GIS TOOL ON FUEL RESOURCES OF THE AMERICAN CONTINENT

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
Nachiket Anil Tanksale
Summer 2014
SAN DIEGO STATE UNIVERSITY

The Undersigned Faculty Committee Approves the

Thesis of Nachiket Anil Tanksale:

GIS Tool on Fuel Resources of the American Continent

Carl Eckberg, Chair
Department of Computer Science

Roger Whitney
Department of Computer Science

Usha Sinha
Department of Physics

May 6, 2014
Approval Date
DEDICATION

This thesis is dedicated to my family for their unconditional love and support. It wouldn’t have been possible if not their consent for me pursuing a Masters Degree. A special thanks to Rohit and Kanchan for their continuous encouragement and motivation. Also, thanks to Kshitij and Charu for their continuous support and care. I take this opportunity to thank all my dear friends for their wonderful support during this entire tenure and for the times to come.
ABSTRACT OF THE THESIS

GIS Tool on Fuel Resources of the American Continent
by
Nachiket Anil Tanksale
Master of Science in Computer Science
San Diego State University, 2014

The motivation behind the development of this software is to employ current technology in the interest of an innovative education approach. Learning about the fuel resources, their distribution, production and usage is much more interesting when seen through images, fact sheets and videos.

A GIS tool for learning about fuel resources has been developed keeping in mind a student’s and a common user’s perspective. Though the idea of fuel production and consumption is very important for anybody, its knowledge in people is sadly lacking. All efforts have been made to design this software in such a way that it engages the users by using videos, images, short summary and quick links to get more information from html pages as well. Each of the geographic location types was established as a layer on a world map. According to the requirement, the layer created was a point layer or polygon layer or a line layer. This software can be customized according to the user’s requirement and is easy to install and create using GIS technologies like MapObjects – JAVA edition and JAVA along with the Eclipse IDE.

Attention is paid to each significant fuel, for each of Canada, Mexico and the United States.

A large focus is on display of information using maps, since maps and geography are well known weaknesses in K-12 students.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT .................................................................................................................. v</td>
</tr>
<tr>
<td>LIST OF TABLES ........................................................................................................ viii</td>
</tr>
<tr>
<td>LIST OF FIGURES ...................................................................................................... ix</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS ................................................................................................. xi</td>
</tr>
</tbody>
</table>

## CHAPTER

1. **INTRODUCTION** ............................................................................................................ 1
2. **TECHNOLOGY** ............................................................................................................ 2
   2.1 JAVA .......................................................................................................................... 2
   2.2 MapObjects Java Edition ......................................................................................... 3
3. **THE REQUIREMENTS** ................................................................................................. 5
   3.1 Data Requirements .................................................................................................. 5
   3.2 Platform Requirements ......................................................................................... 5
   3.3 Functional Requirements ..................................................................................... 6
4. **PROTOTYPING** ........................................................................................................... 7
5. **SOFTWARE DESIGN** .................................................................................................. 9
   5.1 High Level Architecture Diagram ......................................................................... 9
   5.2 Classes and Their Diagram ................................................................................. 9
6. **ECLIPSE AND MAPOBJECTS CONFIGURATION** .............................................. 11
7. **MAPOBJECTS TOOLBAR** ......................................................................................... 16
   7.1 com.esri.mo2.ui.tb.ZoompanToolbar .................................................................. 16
   7.2 com.esri.mo2.ui.tb.SelectionToolbar ................................................................. 17
   7.3 JAVA/Custom Toolbar ......................................................................................... 17
8. **CUSTOM TOOLBAR** ................................................................................................. 18
   8.1 Print ....................................................................................................................... 18
   8.2 Add Layer ............................................................................................................. 19
   8.3 Pointer ................................................................................................................... 20
   8.4 Hotlink .................................................................................................................. 20
8.5 XY Tool ..............................................................................................................21
8.6 Distance Tool........................................................................................................21
8.7 Help Tool ..............................................................................................................21

9 ADDITIONAL TOOLS ............................................................................................25
9.1 Layer Control ......................................................................................................25
9.2 Help Tab ..............................................................................................................26
9.3 Co-Ordinates Display ........................................................................................26
9.4 Legend Editor ......................................................................................................28

10 SCREENSHOTS ...................................................................................................30

11 CONCLUSION AND OBSTACLES ....................................................................41

12 FUTURE ENHANCEMENTS ...............................................................................42

REFERENCES .............................................................................................................43
LIST OF TABLES

PAGE

Table 5.1. Importing Classes and Classes Used for Development .................................10
Table 8.1. Custom Toolbar ..........................................................................................18
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Prototype model for SDLC</td>
<td>8</td>
</tr>
<tr>
<td>5.1</td>
<td>Highlevel architecture design</td>
<td>10</td>
</tr>
<tr>
<td>6.1</td>
<td>Starting Eclipse</td>
<td>12</td>
</tr>
<tr>
<td>6.2</td>
<td>Creating a new Java project</td>
<td>12</td>
</tr>
<tr>
<td>6.3</td>
<td>Creating a new library</td>
<td>13</td>
</tr>
<tr>
<td>6.4</td>
<td>List of existing libraries</td>
<td>13</td>
</tr>
<tr>
<td>6.5</td>
<td>Adding external JARs for MOJO support</td>
<td>14</td>
</tr>
<tr>
<td>6.6</td>
<td>Configure Java build path</td>
<td>14</td>
</tr>
<tr>
<td>6.7</td>
<td>Specifying the Javadoc location</td>
<td>15</td>
</tr>
<tr>
<td>6.8</td>
<td>Specifying the Java code style</td>
<td>15</td>
</tr>
<tr>
<td>7.1</td>
<td>MapObjects toolbar</td>
<td>16</td>
</tr>
<tr>
<td>8.1</td>
<td>Custom toolbar</td>
<td>18</td>
</tr>
<tr>
<td>8.2</td>
<td>Print screen</td>
<td>19</td>
</tr>
<tr>
<td>8.3</td>
<td>Add layer dialogue window</td>
<td>20</td>
</tr>
<tr>
<td>8.4</td>
<td>GIS tool with hotlink window</td>
<td>21</td>
</tr>
<tr>
<td>8.5</td>
<td>Hotlink tool pick window</td>
<td>22</td>
</tr>
<tr>
<td>8.6</td>
<td>GIS tool with XY tool window</td>
<td>23</td>
</tr>
<tr>
<td>8.7</td>
<td>GIS tool with distance tool window</td>
<td>23</td>
</tr>
<tr>
<td>8.8</td>
<td>GIS tool with help tool window</td>
<td>24</td>
</tr>
<tr>
<td>9.1</td>
<td>Layer control before using the menu</td>
<td>25</td>
</tr>
<tr>
<td>9.2</td>
<td>Layer control after promoting a states layer</td>
<td>26</td>
</tr>
<tr>
<td>9.3</td>
<td>Help tool tab</td>
<td>27</td>
</tr>
<tr>
<td>9.4</td>
<td>Help tool usage</td>
<td>27</td>
</tr>
<tr>
<td>9.5</td>
<td>Coordinates display</td>
<td>28</td>
</tr>
<tr>
<td>9.6</td>
<td>Legend editor for selected layer</td>
<td>29</td>
</tr>
<tr>
<td>9.7</td>
<td>GIS tool with labeling on selected layer</td>
<td>29</td>
</tr>
<tr>
<td>10.1</td>
<td>Loading a shape file wizard</td>
<td>31</td>
</tr>
</tbody>
</table>
Figure 10.2. Shape file browser wizard. ................................................................. 31
Figure 10.3. After finding a .shp file. ................................................................. 32
Figure 10.4. World map shape file loaded screen.............................................. 32
Figure 10.5. csv file wizard.................................................................................. 33
Figure 10.6. csv file loaded................................................................................. 33
Figure 10.7. Hotlink tool pick on Kayenta Mine, Arizona. ............................... 34
Figure 10.8. Hotlink click on ‘Watch my Video’ leading to YouTube............... 34
Figure 10.9. Sample welcome page of the information webpage.................... 35
Figure 10.10. Welcome page content................................................................. 35
Figure 10.11. Webpage regarding current status of the three countries. ......... 36
Figure 10.12. Sample webpage regarding coal extraction methods.............. 36
Figure 10.13. Sample webpage regarding controversial history of coal mining. 37
Figure 10.14. Sample webpage showing forecast stats..................................... 37
Figure 10.15. Query builder tool......................................................................... 38
Figure 10.16. Usage of identify tool on map of India........................................ 38
Figure 10.17. Usage of “find” tool........................................................................ 39
Figure 10.18. Usage of “buffer” tool ................................................................. 39
Figure 10.19. After using buffer tool................................................................. 40
ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Carl Eckberg from the bottom of my heart for giving me an opportunity to work on this thesis and for his immense guidance and support during the project period.

I am very grateful to Dr. Roger Whitney, Professor, Computer Science Department of San Diego State University for accepting my thesis and for his invaluable guidance and support in encouraging me to get this work going.

I am also very thankful to Dr. Usha Sinha, Professor, Department of Physics of San Diego State University for being a committee member and her co-operation throughout.

I am thankful to my family and friends for their love and motivation without which it would have been a much difficult task to accomplish.
CHAPTER 1

INTRODUCTION

Human’s curiosity to know about the environment and adapt accordingly has lead to advancement in technologies. Existing technology has helped him understand the surroundings that he lives in and helped make a better place to live. He has made good use of technology to explore the surroundings and the natural resources for the betterment of his living. The advent of electronic media has created a new wave. People are making use of different web-interactive tools and multimedia for learning and understanding. E-commerce and E-learning are common terms.

The idea behind developing this thesis was to come up with an interactive tool to study the existing resources of fuel and energy in USA, Canada and Mexico where one can add layers, remove layers and get information about particular points of interest through hotlinks. The hotlinks which can provide information through images and web links which provide concise information about the energy resources at that point. The information available on the internet is very robust and not easy for a common man to analyze its content. This tool has a simple web interface which provides information about current fuel resources, their stats and future estimate.

This GIS tool has been developed using MapObjects and Java technologies. Eclipse IDE is used as the development platform.

The thesis report is organized into 12 chapters. The first two chapters give an introduction about the thesis and the technologies that were used in the development of the tool. Chapters 3 and 4 describe the requirements for this thesis and the prototype followed for it’s successful completion. Chapters 5 and 6 explain about the software design of the tool, and the process of setting up Eclipse IDE for development of Java based MapObjects classes. Chapters 7, 8, and 9 explain about the various toolbars used and the added functionalities in detail. Chapter 10 gives an insight to the entire project through screenshots. Chapter 11 and 12 discusses about the obstacles faced during the development cycle and few possible future enhancements.
CHAPTER 2

TECHNOLOGY

This chapter mainly focuses on the technologies that were used to give a final shape to this thesis. Java and MapObjects Java Edition are two softwares that have been used to develop this software tool as suggested by Dr. Carl Eckberg. Java is a platform independent programming language and it gels well with MapObjects Java Edition. The two technologies are discussed in detail in this chapter.

2.1 JAVA

Java was developed by Sun Microsystems. It is platform independent and this ability makes it one of the most popular and widely used languages. Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers “write once, run anywhere” (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to bytecode (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture [1].

Various advantages of JAVA are [1, 2]:

- **Java is simple** – Java uses automatic memory allocation and garbage collection that makes it so easy to write, compile, debug and learn in java.

- **Java is object-oriented** – Java allows creating modular programs and reusable code. It is object-oriented programming language as it is centered on creating objects, manipulating objects, and making objects work together.

- **Java is platform-independent** – One of the most significant advantages of Java is its ability to move easily from one computer system to another.

- **Java is distributed** – Distributed computing involves several computers on a network working together. Java is designed to make distributed computing easy with the networking capability that is inherently integrated into it.

- **Java is interpreted** – An interpreter is needed in order to run Java programs. The programs are compiled into Java Virtual Machine code called bytecode. This
bytecode is machine independent and can run on any machine that has java interpreter.

- **Java is secure** – Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind.

- **Java is robust** – Robust refers to reliability. Java puts a lot of emphasis on early checking for possible errors, as Java compilers are able to detect many problems that would first show up during execution time in other languages.

- **Java is multithreaded** – Multithreaded is the capability for a program to perform several tasks simultaneously within a program. In Java, multithreaded programming has been smoothly integrated. Multithreading is a necessity in visual and network programming.

- **Java is easily deployed** – Java applications are easy to transport and deploy by using executable java archives, i.e. jar files.

### 2.2 MapObjects Java Edition

MapObjects Java Edition can be used to build custom applications that incorporate GIS mapping capabilities or to extend the capabilities of existing applications. MOJO (MapObjects Java Objects) is the core Java API (Application Programming Interface) used for building GIS application. It is a developer’s toolkit that includes a set of jar files containing pure java components that can be used to develop standalone GIS application. MapObjects Java Edition is a powerful collection of client and server side components used to build custom, cross platform mapping and GIS applications. It has more precise control over the behavior of the application; a user can make his application from the bottom up, adding only those features that he requires; a GIS class at San Diego State University covers MapObjects in depth and has a published guide named Notes On MapObjects Java Edition by Dr. Carl Eckberg, which was one of the main criteria for choosing MapObjects [3].

Important features as cited from the MapObjects Java Edition brochure are as follows [3, 4]:

- Applications which are built using MapObjects Java Edition can support activities such as labeling map features, thematic mapping, panning and zooming through multiple map layers, specifying projections, querying spatial and attribute data, performing geometric operations, measuring distances, displaying real time geographic data, and creating layouts, creating shape files, projecting on particular locations.
• One can easily distribute MapObjects Java Edition applications over an internet or intranet through browser hosted applets or simplify web delivery of your application with the use of Sun Java web start technology, the industry standard launching mechanism for distributing Java applications over the web; one can use custom created web pages to display the information related to the topics.

• By using the Swing components included with MapObjects Java Edition, you can quickly build applications that include functional toolbars, dynamic symbol control, query dialogs, overview and insert maps, and intelligent legends, pop up windows, alert boxes, java panes which have the ability to display multiple format of data that make your custom applications easy to use and even easier to develop and understand.

• By using applications created with MapObjects Java Edition, end users can combine local data with Internet and Intranet data to create their own customized maps. They can also easily integrate these maps with different locations corresponding to the map projected using comma separated value files which is having latitude and longitudes of the locations. It also supports different data formats including shape files, ArcSDE layers, ArcIMS image and feature services, image formats such as BMP, TIFF, PNG, JPG, GIF. Generally these image formats come into the picture when we try to create different toolbars, having their own images to recognize; you can also use MapObjects Java Edition to access files from your own custom data sources for easy integration.

• The server side map components in MapObjects Java Edition allow developers to build map services, Java Server Pages (JSPs) and servlets, or custom Enterprise JavaBeans (EJB) Web based mapping applications. Several extensive sample applications based on JSPs and EJBs are provided to demonstrate how to build robust Web applications with the server side map objects.

• The different MapObject packages contain many modified classes, which are used to create various tools; this makes a developer’s job very easy.

• MapObjects describes itself as “Component Architecture“, somewhat analogous to Visual Basic. But the application must be written in Java, utilizing MapObjects components.
CHAPTER 3

THE REQUIREMENTS

The GIS tool for learning about fuel resources is an interactive tool developed for students/users to enable a more efficient and interesting knowledge system about fuel resources in Canada, Mexico and the United States. The thesis/project has been developed under the guidance of Dr. Carl Eckberg from the San Diego State University Computer Science Department and he was the major contributor of the requirements for this thesis. The requirements gathered have been classified as follows:

- Data requirements
- Platform requirements
- Functional requirements

3.1 DATA REQUIREMENTS

Data requirements refer to the data that should be collected to be represented in the tool. The display of data is covered in the Functional Requirements. Dr. Carl Eckberg helped in identifying important information about fuel resources in different countries and points of interests to be displayed for the respective countries of interest.

All the data required for this thesis has been collected under the supervision of Dr. Carl Eckberg. The data includes all the shape files in the first place. These files have information about geographic coordinates of the places to be shown, the year of operation, the fuel production of that place, images related to it, methods of extraction of that particular fuel and data on any controversial situations.

3.2 PLATFORM REQUIREMENTS

Platform requirements refer to the development language to be used and the GIS solution that should be implemented. Both of these information was collected from Dr. Carl Eckberg. The entire set up was framed on Windows 7 platform which has various advantages. These technologies offer a wide range of mapping and GIS functionality.
MapObjects Java Edition helps build applications that perform a variety of geography-based display, query and data retrieval activities at the client, presentation, Web, and server tiers.

Thus they allow representing the geographical data on maps which make the tool more interactive and informative [3, 4].

The following has been used:
- JAVA as Programming Language, specifically Java 1.7
- MapObjects Java Edition as the GIS API

### 3.3 Functional Requirements

Functional requirements of this GIS tool for learning about fuel resources were gathered from Dr. Carl Eckberg. It refers to the functional details of the tool and the GUI (Graphical User Interface) requirements.

The functional requirements are as follows:
- The software should come up with a default map of the World.
- The software shall provide tools to zoom in and zoom out.
- The software shall provide pan options.
- It should be capable of adding/removing layers e.g., fuel resources types, their resource areas etc.
- The places should be identifiable with the help of an identification tool.
- It should provide a built-in print option.
- All the layers should be present when the tool is launched and the user can choose to make a particular layer visible/invisible.
- GIS Tool should be able to allow the users to navigate to different web pages.
- The software should be capable of labeling.
- Provision to click on a particular feature to get the detailed information about it.
CHAPTER 4

PROTOTYPING

Prototyping is the most important part in developing any application. It is the blueprint of the work from the starting to the end. It describes the flow and phases of the work. It is essential because it gives a clear idea to the developer as well as the end user of how the application is going to be what exactly it is supposed to do. Also, if there are any changes at any point of time, it can be well adjusted in the prototype model. For example, the all the information regarding ‘Kayenta coal mine’ can be said to be the prototype. Any changes in steps and data display were first implemented on the prototype. Basically prototyping follows the software development life cycle which includes:

- **Requirement Analysis** – This is the first step where in we decide all the requirements for the entire thesis/project. It includes data requirement, functional requirement and platform requirement.

- **Design** – This step includes analyzing the design of the application in broader terms which is like creating an initial draft about the features and operations in detail. It also includes the screen layouts.

- **Development/Implementation** – It includes the actual coding part where in all the requirements are implemented and appropriate code is written to get the desired result.

- **Verification/Review** – This step includes getting the application developed so far approved by the end user, in this case by Dr. Carl Eckberg to assure that all the requirements have been successfully met. In this step after getting the reviews essential changes are made.

- **Maintenance** – This step pertains to the help/maintenance required once the application is deployed but in this case it is not of much importance as it is out of the scope of this thesis.

- **End User Testing** – This topic may not closely fit any particular University course. The government has a Department of Energy but SDSU does not. Environment issues loom large, so Engineering and Ecology are involved, but the Department most closely involved is Geology. Thus the application is best reviewed by Geologists, and was also on a variety of college and high school students.

Figure 4.1 [5] of the software development lifecycle which shows the important steps included in it. Though in this thesis the requirements were changed from time to time, this
development model was not completely followed. This was because for any new feature addition the whole SDLC was to be followed again.

There were modifications to the initial prototype according to the feedback received by the thesis supervisor. All the required features and functionality were implemented and reviewed within the given time frame.
CHAPTER 5

SOFTWARE DESIGN

This chapter mainly focuses on the high level software design/architecture and the class diagrams. It comes into picture once all the data and functional requirements have been met. The entire application has been designed to provide easy installation and deployment keeping the end user in mind.

5.1 HIGH LEVEL ARCHITECTURE DIAGRAM

Figure 5.1 shows a high level architecture diagram for GIS tool for learning about fuel resources in three countries. It is further subdivided into 3 parts which are discussed here:

- **User Interface** – It defined the end user interaction with the application developed. It shows the screen which is launched when the application starts and the way user can use this application to learn about the fuel resources. It displays all the layers, giving the user the ability to add/remove layers, color them, zoom in, zoom out, pan, zoom to active layer, help, select various features, open attribute tables and much more.

- **MapObjects and Java Classes** – It contains all the functionality and actions of the various toolbars like custom toolbar, selection toolbar, button, menus and other items on the main interface are defined and created in different Java class.

- **Shape/Resource/Text Files** – All the ‘.csv’ files and text files having the location (latitudes/longitudes) comes under this section. These ‘.csv’ files are displayed on the user interface using MOJO and JAVA classes.

5.2 CLASSES AND THEIR DIAGRAM

Class diagrams are very useful and helpful in organizing the entire code and it is structure. It gives a clear understanding of how different classes interact with each other. It also accounts for the interdependencies of various classes used in this project for the application development. See Table 5.1.
Figure 5.1. Highlevel architecture design.

Table 5.1. Importing Classes and Classes Used for Development

<table>
<thead>
<tr>
<th>JAVA COMPONENTS</th>
<th>JAVA CLASSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>import com.esri.mo2.ui.bean</td>
<td>TocAdapter, TocEvent, Map, Layer, Toc, Legend, Tool</td>
</tr>
<tr>
<td>import javax.swing</td>
<td>MenuBar, JMenuitem, AbstractTableModel, TableColumn, JButtons, Jpanel, JsplitPane, JToolBar, ImageIcon</td>
</tr>
<tr>
<td>import com.esri.mo2.ui.tb</td>
<td>ProjectToolBar, ZoomPanToolBar, SelectionToolBar</td>
</tr>
<tr>
<td>import com.esri.mo2.map.dpy</td>
<td>BaseFeatureLayer, Layerset, FeatureLayer</td>
</tr>
<tr>
<td>com.esri.mo2.file.shp</td>
<td>ShapefileFolder, ShapefileWriter</td>
</tr>
<tr>
<td>com.esri.mo2.map</td>
<td>DatasetLayer, TrueTypeMarkerSymbol, Layerset, featureLayer, BaseFeatureLayer</td>
</tr>
</tbody>
</table>

Note: The packages containing “mo2” are files from MapObjects version 2.3.
This project was developed in Java platform, edited and compiled in Eclipse IDE. Eclipse provides a rich Java development environment and plug-in system. It easily works with MapObjects Java Edition when appropriate libraries are added. JAR files are added from the MapObjects folder.

Following steps explain how to configure Eclipse IDE and MapObjects in detail.

2. Install Eclipse IDE.
3. Install MapObjects Java Objects given by Dr. Carl Eckberg [3].
4. Install Java Software development kit.
5. Launch Eclipse IDE from Start → Programs → Eclipse by double clicking the eclipse icon. Figure 6.1 shows the process for step 4. 6.
6. Create a new java project. Go to File → New → Java Project, it will open a window. Name the project and click next. See Figure 6.2.
7. Click on libraries tab, click on Add Libraries, now select User Libraries and click next. See Figure 6.3.
8. It will open a preference window click User Libraries and then click new and enter the name of the library. See Figure 6.4.
9. Now click on Add Jars button, navigate to the folder where MOJO is installed.
10. Select all the JAR files. See Figure 6.5.
11. Make sure that the newly created library checkbox is selected. See Figure 6.6.
12. Next step is to attach Javadocs to the jar files. Expand one of the JAR file and click on Javadoc location and then click edit. See Figure 6.7.
13. In the Javadoc location path, navigate to the Javadoc folder in ESRI documentation directory and then click OK. See Figure 6.8.
14. Now repeat step 13 for all other JAR files.
Figure 6.1. Starting Eclipse.

Figure 6.2. Creating a new Java project.
Figure 6.3. Creating a new library.

Figure 6.4. List of existing libraries.
Figure 6.5. Adding external JARs for MOJO support.

Figure 6.6. Configure Java build path.
Figure 6.7. Specifying the Javadoc location.

Figure 6.8. Specifying the Java code style.
CHAPTER 7

MAPOBJECTS TOOLBAR

JavaBeans are reusable software components for Java. Multiple beans are provided by MapObjects Java Edition which the user can drag and drop onto eclipse workspace to provide basic functionality (see Figure 7.1). MapObject toolbar provide certain basic functionality like zoom in, zoom out, pan, identify, etc. which are essential for any visual display. This chapter discusses how this toolbar have been integrated into the GIS tool and basic functionality of each tool. Following toolbars have been used:

1. Zoompan toolbar
2. Selection toolbar
3. Java Toolbar

Figure 7.1. MapObjects toolbar.

7.1 COM.ESRI.MO2.UI.TB.ZOOMPANTOOLBAR

The Zoompan toolbar provides user to change the map visual in a number of ways. It provides facilities like zooming, panning, etc. The actions performed by the various icons in the Toolbar are explained here:

- **Previous Extent** – Zoom to the previous extent stored in the history.
- **Next Extent** – Zoom to the next extent stored in the history.
- **Zoom To Active Layer** – It zooms the map to all selected features in the selected layer.
- **Zoom To Full Extent** – It zooms the map to extent all layers within the map.
- **Zoom In** – It provides a function to click or drag a rectangle on the map to zoom in (To view the selected area bigger).
- **Zoom Out** – It provides a function to click or drag a rectangle on the map to zoom out (To view the selected area smaller).
• **Pan** – Provides a tool for dragging the map to a new location without altering the zoom level.

• **Pan In One Direction** – Provides a tool for dragging the map in one direction (North, South, East, West).

• **Identify** – Identifies the selected layer and display the features of that layer in an attribute table.

**7.2 com.esri.mo2.ui.tb.SelectionToolbar**

This Toolbar provides functions that perform future selection bases upon Attribute or query.

• **Search** – Opens a dialog for locating features based on a predefined “stored query “.

• **Find** – It opens a dialog for locating features whose attributes contain an end user provided string.

• **Query Builder** – Opens a dialog for locating features based on a query that an end user gives.

• **Select Features** – It provides a tool for selecting specified areas by different means like rectangle, polygon, line &circle in the map.

• **Clear All Selection** – As the name suggests, it clears the current selection.

• **Buffer** – It allows to construct a polygon around the currently selected features.

• **Attributes** – It displays the attributes of the layer selected on the TOC.

**7.3 Java/Custom Toolbar**

The Project Toolbar allows user to add a layer, print a map image to file, etc. The actions performed by the various options in the project toolbar are as follows [6]:

• **Print Map** – Use this option to print the map for the selected layer(s) in the TOC.

• **Add Layer** – Use this option to add layer in the TOC which shows different map.

• **Pointer** – This is use to resume the cursor from previous selection.

• **Hot Link** – It provides a functionality where in user can get more information about the locations visible on the map like the link to wiki, images, etc.

• **XY Tool** – It allows the user to add a layer of points using a .CSV file having the latitudes and longitudes.

• **Distance Tool** – It allows user to find the distance between two points on map in miles as well as kilometers.

• **Tool help tool** – Left click on this tool and right click on the tool to get the tool information.
CHAPTER 8

CUSTOM TOOLBAR

In addition to the toolbar provided by MapObjects, a custom toolbar has been created using Java to enable a better interaction of user and the GIS tool. See Figure 8.1.

The tools that are present in first set of the MapObjects toolbar is the custom toolbar, it has been discussed in detail in this chapter to give a clear view of the functionality of each icon (see Table 8.1).

Table 8.1. Custom Toolbar

<table>
<thead>
<tr>
<th>Number</th>
<th>Tool Name</th>
<th>Class or Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Print</td>
<td>com.esri.mo2.ui.bean.Print</td>
</tr>
<tr>
<td>2</td>
<td>Add Layer</td>
<td>com.esri.mo2.ui.tb.LayerToolBar</td>
</tr>
<tr>
<td>3</td>
<td>Pointer</td>
<td>Arrow.java</td>
</tr>
<tr>
<td>4</td>
<td>Hotlink</td>
<td>HotLink.java</td>
</tr>
<tr>
<td>5</td>
<td>XY Tool</td>
<td>AddXYtheme.java</td>
</tr>
<tr>
<td>6</td>
<td>Distance Tool</td>
<td>DistanceTool.java</td>
</tr>
<tr>
<td>7</td>
<td>Help Tool</td>
<td>HelpTool.java</td>
</tr>
</tbody>
</table>

8.1 PRINT

It provides the functionality to make a print out of the map; when a user clicks on this icon it opens a dialog box to select the type of printer. The snapshot of the print screen is shown in Figure 8.2.
This tool allows the user to add a new layer to the table of contents (TOC). Figure 8.3 shows how the layer can be added to the map. User has to first click on the plus symbol in the toolbar or click on “Add Layer“ tab in the file menu; it will open a window from where the user can navigate to a specific location to add the layer; also there is an option to make the layer visible/invisible by checking the checkbox in TOC.

Steps to add layer:
1. Click on the plus sign from the custom toolbar, or “ADD LAYER “ from the file menu.
2. A new window will appear, browse to the location of file to be added.
3. Select the shape file (.shp file).
4. Click Ok.
5. New Layer would be added to the existing map.

Steps to remove layer:
1. Click on the “REMOVE LAYER “ from the file menu.
2. By default it will delete the current selected layer.
3. To delete a particular layer, first select the layer and then click on the remove layer.
8.3 POINTER

The Pointer tool is the default tool selected when the application begins, when the user selects any particular tool while working on application and then wants to switch back to the default tool, it can be achieved by clicking on the pointer tool.

8.4 HOTLINK

Hotlink allows the user to explore more about the layer/feature selected by them. When the user clicks on the hotlink icon, he then selects a layer feature to get detailed information, it opens up a dialog box displaying the information in form of text, image and it also presents the user with a choice to view detail or see the web link and/or a video related to the fuel resource. Depending upon the choice made by the user the appropriate content is loaded within the application. The hotlink tool icon looks like this. The popup window with options is shown in Figure 8.4 and 8.5.
8.5 **XY TOOL**

It gives the user the ability to display new locations on the map by using a comma separated value file (.csv). When the user clicks on this icon a pop up window will show up allowing the user to choose the csv file having the latitudes and longitudes of the location to be displayed. It displays the data points immediately on the map. See Figure 8.6.

8.6 **DISTANCE TOOL**

It gives the user the ability find out the distance between any two points on the map in miles as well as in kilometers. The display is at the bottom of the map in the status bar. See Figure 8.7.

8.7 **HELP TOOL**

After clicking on this tool the user is supposed to right click on any tool to open the help content which gives usage information about that particular tool (please refer Figure 8.8).
Figure 8.5. Hotlink tool pick window.

Some info: Coal. The Kayenta mine is a surface coal mine operated by Peabody Western Coal Company (a subsidiary of Peabody Energy) on the Navajo Indian Reservation.

Fuel/Energy Production(units): 8.7 Million tonnes

Web Site:

Watch my video
Figure 8.6. GIS tool with XY tool window.

Figure 8.7. GIS tool with distance tool window.
Figure 8.8. GIS tool with help tool window.
CHAPTER 9

ADDITIONAL TOOLS

The GIS tool for learning about the fuel resources in American continent gives a couple of more tools in addition to the custom toolbars and MOJO tools to provide better user interaction and know how. Some of the tools and their functionality are mentioned in this chapter.

9.1 LAYER CONTROL

This GIS tool works on the principle of layers. Any shapefiles are displayed as a ‘layer’ which are one upon the other. The layer order affects the visibility of the shapefile it is displaying. Select the layer you want to promote or demote from the menu provided by this tab. To get an idea how this feature works, use multiple shapefiles to promote and demote the layers. See Figure 9.1 and Figure 9.2.

Figure 9.1. Layer control before using the menu.
Figure 9.2. Layer control after promoting a states layer.

9.2 HELP TAB

There is a help tab in the menu bar at the top of the screen, when the user click on it a dropdown list appears which look like the one shown in Figure 9.3. It has four different options and a sub menu providing help about ‘Layer Control’, ‘Legend Editor’ and ‘Table of Contents’. Rest of the options being:

1. **Help Tool** – Information about Help tool itself
2. **Contact Us** – Provides contact information about the developer.
3. **About MOJO** – Some basic information about MOJO.

Figure 9.4 shows an example of the usage of ‘Help Tool’ option selection from the ‘Help’ tab.

9.3 CO-ORDINATES DISPLAY

It is the functionality that tracks the co-ordinates of the pointer on the map and displays it in the bottom status bar, it has been implemented from the MOJO class guide by Dr. Carl Eckberg (see Figure 9.5).
Figure 9.3. Help tool tab.

Figure 9.4. Help tool usage.
9.4 LEGEND EDITOR

This is another important tool that GIS toolkit provides. It allows the user to modify the rendering representation of the data and is known as Layer Properties tool. It enables a user to change the properties of a layer such as the symbol, color, etc. The layer properties tool has been implemented using the com.esri.mo2.ui.ren.LayerProperties class. It can be accessed via the File Menu. Figure 9.6 shows the Layer Properties window. It has three tabs: Symbols, Labels and General. In the ‘Symbol’ tab we get option to select the single symbol, graduated symbol and unique symbol. By using these symbols the layer feature can be represented. By using label tab we can provide labels to the layer feature and it can be labeled by different categories stored in the attribute table. By using general tab, users can select the option of when the feature layer should be shown. A layer with unique colors is shown in Figure 9.7.
Figure 9.6. Legend editor for selected layer.

Figure 9.7. GIS tool with labeling on selected layer.
CHAPTER 10

SCREENSHOTS

This chapter contains the screenshots from the entire project, some of the screenshots have already been covered in the earlier chapters:

- Shapefile wizard is shown in Figure 10.1.
- Shapefile browser is shown in Figure 10.2.
- After adding the shapefile shown in Figure 10.3.
- Loading of a world map shapefile is shown in Figure 10.4.
- CSV file wizard is shown in Figure 10.5.
- CSV file loaded shown in Figure 10.6.
- Hotlink pick on Kayenta Mine sample is shown in Figure 10.7.
- Hotlink Dialogue box Hotlink click on Youtube video button shown in Figure 10.8
- Sample welcome webpage for a fuel shown in Figure 10.9
- Welcome page complete content shown in Figure 10.10
- Sample webpage regarding current status of the three countries in terms of that fuel shown in Figure 10.11
- Sample webpage regarding coal extraction methods shown in Figure 10.12
- Sample webpage regarding controversial events in history of that fuel shown in Figure 10.13
- Sample webpage showing future prediction of that particular fuel production for all three countries shown in Figure 10.14
- Sample Query builder tool. Its usage is discussed in Chapters 1-9. See Figure 10.15.
- Usage of Identify tool shown in Figure 10.16
- Usage of ‘Find’ tool shown in Figure 10.17
- Usage of Buffer tool shown in Figure 10.18
- Figure 10.19 shows result of using Buffer tool.
Figure 10.1. Loading a shape file wizard.

Figure 10.2. Shape file browser wizard.
Figure 10.3. After finding a .shp file.

Figure 10.4. World map shape file loaded screen.
Figure 10.5. csv file wizard.

Figure 10.6. csv file loaded.
Figure 10.7. Hotlink tool pick on Kayenta Mine, Arizona.

Figure 10.8. Hotlink click on ‘Watch my Video’ leading to YouTube.
Figure 10.9. Sample welcome page of the information webpage.

Figure 10.10. Welcome page content.
Current Status

Towards the end of the twentieth century, the demand for coal generated by the exponential growth of the railroad industry and strip mining operations, began to suffer from changes in coal mining practices. In the United States, with reserves of more than 200 billion tons of coal, coal became the dominant source of energy. The growth of coal-based power generation has slowed considerably from what it was in the late 1980s.

Pennsylvania was the fourth largest coal-producing state in the United States. Pennsylvania produces two kinds of coal, anthracite (hard coal) and bituminous (soft coal).

Figure 10.11. Webpage regarding current status of the three countries.

Extraction Methods

Auger mining

In the eastern United States, auger mining is used on shallow terrains. It requires a surface cut to allow the auger to access the bed. It is also used in the case of coal left from underground mining. In the western United States, auger mining is used in conjunction with strip mining. Coal mining by the auger method involves boring horizontal holes in an exposed face of the coal, and loading of removed coal. Single, double, or triple auger heads can remove up to 20 inches of coal for a distance of over 200 feet. It is also used where the terrain is too steep for overburden removal.

Open pit mining

In open-pit mining, overburden is removed and placed outside the mining area. The overburden can be removed with either scalper or dozer and loaded into trucks. Mining begins by drilling and blasting waste rock to expose the coal seams, advancing additional overburden, and removing and transporting the coal. The pit increases in size and depth as mining progresses, and it is unusual that overburden, once removed, is ever returned to the pit. The open pit method is sometimes used in coal mining where numerous seams lie parallel to each other and outcrops on mesas and plateaus.

Room & pillar mining

Room-and-pillar mining has been used in the United States longer than any other.

Longwall mining

Longwall mining is used most effectively in uniform coal seams of medium thickness (42 to 60 inches). A single longwall mine has a width of

Figure 10.12. Sample webpage regarding coal extraction methods.
Figure 10.13. Sample webpage regarding controversial history of coal mining.

Figure 10.14. Sample webpage showing forecast stats.
Figure 10.15. Query builder tool.

Figure 10.16. Usage of identify tool on map of India.
Figure 10.17. Usage of “find” tool.

Figure 10.18. Usage of “buffer” tool.
Figure 10.19. After using buffer tool.
CHAPTER 11

CONCLUSION AND OBSTACLES

Working on this topic was an interesting yet challenging task. Especially, the data collection part was crucial since this topic has both ecological and political ripples. The scope of the project started off with only USA into consideration. Later, with Dr. Eckberg’s and my mutual decision we decided to include data on Canada and Mexico as well. The number of types of fuel resources is itself a lot and the effort to know about them was fun. The layered approach is a point of distinction about MOJO. Many layers were created from scratch which includes collecting data, getting the location (latitudes and longitudes), creating a .CSV file, converting the .CSV file to a shape file. The polygon layer for Natural Gas resources was created using a Polygon tool which was another thesis work under Dr. Carl Eckberg. This helped a lot. Designing of webpages and deciding their content was another point of challenge in the project which was resolved with the help of Dr. Carl Eckberg.
CHAPTER 12

FUTURE ENHANCEMENTS

This GIS tool for learning about fuel resources in the North American continent can be extended in several ways. It is coded using JAVA, which is easy to write and allows the programmer to reuse the existing classes and thus enables an easy addition of other features with minimum code changes. Few enhancements to increase the scope of the thesis are listed here:

- Making it an entirely web based application, to make it available to a larger extent.
- Converting it to a smart devices application.
- Timeline can be added to show the data from 1900 to the current date.
- Some customized tools can be added like the one which can update it with all the current/latest information.
- The webpages providing more information which is the most important aspect of the project can be enhanced to become more interactive and better looking using latest web designing tools.
- Implementing interactive Quizzes for users to help them to get better understanding about the fuel resources can be developed.
REFERENCES


