GIS TOOL TO DEMONSTRATE FREEWAY EVOLUTION IN SAN DIEGO

A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Anand V. K. S. Bikkavilli
Summer 2012
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Anand V. K. S. Bikkavilli:

GIS Tool to Demonstrate Freeway Evolution in San Diego

__________________________________________
Carl Eckberg, Chair
Department of Computer Science

__________________________________________
Joseph Lewis
Department of Computer Science

__________________________________________
Dale Sevier
Center for Bio/Pharmaceutical and Biodevice Development

May 8, 2012
Approval Date
Copyright © 2012
by
Anand V. K. S. Bikkavilli
All Rights Reserved
DEDICATION

Dedicate to my friends and family for their love, support and for being there for me when it mattered the most.
ABSTRACT OF THE THESIS

GIS Tool to Demonstrate Freeway Evolution in San Diego
by
Anand V. K. S. Bikkavilli
Master of Science in Computer Science
San Diego State University, 2012

The modern freeway system is a result of meticulous planning and strategic foresight from 1947. The motivation behind this thesis is to design an informative tool on the evolution of the freeway system in San Diego. The implementation of this project is two-fold. An interactive GIS tool is developed in Java using Map Objects distributed by ESRI, and a website with detailed information about all the freeways in San Diego County is designed using Drupal. The essence of this tool is to organize the freeway data obtained from SANDAG (San Diego Association of Governments) and present it in a manner that is easier to comprehend. The user gets to select any freeway from the tool and it navigates to the corresponding web link. The tool also comprises of a timeline mechanism that shows freeway construction in chronological order.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>x</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2 SOFTWARE TECHNOLOGY</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Map Objects Java Objects (MOJO)</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Java Programming Language</td>
<td>4</td>
</tr>
<tr>
<td>2.3 Drupal</td>
<td>5</td>
</tr>
<tr>
<td>3 THE REQUIREMENTS</td>
<td>6</td>
</tr>
<tr>
<td>3.1 Data Requirements</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Platform Requirements</td>
<td>6</td>
</tr>
<tr>
<td>3.3 Functional Requirements</td>
<td>7</td>
</tr>
<tr>
<td>4 DEVELOPMENT</td>
<td>8</td>
</tr>
<tr>
<td>5 CUSTOM TOOLBARS</td>
<td>11</td>
</tr>
<tr>
<td>5.1 Custom Toolbar</td>
<td>11</td>
</tr>
<tr>
<td>5.1.1 Print Tool</td>
<td>11</td>
</tr>
<tr>
<td>5.1.2 Add Layer Tool</td>
<td>12</td>
</tr>
<tr>
<td>5.1.3 Pointer Tool</td>
<td>14</td>
</tr>
<tr>
<td>5.1.4 XY Tool</td>
<td>14</td>
</tr>
<tr>
<td>5.1.5 Hotlink Tool</td>
<td>14</td>
</tr>
<tr>
<td>5.1.6 Help Tool</td>
<td>16</td>
</tr>
<tr>
<td>5.2 Timeline Toolbar</td>
<td>18</td>
</tr>
<tr>
<td>6 MOJO TOOLBARS</td>
<td>19</td>
</tr>
<tr>
<td>6.1 ZoomPanToolBar</td>
<td>19</td>
</tr>
<tr>
<td>6.1.1 Previous Extent</td>
<td>20</td>
</tr>
<tr>
<td>6.1.2 Next Extent</td>
<td>20</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

PAGE

Figure 4.1. Transportation tool initial screen.................................................................10
Figure 4.2. Illustration of the timeline functionality.......................................................10
Figure 5.1. Custom toolbar. ..........................................................................................11
Figure 5.2. Print tool .......................................................................................................12
Figure 5.3. Print dialog. .................................................................................................12
Figure 5.4. Add layer tool. ............................................................................................12
Figure 5.5. Add layer dialog window. .............................................................................13
Figure 5.6. Map showing the San Diego roads system shape file. .................................13
Figure 5.7. Pointer tool. ...............................................................................................14
Figure 5.8. XY tool .......................................................................................................14
Figure 5.9. CSV file dialog. .........................................................................................15
Figure 5.10. Map with CSV file plotting the state routes in San Diego. .......................15
Figure 5.11. Hotlink tool ............................................................................................16
Figure 5.12. Dialog window opened up via hotlink tool. .............................................16
Figure 5.13. Web page showing detailed information about the selected freeway. .......17
Figure 5.14. Help tool. ...............................................................................................17
Figure 5.15. Web help for toolbars. ............................................................................17
Figure 5.16. Timeline toolbar. ....................................................................................18
Figure 5.17. Freeways built before the year 1990. .......................................................18
Figure 6.1. ZoomPanToolBar .......................................................................................19
Figure 6.2. Previous extent .........................................................................................20
Figure 6.3. Next extent...............................................................................................20
Figure 6.4. Zoom to active layer ..................................................................................20
Figure 6.5. Zoom to full extent ....................................................................................21
Figure 6.6. Zoom in. ....................................................................................................21
Figure 6.7. Zoom out. .................................................................................................21
Figure 6.8. Pan ............................................................................................................21
Figure 6.9. Pan one direction .................................................................22
Figure 6.10. Identify .............................................................................22
Figure 6.11. Window that shows the results of the identify tool. ..........22
Figure 6.12. SelectionToolBar .............................................................23
Figure 6.13. Find tool ...........................................................................23
Figure 6.14. Window opened up when a user clicks on find tool ..........24
Figure 6.15. Results returned from the find tool .................................24
Figure 6.16. Query builder .................................................................25
Figure 6.17. Query builder dialog window ............................................25
Figure 6.18. Result shown on the map upon query execution .............26
Figure 6.19. Select features icon ..........................................................26
Figure 6.20. various shapes of selecting features. .................................26
Figure 6.21. Selecting a part of the freeway map using the select features tool ..........................................................27
Figure 6.22. Clear selection tool ..........................................................27
Figure 6.23. Selected parts of the map are cleared ...............................28
Figure 6.24. Buffer tool .................................................................28
Figure 6.25. Buffer window .................................................................28
Figure 6.26. Map showing the result of the buffer tool ......................29
Figure 6.27. Attributes tool .................................................................29
Figure 6.28. Map showing the result returned from the attributes tool. 30
ACKNOWLEDGEMENTS

I would like to extend my sincere gratitude to my thesis advisor Dr. Carl Eckberg for taking the time to guide and support me through my thesis completion. I would like to thank Mr. John Hofmockel, SANDAG for his continuous support; Dr. Joseph Lewis for his time and support; Dr. Dale Sevier, for his invaluable ideas in improving my thesis. Furthermore, I thank my friends who stood by me and made this happen.
CHAPTER 1

INTRODUCTION

A geographic information system (GIS) excels in capturing, managing, analyzing and displaying various forms of geographically referenced information by integrating hardware, software, and data. GIS reveals relationships, patterns, and trends in the form of maps, globes, reports, and charts by allowing us to view, understand, question, interpret, and visualize data. A GIS displays data in a manner which is easily understood and shared so that it is helpful in answering questions and solving problems. GIS provides utility and communication companies with a common platform to access business data, manage assets, update network information, integrate work orders, find customer information, and prepare reports. GIS technology can be integrated into any enterprise information system framework.

The advent of GIS has revolutionized the way users can store, display, manipulate, analyze and link data locations on maps. The feature data related to maps is usually stored in an Environmental Systems Research Institute (ESRI) shape file format [1]. An application is represented in the form of layers where each layer corresponds to a specific type of data and the order of the layers is of great significance. An application that displays a layer of all major roads in a city might have the country and state layers underneath the city layer. Users can edit these layers individually or in combination. Each layer is displayed on a map. Attribute data is also available for each layer and the combination gives users a wealth of total information. ESRI provides a lot of layers for public usage and by the academic usage of GIS, students enhance their critical thinking, improve global awareness and develop a sense of community.

Map objects for Java (MOJO) is a java based Map API that provides developers with a highly customizable way of building map applications using the open-source Java programming language [2]. Map Objects was developed and distributed by ESRI, however the technical support is discontinued and no new development resources are being put in for further enhancements. Java however, continues to evolve, so the combination remains dynamic.
San Diego Association of Governments (SANDAG), is an association of local San Diego county and city governments. It is the metropolitan planning organization for the county with all the high profile policy makers. SANDAG has capital and fare setting powers for the county’s transit systems.

My thesis originated from the idea of creating an informative tool about the San Diego freeway system. The initial idea was to design a tool that elaborates cultural evolution in San Diego. It went through a series of valuable refinements by Prof Dr. Carl Eckberg, SDSU and SANDAG to be in the position it is now.

The information tool termed as the “Transportation Tool” is two-fold as it comprises of the tool and the website. The tool is designed using Map Objects for JAVA (MOJO) and it revolves around a single freeway shape file that has all the freeways in and around San Diego County. To make it San Diego-centered, a San Diego shape file is used as the base layer on top of which the freeway shape file is placed. Here, each freeway is marked using hotlinks which derive their attributes from the fields defined in a CSV file. The hotlinks navigate the user to a web link that provides information about that particular freeway.
CHAPTER 2

SOFTWARE TECHNOLOGY

This chapter elaborates the software technology used to design and develop the tool. Map Objects for Java which is licensed by Environmental Systems Research Institute (ESRI) and the Java SDK are used in the tool development. Drupal, a content management system (CMS) is used to develop the website with freeway information.

Java derives most of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Applications developed in Java are typically compiled to bytecode so that they run on any Java Virtual Machine (JVM) regardless of the computer architecture. Its “write once, run anywhere” characteristic encourages programmers to tend to Java in developing an application [3]. Java excels in providing an appealing look and feel as compared to its preceding programming technologies. Some of the special classes supporting this functionality include applets, servlets and swings. Swing is a graphical user interface library for the Java SE platform. It is possible to specify a different look and feel through the pluggable look and feel system of Swing. However, the User Interface for this is built using another special Toolkit as well called the Abstract Windowing Toolkit (AWT) [4]. It serves as an interface between a native windowing system and Java. It supports events and layout manager functionalities. And AWT provides a basic set of GUI widgets such as buttons, text boxes, and menus.

2.1 Map Objects Java Objects (MOJO)

Map Objects with Java could be viewed as:

- A Java application
- An example of component architecture
- A GIS application development tool

Map Objects is available as a collection of JAR files from which the user can extract objects that are needed on the application. MOJO takes a component architecture approach. A component is a piece of software that could easily be modified allowing a developer to
incorporate it in a required application. This approach encourages the user to focus on a single component to either modify its functionality or modify the state of the object. Some advantages of the component architecture are

- The user has precise control over the behavior of the application
- Since the application is made using the bottom-up approach, only those features necessary would be added

Visual Basic is another example of component architecture. Indeed, there is also Map objects for Visual Basic.

Map Objects supports the following functionalities:

- Capable of supporting multiple data sources
- Display real-time geographic data
- Easy to use user interface that allows exporting maps to a user-friendly viewing format such as jpeg
- Capable of supporting creation, addition and removal of shape files
- Query data necessary from the entire list of attributes supported by a shape file
- Enhancing user-accessibility by the use of labeling and other customary functionalities

2.2 Java Programming Language

Java programming language is based on the following principles [5]:

- The language should be easy to use and should be object-oriented
- It should be robust and provide proper security and authentication for the user applications
- The language should be portable so that once its compiled, it should run on any machine
- It should not compromise on performance
- It should be interpreter-oriented, support multi-threading, and should be dynamic

The Transportation tool takes almost all the key features described above into consideration. The key ones among the above mentioned principles include:

- Object-Oriented: Java is an object-centric language as the programming methodology revolves around the most basic unit called the object. An application in Java is divided into number of modules called classes which instantiate an object. The functionality of a class can be used only by instantiating an object. Objects could also manipulate the class methodology. Java is more strongly typed and purely OOP than C++.
Multithreaded and Dynamic: Multithreading is the ability to support and efficiently execute multiple threads. Multiple threads are allowed to exist within the context of a single process. These threads share the process' resources but are able to execute independently. The threaded programming model provides developers with a useful abstraction of concurrent execution. Some features in the Transportation tool use threading as it effectively reduces the wait time on a resource.

Portability and Reusability: Applications developed using Java are portable and once compiled, they could be executed on any Java virtual machine. A group of classes could be put together and a package is created which could be reused to achieve a particular functionality without having to implement the application from scratch.

Easy deployability: Executable jar files make Java very easy to deploy.

2.3 Drupal

Drupal is a content management system (CMS) and content management framework (CMF) developed using PHP [6]. It is a free and open source system distributed under the GNU General Public License. Drupal is an all powerful tool that encourages the users to reuse existing code chunks called modules. Any user is allowed to upload modules to the already existing module repository. Certain modules called “Core” modules co-exist along with user defined modules. Core modules enhance the functionality of the content management system by providing a number of features including:

- Access statistics and logging
- Advanced search
- Multi-level menu system
- Multi-site support
- User profiles
- Workflow tools and many more.
CHAPTER 3

THE REQUIREMENTS

The Transportation tool is developed for SANDAG and it would be available for public access. It would serve as an informative tool for historians and for anyone interested in getting to know about the freeway system and its evolution in San Diego.

The requirements are classified into the following types:

- Data Requirements
- Platform Requirements
- Functional Requirements

3.1 DATA REQUIREMENTS

Data requirements comprise of the essential data chunks necessary for the application as agreed upon prior to the start of implementation. Dr. Carl Eckberg suggested the following data requirements in accordance with the idea proposed by SANDAG.

- Freeway Name
- Year Completed
- Length in miles
- Web link
- Brief Description

3.2 PLATFORM REQUIREMENTS

This section lists and describes the platform requirements for the design and development of the Transportation tool.

- JAVA
- Map Objects Java Edition
- Drupal
- Windows OS

The Tool is designed and developed using the Java programming language and the predefined classes in Map Objects. Though a vast variety of IDEs that incorporate Java exist,
Eclipse is preferred, owing to its ease of use. Since Java is used as the core programming language in this implementation, the tool could be obtained as a JAR file that could run on any machine.

### 3.3 Functional Requirements

Functional requirements for the Transportation tool gradually evolved from the series of discussions with Mr. John Hofmockel, GIS Analyst from SANDAG. However, Dr. Carl Eckberg’s suggestions fine tuned the functionality of the tool making it more versatile in that it truly justifies its purpose of showing the evolution of transportation.

The functional requirements identified are:

- The tool by its entirety comes up with the latest freeway map on top of a San Diego County border map.
- The tool should provide tools to zoom in and zoom out and it should be capable of adding more layers e.g. countries, states, counties, rivers etc.
- The tool should provide information regarding each freeway that might include geographic pictures of the freeway, year built and other statistics such as the length of the freeway.
- The tool should also provide links to a website which should contain data related to each freeway in separate pages.
CHAPTER 4

DEVELOPMENT

The transportation tool is developed in Java using the Eclipse IDE [7]. Hence, it is platform independent. The Eclipse SDK is free and open source software. The Eclipse SDK comprises of the Eclipse Java development tools (JDT) and it provides an IDE with a built-in incremental Java compiler along with a full model of the Java source files. Eclipse capabilities extend to advanced refactoring techniques and code analysis. Using the Eclipse IDE, a powerful GUI can be built that supports simplified Swing Application Framework and the Abstract Window Toolkit. These features enhance the look and feel and ease-of-design for a Java application.

Map Objects Java Objects (MOJO) is available in the form of JAR files that could be added to the list of libraries in the application project. Each JAR file corresponds to an aggregation of many compiled Java classes. A freeway shape file representing all existing freeways in and around San Diego and a San Diego County boundary shape file are used in the tool development. The hotlink functionality is obtained by making the application read attribute data from a set of CSV (Comma Separated Values) files that are categorized based on a time period. The hotlinks are actually plotted on the freeway shape file depending on the latitude and longitude values read from the CSV file.

CSV files could be added into the tool using the XY button in the custom tool bar. When a user clicks on the XY button an add dialog opens up that points to the list of available CSV files that could be added. A CSV file, upon addition appears as a legend in the table of contents which is located to the left of map. Each member in the table of contents is called a legend and it represents a layer. The latitudes and longitudes that are read from the CSV file are converted into points and are stored in base points array.

The hotlinks populated from the CSV files would appear in various shapes and colors on the shape file and are based on the shape and color specified for each CSV file in the table of contents. All the necessary CSV files will be loaded when the software is executed. A checkbox is located adjacent to each legend and any legend be toggled at any time either by
Another important functionality provided by the tool is a timeline. A timeline represents a mechanism to show which chunks of a freeway are built at a given time period. The design of California Freeway System began as early as 1947 and some of the major freeways which are mostly interstates were open to public as early as 1970 [8]. As this data was not stored electronically until 1990, the earliest data obtained from the attribute table of the freeway shape file started from 1990. The timeline tool shows the freeways in the range of 1990 to 2011.

The QueryBuilder class which is a predefined class in MOJO is used to develop the timeline functionality [9]. A set of buttons are created and are grouped into a tool bar. Each button represents a particular time period to show the various freeways or parts of freeways built during the time period. When a user clicks a button, a query is executed by using data from the attribute table of freeway shape file. The result obtained from the query executed is highlighted on the freeway map.

Figure 4.1 shows the tool as it opens up upon user execution. Figure 4.2 shows the transportation tool when parts of the freeway map are highlighted upon the execution of timeline functionality.
Figure 4.1. Transportation tool initial screen.

Figure 4.2. Illustration of the timeline functionality.
CHAPTER 5

CUSTOM TOOLBARS

The functionality planned for the transportation tool is implemented by various toolbars, some customized and some predefined toolbars. This section explains all the components involved in the customized toolbars. The transportation tool has two customized toolbars:

- Custom toolbar
- Timeline toolbar

5.1 CUSTOM TOOLBAR

Custom toolbar comprises of six tools, each capable of performing a particular operation. The six tools are:

1. Print tool
2. Add layer tool
3. Pointer tool
4. XY tool
5. Hotlink tool
6. Help tool

Figure 5.1 shows the custom toolbar. Each tool is discussed in detail in the sub sections below.

![Custom Toolbar](image)

**Figure 5.1. Custom toolbar.**

5.1.1 Print Tool

The print tool delivers some basic printing functionality. High quality printing can be achieved by disabling double buffering. This enhances the clarity of the characters but might change the appearance of text. When a user clicks on the print tool as shown in Figure 5.2, a print dialog opens up, as shown in Figure 5.3.
5.1.2 Add Layer Tool

Add Layer tool provides additional layer functionality to the transportation tool. Layers apart from the existing layers can be added on to the map using the add layer tool. Any type of geographic data can be added as a layer. The layer data is represented in the form of a shape file. Figure 5.4 shows the add layer tool.

When a user clicks on the add layer tool, a dialog window opens up prompting the user to select a shape file from the list of existing shape files as shown in the Figure 5.5.

Figure 5.6 shows the map of San Diego roads system, when the roadways shape file is added through the add layer dialog window. For example, trolley lines and railroad lines could be addable, in addition to example in Figure 5.6.
Figure 5.5. Add layer dialog window.

Figure 5.6. Map showing the San Diego roads system shape file.
5.1.3 Pointer Tool

The Pointer tool serves as the arrow tool. Its main functionality is to get the cursor back to normal after clicking on a particular tool as the cursor would exhibit the previous tool’s functionality until brought back to normal. As an example, when a user clicks on the zoom in button from the Zoom Pan Toolbar and clicks on a particular area on the map, it zooms in and this functionality is attached to the cursor until it is removed. At this point, the pointer tool is used to bring back the cursor functionality. Figure 5.7 shows the pointer tool from the custom toolbar.

![Figure 5.7. Pointer tool.](image)

5.1.4 XY Tool

The XY tool reads the values from the fields of a CSV (Comma Separate Value) file and adds the data as a layer of points on to the map. The layer is added as a legend to the table of contents. Typically, a CSV file contains any attribute data separated by commas. In the transportation tool, a CSV file contains latitude and longitude of the point to be placed as a hotlink, year in which the freeway is built, length of the freeway and other information which is freeway specific. Figure 5.8 shows the icon used to represent the XY tool in the custom toolbar. When a user clicks on the XY tool icon, a file dialog opens up prompting the user to select a CSV file from the list of available CSV files. The file dialog is shown in Figure 5.9. Figure 5.10 shows the map when a state routes CSV file is added by the user.

![Figure 5.8. XY tool.](image)

5.1.5 Hotlink Tool

The hotlink tool provides information about a particular point on the map. A lightning bolt icon is used to represent the hotlink tool. Even the cursor value changes to bolt icon when a user clicks on the hotlink tool. The hotlink tool works with the active layer of the map. If a user clicks the hotlink tool first and clicks on any highlighted point in the map, a
Figure 5.9. CSV file dialog.

Figure 5.10. Map with CSV file plotting the state routes in San Diego.
dialog window opens up showing brief information about a particular freeway. Figure 5.11 shows the hotlink tool icon.

![Figure 5.11. Hotlink tool.](image)

The dialog window contains brief information about the name of the freeway, length of the freeway, year built and a brief description. It also contains a web link button which when clicked, directs to a web page that shows more information about that particular freeway. Figure 5.12 shows the dialog window opened up, when a user clicks on the hotlink tool.

![Figure 5.12. Dialog window opened up via hotlink tool.](image)

Figure 5.13 shows the snapshot of a webpage accessed via the hotlink’s dialog window.

5.1.6 Help Tool

The help tool is a special tool which when clicked by a user, directs to a web link that displays brief information about each tool in the toolbars. Figure 5.14 shows the icon used to represent the help tool. Figure 5.15 shows the web link pointed to by the help tool.
Figure 5.13. Web page showing detailed information about the selected freeway.

Figure 5.14. Help tool.

Figure 5.15. Web help for toolbars.
5.2 Timeline Toolbar

The timeline toolbar supports an innovative functionality in which the freeways or chunks of freeways built over various time periods were highlighted in the freeway map. The timeline toolbar comprises of five buttons, each representing a particular period of time. The timeline ranges from 1990 to 2011 as the year data starts at 1990 for the freeway shape file.

The area highlighted by clicking on a button is erased when another button representing a different time period is clicked. Figure 5.16 shows the timeline toolbar.

![Figure 5.16. Timeline toolbar.](image)

To observe timeline functionality, the freeway layer has to be selected first from the table of contents. Once the freeway layer is selected, clicking on any button would highlight the freeway parts in San Diego built during that time period. This data is being read from the attribute table of the freeway shape file. Figure 5.17 shows a screen shot of the map with highlighted freeway parts built before the year 1990 when the user clicks on button 1990.

![Figure 5.17. Freeways built before the year 1990.](image)
CHAPTER 6

MOJO TOOLBARS

Map Objects Java Edition provides a set of toolbars, each with a collection of predefined tools that perform individual functions [10]. The toolbars used for the transportation tool are

- ZoomPanToolBar
- SelectionToolBar

This section explains each toolbar and all the tools contained in them.

6.1 ZOOMPANTOOLBAR

The ZoomPanToolBar’s functionality justifies its name. As the name goes, ZoomPanToolBar consists of tools that provide basic zoom in, zoom out, pan and other essential functionalities. This toolbar allows map resizing which is achieved by saving the initial context of the map and switching back to it from a recent context that could either be a zoomed in or a zoomed out state. The tools supported by the ZoomPanToolBar are

- Previous Extent Tool
- Next Extent Tool
- Zoom to Active Layer Tool
- Zoom to Full Extent Tool
- Zoom In Tool
- Zoom out Tool
- Pan Tool
- Pan One Direction Tool
- Identify Tool

Figure 6.1 shows the ZoomPanToolBar with all its supported tools.

Figure 6.1. ZoomPanToolBar.
6.1.1 Previous Extent

The state of the map is saved every time the user modifies it. Modification here means changing the way the map is viewed, like zooming in or zooming out. When a user clicks on the previous extent tool, the most recent state of the map shows up. Figure 6.2 shows the icon used in the ZoomPanToolBar to represent the previous extent tool.

![Figure 6.2. Previous extent.](image)

6.1.2 Next Extent

The next extent tool comes into play when a user already has a future context saved and is currently viewing an old context. When the next extent tool is clicked, the future context which is saved gets loaded onto the map. The previous extent and the next extent tools could be viewed as forward and backward buttons respectively (see Figure 6.3).

![Figure 6.3. Next extent.](image)

6.1.3 Zoom to Active Layer

Zoom to active layer tool is functionally available only when a selection is made in the table of contents i.e., when a legend is selected by clicking on the legend itself. Active layer refers to the layer corresponding to the legend selected from the table of contents. When a user clicks on the Zoom to active layer tool, the active or the selected layer is shown on the map. Figure 6.4 shows the icon used to represent zoom to active layer tool.

![Figure 6.4. Zoom to active layer.](image)

6.1.4 Zoom to Full Extent

A tool like Zoom to full extent is always handy to have as it helps the user to roll back and shows the map in its entirety. When a user repeatedly zooms in or zooms out on a particular part of the map and is lost, a click on the zoom to full extent would show the full
map how it was intended to be viewed. Figure 6.5 shows the icon used to represent zoom to full extent tool.

![Figure 6.5. Zoom to full extent.](image)

### 6.1.5 Zoom In & Zoom Out

The zoom in and zoom out tools facilitate zooming which is a very simple functionality but an important one. When a user clicks on the zoom in button, it zooms into the map and the cursor value changes which would zoom in whenever clicked. The zoom out tool zooms out of the map and also changes the cursor value. The pointer tool brings back the cursor’s pointing functionality when clicked. Figures 6.6 and 6.7 show the icons used to represent the zoom in and the zoom out tools respectively.

![Figure 6.6. Zoom in.](image)

![Figure 6.7. Zoom out.](image)

### 6.1.6 Pan

The pan tool is a good option to have when a user zooms into a map and would like to drag it to any direction required. When the pan tool is clicked, the user can drag the map which is out of screen, in any direction comfortable to view. Figure 6.8 shows the icon used to represent the pan tool.

![Figure 6.8. Pan.](image)

### 6.1.7 Pan One Direction

The pan one direction tool isolates the functionality supported by the pan tool and restricts it to a particular direction. When a user clicks on the pan one direction tool, it
prompts the user to select a particular direction to pan. Figure 6.9 shows the icon used and the options that appear when it is clicked.

![Pan one direction](image)

**Figure 6.9. Pan one direction.**

### 6.1.8 Identify

The identify tool serves as an informative tool, for a user to identify the area selected on the map. First, the user should click on the identify tool and then click on any point on the map. A window pops up showing the details about that point on the map. When the tool is clicked, the cursor’s value changes to that of the identify icon and windows keep popping up whenever clicked. The pointer tool when clicked brings back the cursor’s natural functionality. Figures 6.10 and 6.11 show the icon used to represent the identify tool and the window that pops up when a point is clicked.

![Identify](image)

**Figure 6.10. Identify.**

![Window that shows the results of the identify tool](image)

**Figure 6.11. Window that shows the results of the identify tool.**
6.2 SELECTION TOOL BAR

The SelectionToolBar comprises of a set of selection tools that let the user select any part of the map in order to perform a particular function [11]. Figure 6.12 shows the selection tool bar and the tools supported by it.

The selection tool bar comprises of the following tools:

- Find Tool
- Query Builder Tool
- Select Features Tool
- Clear Selection Tool
- Buffer Tool
- Attributes Icon Tool

Each of the tools listed above is explained in detail in this section.

6.2.1 Find

Find tool helps the user in finding a particular data on the map when the data available is huge. Initially, a layer is selected from the table of contents and the find tool is clicked. This opens up the find dialog window which prompts the user to enter the text that needs to be searched. When the find button from the dialog is clicked, the tool searches the entire attribute table of the layer selected and displays the matches if any. Figure 6.13 shows the icon used to represent the find tool. Figure 6.14 shows the window dialog opened when clicked on the find tool. Figure 6.15 shows the results returned from the find tool for the text entered.

6.2.2 Query Builder

Query Builder is a powerful tool which helps users to query for the type of data required from the existing layer data. It also displays the results returned from executing the
Figure 6.14. Window opened up when a user clicks on find tool.

Figure 6.15. Results returned from the find tool.
query onto the map. A layer must be selected from the table of contents before clicking the query builder tool. Figure 6.16 shows the icon used to represent the query builder tool. Figure 6.17 shows the dialog window opened up when the query builder tool is clicked. Figure 6.18 shows the results returned after query execution and shows the map highlighted with them.

Figure 6.16. Query builder.

![Query builder icon](image)

Figure 6.17. Query builder dialog window.

**6.2.3 Select Features**

Select features tool lets the user select a part of the map. When a user clicks on the select features icon, a drop-down menu shows up with four possible shapes of selection. The user is allowed to select one shape and use it select that part of the map that is intended to be selected. Upon successful selection, the area selected would be highlighted in yellow. Multiple areas could be selected by using the shift key. The icon to represent this tool is shown in Figure 6.19. Figure 6.20 shows the drop down that opens up after clicking this tool. Figure 6.21 shows the parts of the map highlighted in yellow after successful selection.
6.2.4 Clear Selection

Clear selection tool is used to clear the selected area from the map. This tool comes into play effectively after select features tool is used. Figure 6.22 shows the icon used to represent this tool. Figure 6.23 shows that the selection made by the select features tool from the Figure 6.21 is cleared.
6.2.5 Buffer

The buffer tool is enabled only when a layer is selected from the table of contents and some parts of interest on the map are selected by using the select features tool. The icon used to represent this tool is shown in Figure 6.24. The icon is initially grayed out. A buffer window shown in Figure 6.25 opens up when a user clicks on the buffer tool. The functionality of this tool is to buffer the selected area depending on the number and type of units selected from the buffer window. Figure 6.26 shows the result of the buffer tool on the map.

6.2.6 Attributes Icon

Attributes icon tool is grayed out by default. To make the tool visible, a layer has to be selected from the table of contents and an area has to be selected from that layer by using
Figure 6.23. Selected parts of the map are cleared.

Figure 6.24. Buffer tool.

Figure 6.25. Buffer window.
Figure 6.26. Map showing the result of the buffer tool.

select features tool. When a user clicks on this tool, an attribute window opens up showing the attributes for the various features selected. Figure 6.27 shows the icon used to represent the attributes tool. Figure 6.28 shows the attributes window with the attributes for the features selection from the map.

Figure 6.27. Attributes tool.
Figure 6.28. Map showing the result returned from the attributes tool.
CHAPTER 7

CONCLUSION

This thesis aimed at designing and developing an interactive GIS tool for SANDAG to demonstrate the evolution of the freeway system in San Diego. The motivation behind developing this tool is to provide an easy to access mechanism for historians and anyone interested in getting to know about how the modern freeway system evolved. The software requirements for this tool were gathered from Prof Dr. Carl Eckberg and SANDAG. This tool is developed using MOJO (distributed by ESRI) and Java [12]. It revolves around two shape files that represent the modern freeway system in San Diego and San Diego county border.

The freeway map covers all the freeways and their exits constructed before 2012. Each freeway has a hotlink associated with it on the map which provides two levels of information about the freeway. The first level of information is obtained by clicking the hotlink window which opens up a dialog window that shows brief information about the freeway. This information includes the name of the freeway, year it was built, length of freeway and a brief description about it. The second level of information is the detailed description when a user clicks on the web link button from the dialog box. The web link button directs the user to a freeway website exclusively created as a part of this application. This website contains information for each freeway and could be reached either from the hotlinks in the tool or could be accessed explicitly without the tool.

The other important feature provided by the tool is the timeline feature. This features highlights certain freeways and parts of freeways that were constructed during a given time period. It comprises of five time periods ranging from 1990 to 2011. This information shows the freeway construction in chronological order. All in all, this tool aims to be the one stop place for San Diego lovers and historians to get information about the modern freeway system.

As is often the case with GIS, the greatest obstacle was obtaining detailed and accurate data. Both California and local government sources needed to be searched.
CHAPTER 8

FUTURE ENHANCEMENTS

The transportation tool is developed using Map Objects for Java [13] and some of the Graphical User Interface is developed using Java Swing which makes adding features and tweaks to the application at a later stage practically possible. The code written is pure Java which supports reusability and could be extended with ease. Some of the areas that could be concentrated upon and considered for enhancements are:

- Implement the timeline feature using a slider bar that would show the freeways for a particular time period.
- Tweak the timeline feature so that when a user wishes to view the freeways that existed in a particular year, all the other future constructions should be removed from the map.
- Adding additional CSV files to the application that classify information based on length of the freeways, traffic and other information.
- Adding a functionality which when enabled, would let the users select a freeway either from a drop-down or a similar mechanism and highlights the entire freeway on the map.
- Adding self test quizzes which target students.
- Enhancing the time evolution feature by developing a tool that flashes a given freeway when a slider reaches the time of its completion.
- Fine tuning all of the above for freeways completed in stages.
BIBLIOGRAPHY


