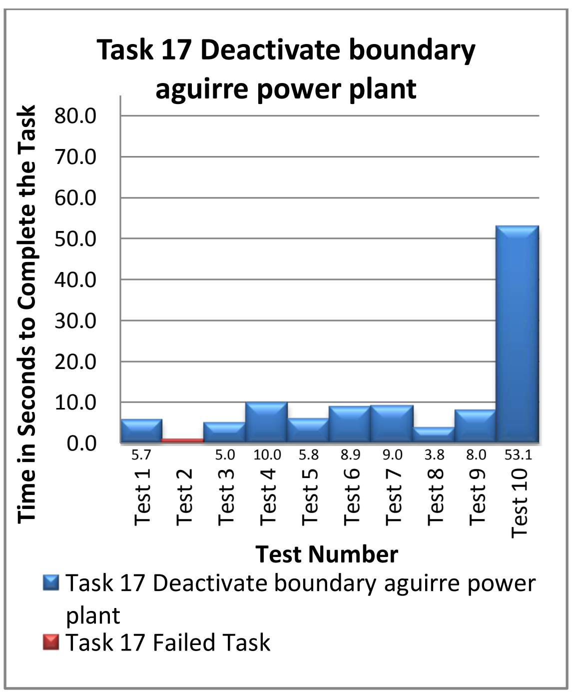

Task 12 Export boundary
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
37.1 43.8 36.6 16.4 44.5 32.7 8.7 49.8 23.9 16.4
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 12 Export boundary
Task 12 Failed Task

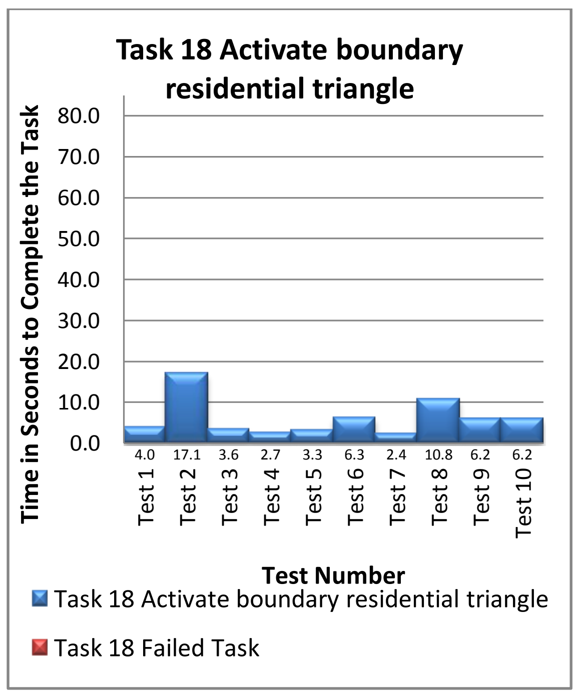

Task 13 Import boundary
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
30.2 27.6 27.9 30.7 25.8 25.9 30.0 20.6 25.3 17.7
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 13 Import boundary
Task 13 Failed Task

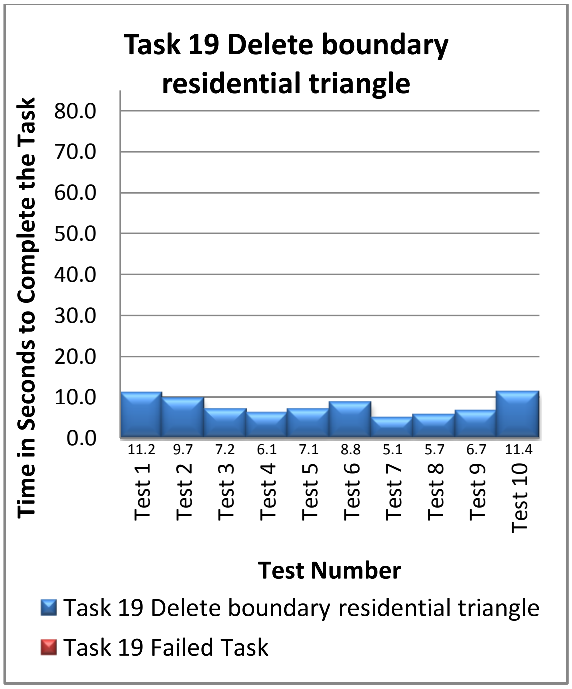

Task 14 Edit data window
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
29.6 29.8 36.9 32.2 33.2 30.3 51.2 32.7 28.9 21.6
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 14 Edit data window
Task 14 Failed Task

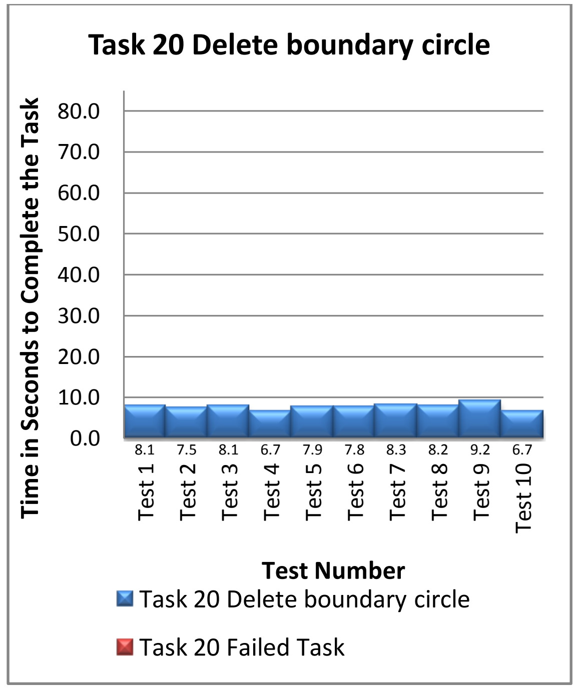

Task 15 Clear button
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
19.4 7.7 4.3 12.2 15.4 19.8 6.2 7.6 5.5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 15 Clear button
Task 15 Failed Task


Task 16 Save boundary circle
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
47.0 43.1 39.6 38.8 44.6 52.7 32.1 46.1 43.3
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 16 Save boundary circle
Task 16 Failed Task

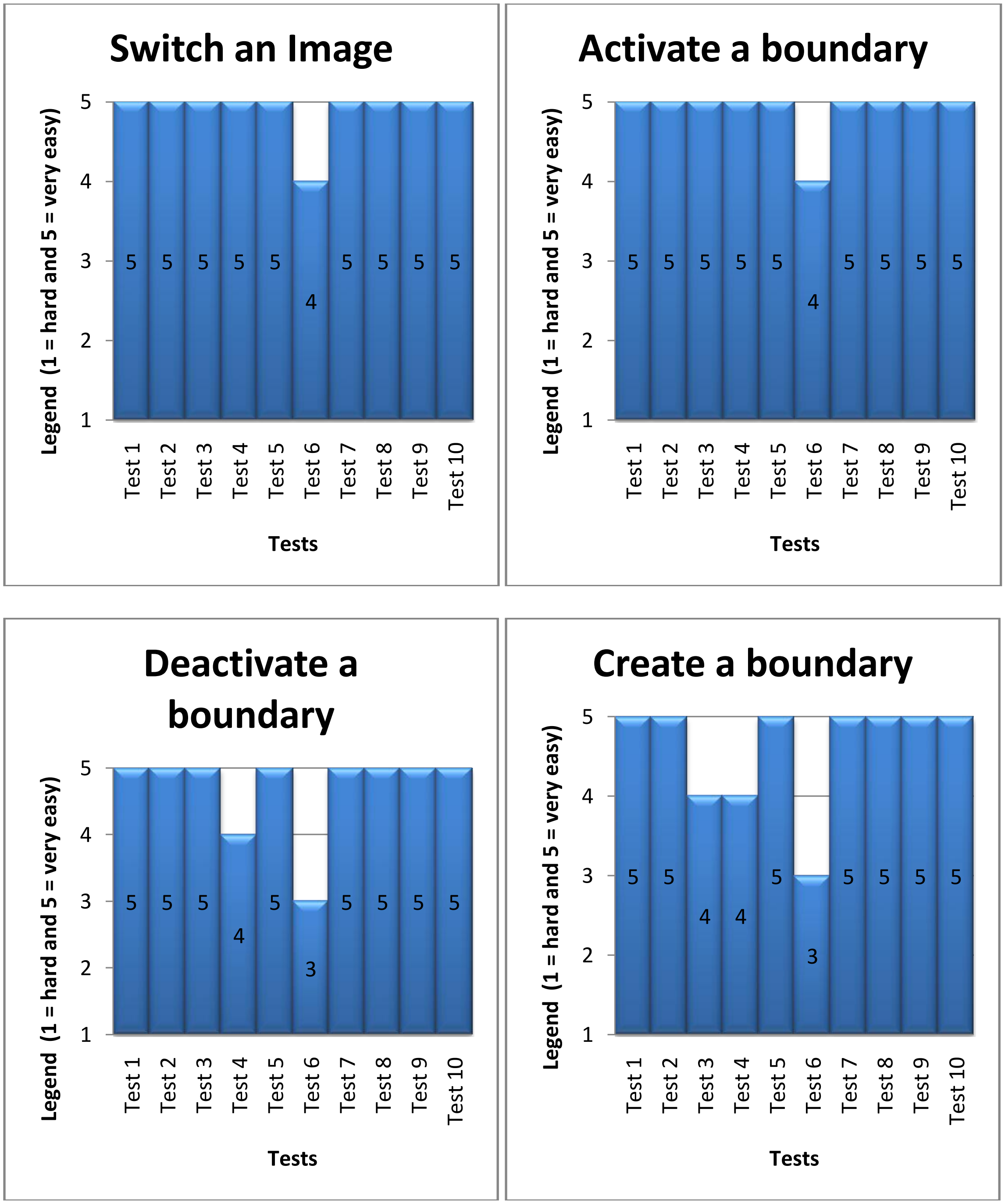

Task 17 Deactivate boundary aguirre power plant
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
5.7 5.0 10.0 5.8 8.9 9.0 3.8 8.0 53.1
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 17 Deactivate boundary aguirre power plant
Task 17 Failed Task

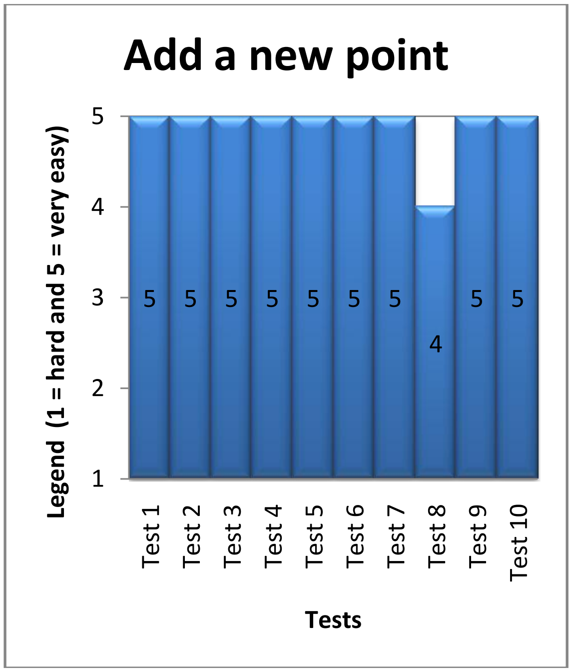

Task 18 Activate boundary residential triangle
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
4.0 17.1 3.6 2.7 3.3 6.3 2.4 10.8 6.2 6.2
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Test Number
Task 18 Activate boundary residential triangle
Task 18 Failed Task

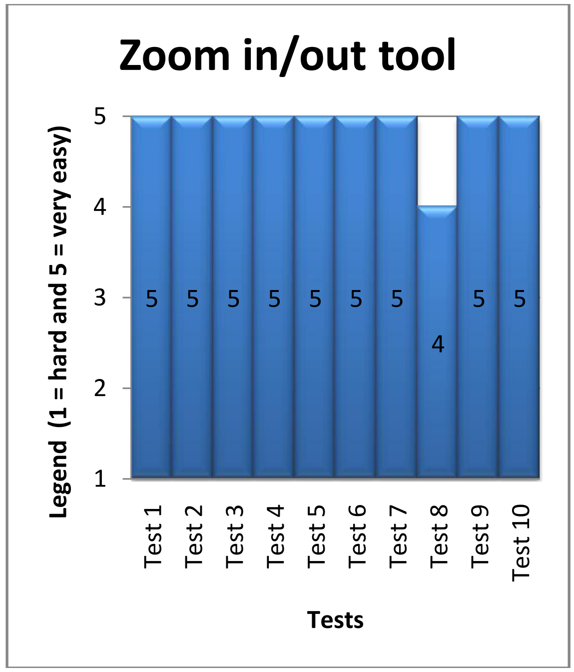

Task 19 Delete boundary residential triangle
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
11.2
9.7
7.2
6.1
7.1
8.8
5.1
5.7
6.7
11.4
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Test Number
Task 19 Delete boundary residential triangle
Task 19 Failed Task

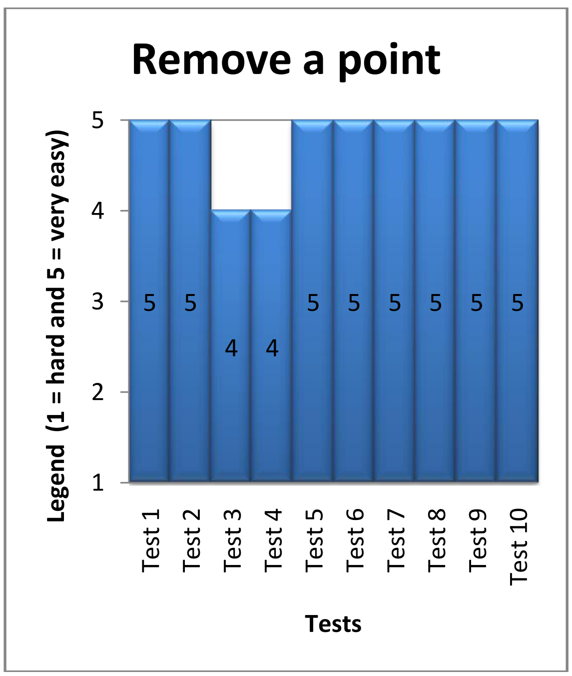

Task 20 Delete boundary circle
Time in Seconds to Complete the Task
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
8.1
7.5
8.1
6.7
7.9
7.8
8.3
8.2
9.2
6.7
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Test Number
Task 20 Delete boundary circle
Task 20 Failed Task

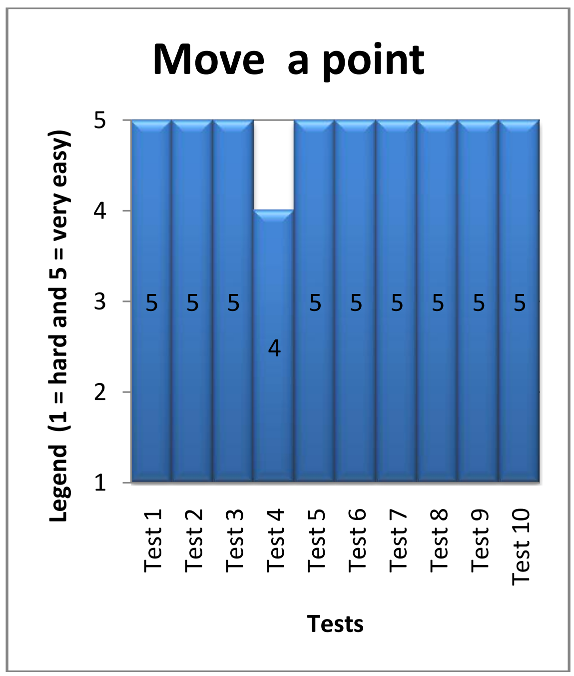
APPENDIX B:

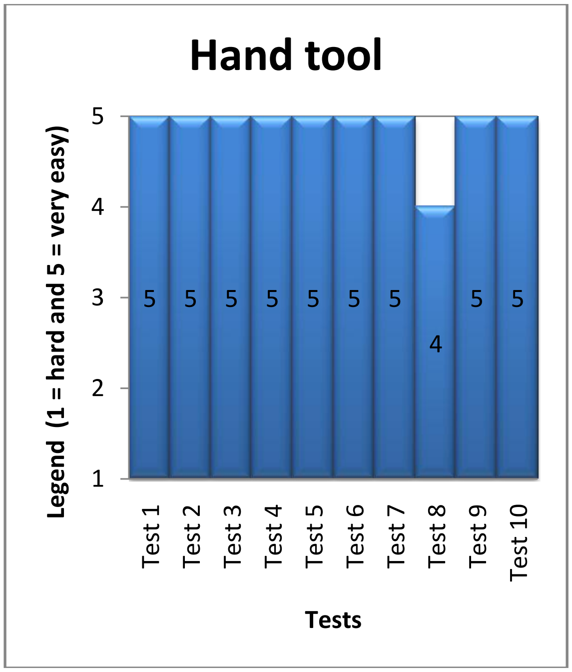
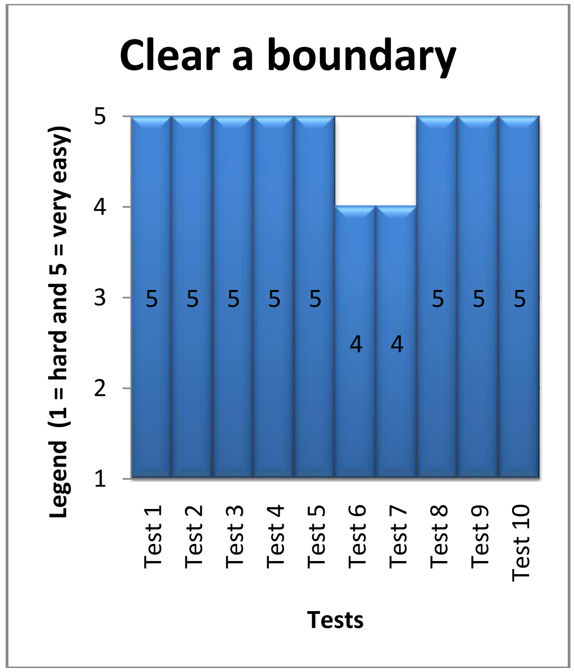
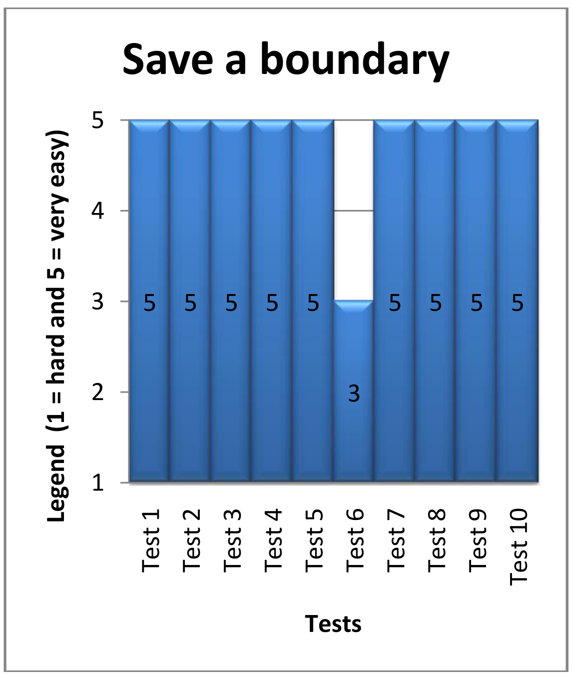
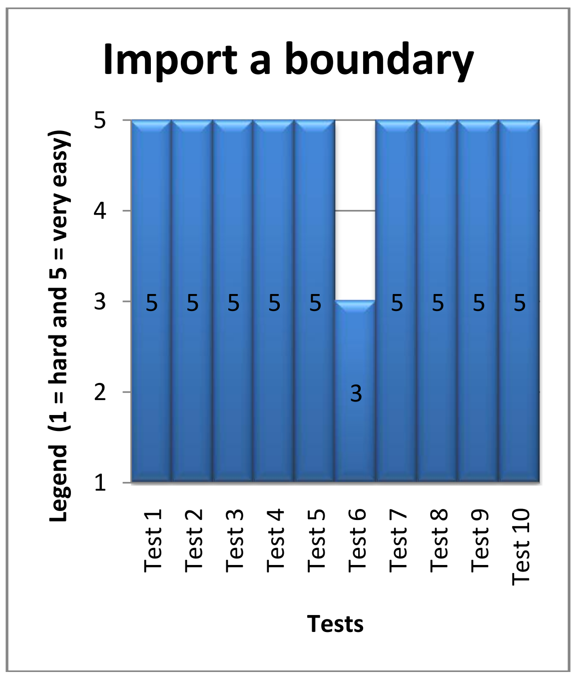

Switch an Image
Legend (1 = hard and 5 = very easy)
5 5 5 5 5 4 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests
Activate a boundary
Legend (1 = hard and 5 = very easy)
5 5 5 5 5 4 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests
Deactivate a boundary
Legend (1 = hard and 5 = very easy)
5 5 5 4 5 3 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests
Create a boundary
Legend (1 = hard and 5 = very easy)
5 5 4 4 5 3 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests

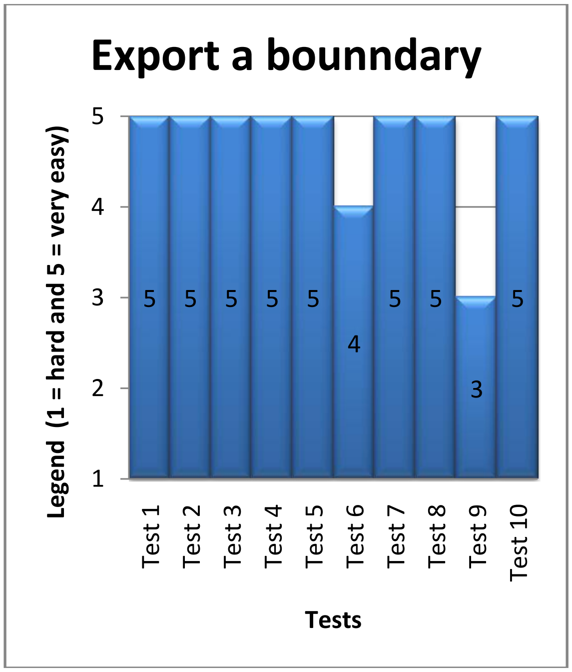

Add a new point
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 5 5 4 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests

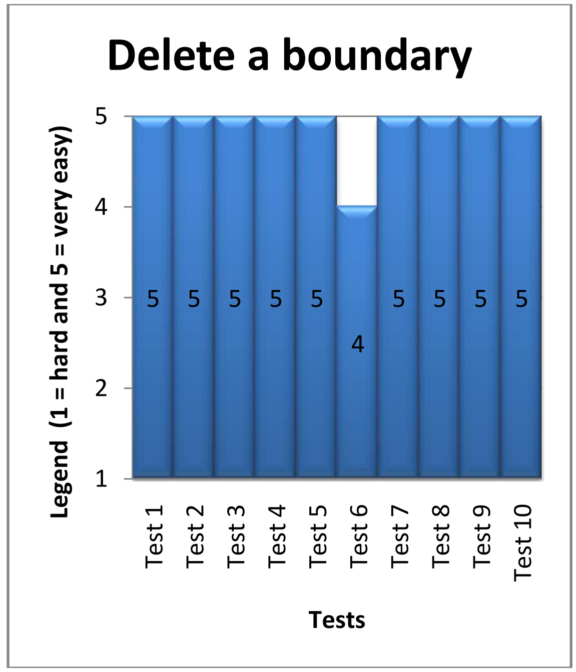

Zoom in/out tool
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 5 5 4 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests

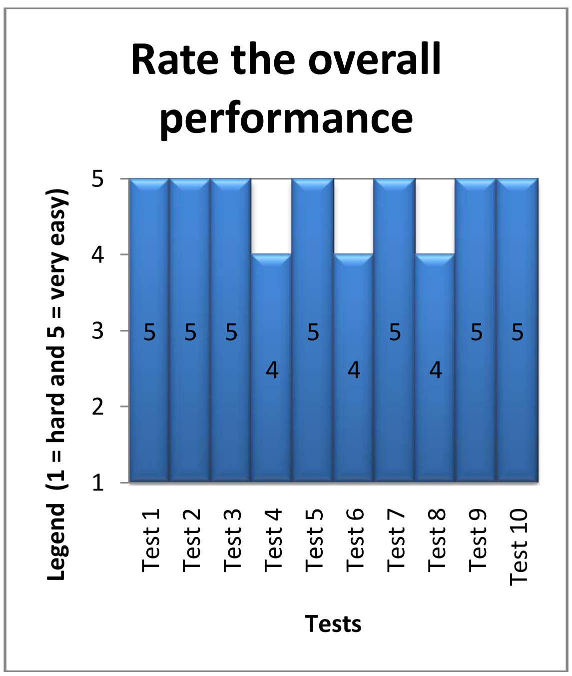

Remove a point
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 4 4 5 5 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests

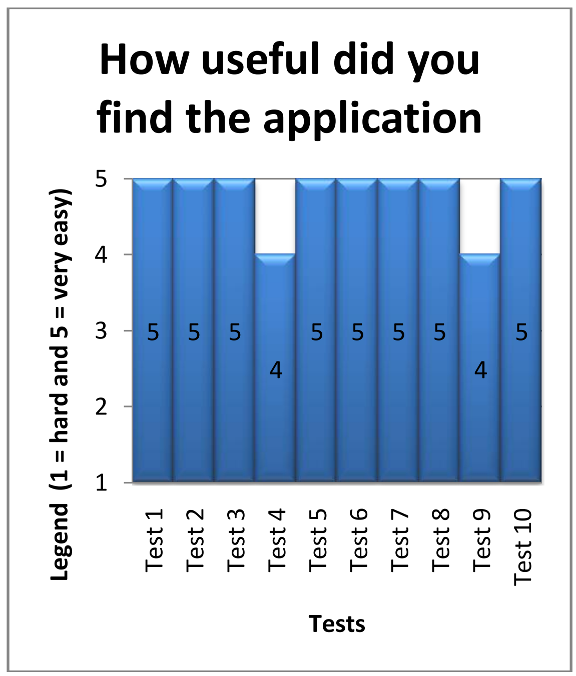

Move a point
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 4 5 5 5 5 5 5
Test 1 Test 2 Test 3 Test 4 Test 5 Test 6 Test 7 Test 8 Test 9 Test 10
Tests


Hand tool
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 5 5 4 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests

Clear a boundary
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 4 4 5 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests

Save a boundary
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 3 5 5 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests

Import a boundary
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 3 5 5 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests


Export a bounndary
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 4 5 5 3 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests


Delete a boundary
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 5 5 4 5 5 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests


Rate the overall performance
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 4 5 4 5 4 5 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests


How useful did you find the application
Legend (1 = hard and 5 = very easy)
5
4
3
2
1
5 5 5 4 5 5 5 5 4 5
Test 1
Test 2
Test 3
Test 4
Test 5
Test 6
Test 7
Test 8
Test 9
Test 10
Tests