A Content Delivery Method and Context-Sensitive User Interface for a Web-Enabled Geographic Information System

by

Kalpesh Raicha

Bachelor of Computer Science, Pune University, 2003

A Project

Submitted to the Graduate Faculty

of the

University of North Dakota

in the partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota
December 2006
This project document submitted by Kalpesh Raicha in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota has been read by the Faculty Advisor under whom the work has been done and is hereby approved.

(Wen-Chin Lin)
(Faculty Advisor)

This report document meets the standards for appearance, conforms to the style and format requirements of the Computer Science Department of University of North Dakota, and is hereby approved.

(Signature)
Graduate Director

10/15/06
Date
PERMISSION

Title: A Content Delivery Method and Context-sensitive User Interface for a Web-Enabled Geographic Information System.

Department: Department of Computer Science

Degree: Master of Computer Science

In presenting this report in partial fulfillment of the requirement for a graduate degree from the University of North Dakota, I agree that the Department of Computer Science shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my work or, in his absence, by the Chairperson of the Department.

Signature

Date 10/05/2006
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERMISSION</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>viii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ix</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Significance</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Objectives</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Project Outline</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Report Organization</td>
<td>3</td>
</tr>
<tr>
<td>II. BACKGROUND</td>
<td>5</td>
</tr>
<tr>
<td>2.1 The Upper Midwest Aerospace Consortium (UMAC)</td>
<td>5</td>
</tr>
<tr>
<td>2.2 The Digital-NGP (Northern Great Plains) or DNGP</td>
<td>6</td>
</tr>
<tr>
<td>2.3 The Unified Modeling Language and IBM Rational Software Modeler</td>
<td>7</td>
</tr>
<tr>
<td>2.3.1 UML</td>
<td>7</td>
</tr>
<tr>
<td>2.3.2 IBM Rational Software Modeler</td>
<td>8</td>
</tr>
<tr>
<td>III. REQUIREMENTS AND SPECIFICATION</td>
<td>9</td>
</tr>
<tr>
<td>3.1 System Requirements</td>
<td>9</td>
</tr>
<tr>
<td>3.2 System Specification</td>
<td>10</td>
</tr>
<tr>
<td>3.2.1 The Map Option</td>
<td>10</td>
</tr>
<tr>
<td>3.2.2 The View Option</td>
<td>11</td>
</tr>
<tr>
<td>3.2.3 The Map Tools Option</td>
<td>15</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1. The main menu .................................................................................................................. 10
2. The options for the map menu .......................................................................................... 11
3. The options for the view menu ....................................................................................... 11
4. The GIS Layer window .................................................................................................. 12
5. The DNGP Layer window .............................................................................................. 13
6. The download and image control window ...................................................................... 14
7. The reference map window ............................................................................................. 14
8. The menu options of map tools ....................................................................................... 15
9. The slide control window ................................................................................................ 16
10. The options of the account sub-menu ........................................................................... 16
11. The options of the AOI sub-menu ................................................................................ 17
12. A use-case diagram ....................................................................................................... 20
13. An activity diagram for the options of the map menu ..................................................... 22
14. An activity diagram for the options of the view menu .................................................... 23
15. An activity diagram for the sub-menu options of download and image control ............ 24
16. An activity diagram for the options of the map tools menu .......................................... 25
17. An activity diagram for the sub-menu option slide show of map tools ......................... 26
18. An activity diagram for the sub-menu options of navigation tools menu ....................... 27
19. An activity diagram for the options of the user menu .................................................... 28
20. An activity diagram for the sub-menu options of AOI menu .......................................... 29
21. A state diagram for the system ..................................................................................... 30
22. A deployment diagram for the system .......................................................................... 31
23. The menu before user logs in................................................................. 46
24. The interface when user clicks LOGIN.................................................. 47
25. The menu after user logs in................................................................. 48
26. The interface with GIS Layer, DNGP Layer and Reference Map visible...... 49
27. The interface when GIS Layer, DNGP Layer and Reference Map are invisible... 50
28. Download and Image Control window (when product type is Original Image)..... 51
29. Download and Image Control window (when product type is NDVI)............. 52
30. GIS layer and DNGP images layer are resized......................................... 53
ACKNOWLEDGMENTS

I would like to express sincere thanks to the many people who have contributed to the completion of this study. I would like to thank Dr. WenChen Hu for his support and advice throughout the duration of this work. It was a great pleasure for me to work with him. Many thanks to my supervisor, Dr. Xiaodong Zhang in UMAC, for the encouragement, advice and assistance on this project. The meetings organized by him, helped me greatly to solve the problems and it was wonderful working with him. Special thanks go to my family for their constant love and understanding, without which would have made this work impossible.
ABSTRACT

Digital-NGP (Northern Great Plains) or DNGP is an online geographic information system (GIS) for archiving, searching, and delivering remote sensing images. It is developed and sponsored by the Upper Midwest Aerospace Consortium at the University of North Dakota. This system has been developed in phases with several developers working on it. The current design suffers several architectural drawbacks. For example, developers usually have problems understanding the changes made by the earlier developers which in turn makes the future development and maintenance difficult. Also, there is still room for improvement of system interface such as user friendliness.

This project is to enhance the functions provided by the DNGP. It includes two major tasks:

- Design and develop a consistent, robust content delivery system.
- Provides Windows-like user interface (with which most users are familiar) with a context-sensitive help system.

The final system will allow the future developers to easily manage the content, layout, and overall appearance of the system independently. Users can use the system menu to access different functions provided by the DNGP and benefit from the context sensitive help that delivers the help topics related to the current context.

The research process involves requirement analysis and specification, design, validation, and testing. The process of requirement analysis includes two steps: i) understanding the user requirements and ii) developing a model based on those requirements. The model reflects the behavior of the system and shows the different functions provided by the system. It is then translated into high-level and low-level designs. Verification and
validation make sure that the system fulfills user requirements and that every function works according to the specifications. The final system is an Internet-enabled Oracle database system using MapServer as spatial image engine, PHP (Hypertext Preprocessor) and JavaScript for Web programming, and CSS (Cascading Style Sheets) for rendering dynamic user interface. The proposed system also modularizes the DNGP and makes the system scalable and easy for future development and maintenance.
CHAPTER I

INTRODUCTION

1.1 Significance

The Upper Midwest Aerospace Consortium (UMAC) supports a repository of digital imagery that is of value to agriculture and has developed a delivery system called as Digital–NGP (Northern Great Plains) to make this data available to UMAC clients in upper Midwest region. Digital-NGP or DNGP is an online geographic information system (GIS) for archiving, searching, and delivering remote sensing images. It is developed and sponsored by the Upper Midwest Aerospace Consortium at the University of North Dakota. This system has been developed in phases with several developers working on it. The current design suffers several architectural drawbacks. For example, developers usually have problems understanding the changes made by the earlier developers which in turn makes the future development and maintenance difficult. Also, there is still room for improvement of system interface such as user friendliness. This software engineering project analyzes the needs of the users and designs and implements an enhanced version of DNGP.

1.2 Objectives

The purpose of DNGP is to provide an easy tool for clients of UMAC to download geospatial data. Through the DNGP Website, users can browse and download remote sensing images covering Montana, Wyoming, Minnesota, North Dakota and South
Dakota. DNGP Website can also be used to explore other geospatial data including highways, rivers, lakes, shaded relief, and census data.

The current Website is a three-tier Web database application. Tier 1 is the user interface. It runs on a desktop PC or workstation (client) and uses a standard graphical user interface (Web browser). In our application, our Web pages act as the Tier 1, which is requested by the user through his Web browser and is sent there from our Web server for display. Then user can use the controls on the web pages to send further queries to Web server, Web server will respond to these queries and send result web pages to user’s Web browser. Tier 2 is the functional module. This level actually processes data. It consists of several separate modules running on an application server. In the existing DNGP, this tier consists of Apache web server, Mapserver4.0, and PHP programs. Tier 3 is the database management system (DBMS). A DBMS on a host computer stores the data required by the middle tier. In the DNGP, this tier is an Oracle 9i database management system in which we store our image data.

This system has been developed in phases with several developers working on it. The current design suffers several architectural drawbacks. For example, developers usually have problems understanding the changes made by the earlier developers which in turn makes the future development and maintenance difficult. In the existing system, it is difficult for the developers to add new functions to the system. The user interface is cluttered and there is not much room available to add new buttons for accessing new functions. The content, layout and appearance of the system are not independent of each other. Also, there is still room for improvement of system interface such as user friendliness. This main goal of this project is to enhance the functions provided by the DNGP. It includes two major tasks:
• Design and develop a consistent, robust content delivery sub-system.

• Provide Windows-like user interface (with which most users are familiar) with a context-sensitive help system.

1.3 Project Outline

The development process involves requirement analysis and specification, design, validation, and testing. The process of requirement analysis includes two steps: i) understanding the user requirements and ii) developing a model based on those requirements. The model reflects the behavior of the system and shows the different functions provided by the system. Verification and validation make sure that the system fulfills user requirements and that every function works according to the specifications. The final system is an Internet-enabled Oracle database system using MapServer as spatial image engine, PHP (Hypertext Preprocessor) and JavaScript for Web programming, and CSS (Cascading Style Sheets) for rendering dynamic user interface.

This system allows the future developers to easily manage the content, layout, and overall appearance of the system independently. Users can use the system menu to access different functions provided by the DNGP and benefit from the context sensitive help that delivers the help topics related to the current context.

1.4 Report Organization

The next chapter gives background information on UMAC, DNGP and UML. Chapter III describes the requirements and specifications of this application. It contains the project requirements stated by the client and the specifications derived from those requirements. Chapter IV describes software design with the IBM Rational Software Modeler. This chapter explains various functions provided by the system using different UML
diagrams. Chapter V focuses on the implementation. It gives an overview of the different modules using some code samples. Chapter VI implements verification and validation. It describes the different strategies used for testing the application and the results obtained from those tests. Chapter VII contains the conclusion.
CHAPTER II
BACKGROUND

This chapter gives a background study of this research project. It includes four sections: (i) The Upper Midwest Aerospace Consortium (UMAC), (ii) The Digital-NGP (Northern Great Plains) or DNGP (iii) The Unified Modeling Language and, (iv) IBM Rational Software Modeler.

2.1 The Upper Midwest Aerospace Consortium (UMAC)

The UMAC is led by the University of North Dakota, and is comprised of eight universities in the five different states of North Dakota, South Dakota, Montana, Wyoming, and Idaho. It develops products and services for Agriculture, for Natural Resource Management, and for K-12 Education, using satellite imagery and other spatial technologies and also provide information and educational outreach services to the general public with respect to regional impacts of environmental and climatic change. UMAC provides products, services, and information to the general public by operating as a Public Access Resource Center, or PARC, focused principally on the agriculture, natural resource management, and education communities.

The goal of the Upper Midwest Aerospace Consortium (UMAC) is to provide information about the environment that enables people to make decisions improving their [1]

- economic competitiveness,
- quality of Life, and
• educational preparedness.

The primary source of the information is the data acquired by satellites and aircraft.

Beneficiaries of the information include:

• farmers practicing precision agriculture,
• ranchers seeking optimum grazing capacities,
• foresters engaged in sustainable forestry,
• educators teaching responsible stewardship, and
• students, K through lifelong learners, of earth system science.

2.2 The Digital-NGP (Northern Great Plains) or DNGP

The DNGP (Digital Northern Great Plains) project is sponsored by the UMAC. The goal of this project is to build an integrated web-based Geographic Information System. This system provides the access to Landsat imagery and Landsat-derived products (NDVI, Green NDVI, and Sugar Beet Yield) from the current season and the entire UMAC Landsat archive (over 400 scenes) covering Montana, Wyoming, Minnesota, North Dakota and South Dakota. DNGP can also be used to explore other geospatial data including highways, rivers, lakes, shaded relief, and census data. The purpose the DNGP project is to provide an easy tool for clients of the UMAC to download geospatial data. The DNGP clients include farmers, ranchers and researchers. UMAC has built the DNGP Website, which is an interactive maps system. Through DNGP Website, users can browse and download remote sensing images covering Montana, Wyoming, Minnesota, North Dakota and South Dakota. DNGP Website can also be used to explore other geospatial data including highways, rivers, lakes, shaded relief, and census data.
2.3 The Unified Modeling Language and IBM Rational Software Modeler

2.3.1 UML

The Unified Modeling Language (UML) is Object Management Group’s (OMG) most-used specification, and the way the world models not only application structure, behavior, and architecture, but also business process and data structure [2]. UML is used to specify, visualize, and document models of software systems, including their structure and design, in a way that meets all of these requirements. It can be used to model just about any type of application, running on any type and combination of hardware, operating system, programming language, and network. It is built upon fundamental OO concepts, it's very helpful for modeling object-oriented languages and environments such as C++, Java, but it can also be used to model non-OO applications. Software modeling and design is very important for large applications. A model plays the similar role in software development that blueprints and other plans play in the construction of a building. Using a model, software developers can assure themselves that project’s functionality is complete and correct, end-user needs are met, and the design supports requirements for scalability, robustness, security, extendibility, and other characteristics. Surveys show that large-software projects have a high probability of failure - in fact, it's more likely that a large-software application will fail to meet all of its requirements on time and on budget than that it will succeed.

UML 2.0 defines thirteen types of diagrams, divided into three categories. Six diagram types represent static application structure; three represent general types of behavior; and four represent different aspects of interactions [3]:

1. Class diagram
2. Object diagram
3. Package diagram
4. Component diagram
5. Deployment diagram
6. Use case diagram
7. Sequence diagram
8. Collaboration diagram
9. Interaction overview diagram
10. State machine diagram
11. Activity diagram
12. Composite structure diagram
13. Interaction flow diagram
• **Structure Diagrams** include (i) a class diagram, (ii) an object diagram, (iii) a component, (iv) a composite structure diagram, (v) a package diagram, and (vi) a deployment diagram.

• **Behavior Diagrams** include the use case diagram (used by some methodologies during requirements gathering); activity Diagram, and state machine diagram.

• **Interaction Diagrams**, all derived from the more general behavior diagram, include the sequence diagram, communication diagram, timing diagram, and interaction overview diagram.

### 2.3.2 IBM Rational Software Modeler

There are many tools available in the market for software modeling and design. The one used in this project is the IBM Rational Software Modeler [4]. It is a customizable, UML 2.0-based visual modeling and design tool that enables architects, systems analysts, designers and others involved in the development process to clearly document and communicate processes, flows and designs. Rational Software Modeler also integrates with other tools in your environment to support team development. Rational Software Modeler is a full-featured UML2 modeling tool. It provides a powerful diagram editor which allows the creation of expressive and interactive diagrams. Customized UML profiles are supported and model constraints can be defined.
CHAPTER III

REQUIREMENTS AND SPECIFICATION

Requirements are the users’ need for the system. A requirements specification is an agreement between the end user and system developer; it is what the implementation must achieve. Dr. Xiaodong Zhang, an assistant professor at the UMAC, provided the requirements of this project.

3.1 System Requirements

The basic requirements of this project requested by Dr. Xiaodong Zhang are:

- A dynamic menu for accessing the functions provided by the system: With the current design it is difficult to add new functionalities to the system. The current system does not have a menu. Whenever a new function is added to the system the developer has to add a new button or some feature to the user interface so that the user can access the new function using that feature. But the current user interface is too cluttered and there is not much room available to add new buttons. In the new design we have a dynamic menu which is used to access the functions provided by the system. The menu can have unlimited number of items and of any depth.

- A windows-like user interface: Users are able to resize, drag, minimize, and maximize, the windows. The system provides a windows-like user interface with which most users are familiar.
• A context sensitive help system: The existing system does not have a user friendly help system. The current help displays help information related to every topic at once. In the new design we have a context sensitive help system. It displays help according to the current context. For, example if the user is using one of the navigation tools like zoom-in or zoom-out, the help system displays help information related to zoom-in or zoom-in.

• Content, layout, appearance of the system being independent of each other and central control over appearance of the system using Cascading Style Sheets (CSS). In the new design the content, layout and appearance of the system are independent of each other. The developer can change the appearance of the system by just making changes in the style sheets without affecting the content and layout.

3.2 System Specifications

As per the requirements, a graphical user interface has been developed with a dynamic menu and Windows like interface that can be easily understood by users. Once the user logs in, main menu is displayed at the top with the following options: (i) Map, (ii) View, (iii) Map Tools, (iv) User and, (v) Help.

![Image of the main menu](image.png)

**Figure 1**: The main menu.

3.2.1 The Map Option
The map option on click displays a sub-menu with the following option:

![Image of map menu options](image)

**Figure 2: The options of the map menu.**

1. **Save Map**: This on click displays the system default save dialog. It lets you save the map as it is in the map window as an image file (only in the PNG format).

2. **Print Map**: This on click displays the system default print dialog. It lets you print the map as it is on the map window on a paper.

### 3.2.2 The View Option

The View option on click displays a sub-menu as shown in Figure 3 with the following options: (i) Select GIS Layers, (ii) Select DNGP Layers, (iii) Download & Image Control, (iv) Reference Map and , (v) Toolbar , which are described as follows:

![Image of view menu options](image)

**Figure 3: The options of the view menu.**
1. **Select GIS Layers**: This on click displays GIS Layer Window. This button can also be used to hide this window when it is already displayed. The layer window features many layers which convey information about various spatial features. Multiple layers can be overlaid on one another to make a map according to your needs and can be viewed in the Map window, or printed and saved. The layers are categorized into different groups which can be minimized or maximized by clicking on the group names. The GIS Layer Window (Figure 4) can be resized, minimized or moved in the web browser according to your convenience.

![GIS Layers](image)

Figure 4: The GIS Layer window.

2. **Select DNGP Layers**: This on click displays DNGP Layer Window. This button can also be used to hide this window when it is already displayed. The layer window consists of a list of Raster images in GeoTIFF format. Different images cover different areas on the map. The image list varies according to the map extent. At any
instance, the list shows images that are covering the area that is viewed in the map window. Each image is shown with a name which describes the acquisition date, sensor type, and coverage information. If an image covers the current map window completely, then it is shown as full otherwise it is shown as partial. The image list can be refined by selecting a particular sensor type in the Select Box on the top of the panel to view images only from that sensor covering the displayed area in the map window. Default sensor type shows all the images from all the sensors. The DNGP Layer Window (Figure 5) can be resized, minimized or moved in the Web browser at your convenience.

![Figure 5: The DNGP Layer window.](image)

3. **Download and Image Control:** This on click displays the download and image control window. This feature allows the user to manipulate the chosen raster image. The buttons over the control panel will be enabled only if an image is selected.
and displayed in the map window. You can download either all the bands or the current view of the raster image in the map window (with selected band combinations; default is RGB) in GeoTIFF format. The Image Control and Download window (Figure 6) can be minimized or moved in the web browser according to your convenience.

![Download and Image Control Window](image1)

**Figure 6:** The download and image control window.

4. **Reference Map:** This button can be used to hide and show the reference map. The reference map gives a reference of the location that is being viewed in the Map window with respect to the entire UMAC coverage by highlighting with a red colored pointer. You can change the current location displayed in the Map window by clicking on any other location on the reference map.

![Reference Map Window](image2)

**Figure 7:** The reference map window.
5. **Toolbar**: This button can be used to hide and display the toolbar on the side of the Map window.

### 3.2.3 The Map-Tools Option

The Map-Tools Option on click displays a sub-menu with following options:

![Figure 8: The menu options of map tools.](image)

1. **Navigation Tools**: This option on click displays a sub-menu with the following options:
   - **Zoom-in**: This button lets users view a larger area with less detail.
   - **Zoom-out**: This button lets users view part of the map in greater detail.
   - **Re-center**: This button allows you to pan (shift) the map to the desired location.
   - **View Whole UMAC**: This button lets you view the entire region covered by UMAC (North Dakota, South Dakota, Montana, Wyoming, Minnesota and Idaho).

2. **Query**: This icon shows the attributes associated with the enabled layers in the Layer window when clicked on any location on the map.
3. **Slide Show**: This button on click displays the slide show window. This window can be minimized or moved in the web browser according to your convenience.

![Slide Control](image)

**Figure 9**: The slide control window.

### 3.2.4 The User Option

This option appears in the menu only after logging in; the User option on click displays a sub menu with the following options:

![Digital Northern Great Plains](image)

**Figure 10**: The options of the account sub-menu.

1. **Account**: This button allows you to manage your account. This option on click displays the sub menu with (i) edit my account, (ii) change my password and, (iii) change my secret question entries.

   - **Edit my account**: After clicking on this menu option a new window pops up showing your current account information. You can change the information in the window.

   - **Change my password**: A new window pops up to let you create a new password but it requires you to enter your current password.
• **Change my secret question:** After clicking on this menu option, a new window pops up to let you change your secret question and answer which would be required if you forget your password.

![Figure 11: The options of the AOI sub-menu.](image)

2. **AOI:** This button allows you to create an Area of Interest (AOI) by entering the bounding coordinates and Delete AOIs. This option on click displays the sub menu with Create AOI, Delete AOI, View AOI, Request AeroCam, Request AgCam entries.

  - **Create AOI:** After clicking on this menu option a new window pops up with options to create a new AOI. The AOI created can be accessed from the View AOI menu option.

  - **Delete AOI:** After clicking on this menu option a new window pops up showing the AOIs that you have created. Select the AOI Names you want to delete and click Delete AOI. AOI values deleted are removed permanently from the database and are irrecoverable.
• **View AOI**: This option on click displays a window. If you have already created AOIs, they will be displayed as a list. Upon selecting an AOI name from the list, the map window displays the map covering that AOI.

• **Request AeroCam and Request AgCam**: After clicking on these options a new window pops up. Follow the instructions to send request for the AeroCam and AgCam image acquisition.
CHAPTER IV
SOFTWARE DESIGN

System design uses the Unified Modeling Language (UML) and IBM Rational Software Modeler based on the requirements analysis. The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

4.1 A Use Case Diagram

A use case diagram is a graph of actors, a set of use cases, possibly some interfaces, and the relationships between these elements. The relationships are associations between the actors and the use cases, generalizations between the actors, and generalizations, extend, and includes among the use cases. The use cases may optionally be enclosed by a rectangle that represents the boundary of the containing system or classifier. Figure 12 shows a use-case diagram for registered users of the system.
Figure 12: A use-case diagram.
4.2 Activity Diagrams

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel. Activity Diagrams are also useful for: analyzing a use case by describing what actions needs to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes. Activity diagrams show the flow of activities through the system. Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. The branch describes what activities will take place based on a set of conditions. All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch. After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

The system provides a dynamic menu which users can use in order to access various functions provided by the system. The Activity diagrams that I have created describe the various menu options. Once the user logs in, the main menu options that are available to the registered users are “Map”, “View”, “Map Tools”, “User”, “Help”. The Activity diagrams that describe the various use cases of the system are shown below.
Figure 13: An activity diagram for the options of the map menu.
Figure 14: An activity diagram for the options of the view menu.
Figure 15: An activity diagram for the sub-menu options of download and image control.
Figure 16: An activity diagram for the options of the map tools menu.
Figure 17: An activity diagram for the sub-menu option slide show of map tools.
Figure 18: An activity diagram for the sub-menu options of navigation tools menu.
Figure 19: An activity diagram for the options of the user menu.
4.3 A State Diagram

The state diagrams represent the dynamic behavior of the system. State diagrams describe all possible states of the system as events occur. They show the transition of the system from one state to another, the events under which the transition occurs and the
The DNGP system has three users. The menu options available to the system depend on the level of access that each kind of user possesses. The different states of the system are shown using the state machine diagram.

Figure 21: A state diagram for the system.
4.4 A Deployment Diagram

Deployment diagrams show the hardware for the system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another. Figure 22 gives an overview of the software and hardware components of the system.

![Deployment Diagram](image)

Figure 22: A deployment diagram for the system.
CHAPTER V
IMPLEMENTATION

After completing the software design using the IBM Rational Software Modeler, the functional modules are implemented. The client-side graphical user interface is implemented by using JavaScript [5] and Cascading Style Sheets (CSS) [6] and the server-side modules are implemented by using PHP [7]. The complete source code can be found in Appendix B. This chapter describes two modules and includes some code samples. The modules are divided into two categories:

- client side modules, and
- server side modules

5.1 Client–Side Modules

The client side modules are implemented by using JavaScript and CSS. The main modules discussed here include:

- A dynamic menu: This menu is used to access the different functions provided by the system. The menu can have unlimited depth and any number of items can be added to the sub-menus.
- Dynamic user interface: The user interface of the system is dynamic which means the interface changes based on the activities performed by the users.

5.1.1 Dynamic Menu

The dynamic menu is implemented by using the following files.
1. **menu.js** contains the source code for `Menu()`.

2. **menu_create.js** contains the function `createMenu()` where the JavaScript code is entered to create the menus and menu items, customize the appearances, etc.

3. **menubar.js** contains the source code for `MenuBar()`.

4. **Stylesheet.css** contains the CSS statements that stylize the menu bar, menu bar item, etc.

The following list gives some of the functions used for creating and initializing the dynamic menu:

- **Menu(w):** Creates a new menu object.
  
The argument `w` is an integer that specifies the width of the menu.

- **addItem(menuItemObj):** Adds a new menu item to the menu.
  
The argument `menuItemObj` is a menu item object that is going to be added to the menu object.

- **menuItem(displayText, actionOnClick):** Creates a new menu item.
  
The arguments are

  1. **displayText:** String that specifies the text to be displayed on the menu item. If `displayText = "-"`, a menu separator will be created instead.
  2. **actionOnClick:** String that specifies the action to be done when the menu item is being clicked. Defaults to "" (no action).

- **setSubMenu(menuObj):** Sets the menu object that will show up when the cursor is over the menu item object.
  
The argument `menuObj` is a menu object that will show up when the cursor is over the menu item object.
• `menuBar = new MenuBar(width, height)`: Creates a new menu bar object.

  The arguments are

  1. `width`: It is optional and is an integer that specifies the width of the menu bar. Defaults to "auto".

  2. `height`: It is optional and is an integer that specifies the height of the menu bar. Defaults to "auto".

• `addMenuBarItem(menuBarItemObj)`: Adds a new menu bar item to the menu bar.

  The arguments `menuBarItemObj` is a Menu bar item object that is going to be added to the menu bar object.

• `menuBarItem(displayText, menuObj)`: Creates a new menu bar item object.

  The arguments are:

  1. `displayText`: String that specifies the text to be displayed on the menu bar item.

  2. `menuObj`: Menu object that is going to be the main menu for the menu bar item. Defaults to null (no menu).

• `moveTo(x, y)`: Moves the menu bar to the x-coordinate specified by the first argument and the y-coordinate specified by the second argument.

• `Show()`: Shows the menu bar.

• `Hide()`: Hides the menu bar.

• `initMenu()`: Checks browser compatibility and create the menus.

• `createMenu()`: This is the function where the code is entered to create the menus, menu items and customize the appearances, etc. Example code is given below. In this
code three main menus mainMenu1, mainMenu2, mainMenu3 are created using the function Menu(), and then added to the menubar created using the function MenuBar().

```javascript
function createMenu(i, w, h, userid) {
    mainMenu1 = new Menu(130);
    with (mainMenu1) {
        addMenuItem(new menuItem("Save Map", "", "code:SaveMap();"));
        addMenuItem (new menuItem("Print Map", ",
            "code:PrintMap("+w+","+h+";"));
    }
    mainMenu2 = new Menu(170);
    with (mainMenu2) {
        addMenuItem(new menuItem("Select GIS Layers", "", 
            "code:showlegend();"));
        addMenuItem(new menuItem("Select DNGP Layers", "", 
            "code:showimagelist();"));
        addMenuItem(new menuItem("Image Control", "", 
            "code:layerShow(LAYER_CONTROL_IMAGECONTROL);"));
        addMenuItem(new menuItem("Reference Map", "", 
            "code:showRefMap();"));
        addMenuItem(new menuItem("Toolbar", "", 
            "code:showNavigation();"));
    }
    mainMenu3_1 = new Menu(150);
    with (mainMenu3_1) {
        addMenuItem(new menuItem("Zoom-in", "", 
            "code:domousec1(\'zoomin\');"));
        addMenuItem(new menuItem("Zoom-out", "", 
            "code:domousec1(\'zoomout\');"));
        addMenuItem(new menuItem("Recenter", "", 
            "code:domousec1(\'pan\');"));
        addMenuItem(new menuItem("View Whole UMAC", "", 
            "code:wholeMap();"));
    }
    mainMenu3 = new Menu(170);
    with (mainMenu3) {
        addMenuItem(new menuItem("Navigation Tools","submenu", 
            ""));
        mainMenu3.items["submenu"].setSubMenu(mainMenu3_1);
        addMenuItem(new menuItem("Query", ",
            "code:domousec1(\'info\');"));
        addMenuItem(new menuItem("Slide Show", ",
            "code:layerShow(LAYER_SLIDE);"));
        addMenuItem(new menuItem("Download", ",
            "code:layerShow(LAYER_CONTROL_DOWNLOAD);"));
    }
    menuBar = new MenuBar();
    with (menuBar) {
        addMenuBarItem(new menuBarItem("Map", mainMenu1));
        addMenuBarItem(new menuBarItem("View", mainMenu2));
        addMenuBarItem(new menuBarItem("Map Tools", mainMenu3));
        addMenuBarItem(new menuBarItem("Help", mainMenu4));
    }
}
```
CSS for the Menu: CSS stands for Cascading Style Sheets. Styles define how to display HTML elements. Styles are normally stored in Style Sheets. Styles sheets define how HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2. Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document. The style sheet for the dynamic menu is stored in Stylesheet.css which defines the style properties for the menu, menubar and menu items. A sample file is shown below.

```css
/*
Menu item related selectors
*/
.jsdomenudiv {
    background-color:#E9E6B4;
    border: 1px;
    border-bottom-color: #000000;
    border-left-color: #FFFFFF;
    border-right-color: #000000;
    border-top-color: #FFFFFF;
    cursor: default;
    position: absolute;
    visibility: hidden;
    z-index: 10;
}
.jsdomenuitem {
    background-color: transparent;
    border: 1px outset;
    color: #59631E;
    font-family: Tahoma, Helvetica, sans, Arial, sans-serif;
    font-size: 12px;
    padding-bottom: 2px;
    padding-left: 22px;
    padding-right: 15px;
    padding-top: 2px;
    position: relative;
}
.jsdomenuitemover {
    background-color: #59631E;
    border: 1px outset;
    color: #E9E6B4;
    font-family: Tahoma, Helvetica, sans, Arial, sans-serif;
    /*
    Menu item related selectors
    */
    .jsdomenudiv {
        background-color:#E9E6B4;
        border: 1px;
        border-bottom-color: #000000;
        border-left-color: #FFFFFF;
        border-right-color: #000000;
        border-top-color: #FFFFFF;
        cursor: default;
        position: absolute;
        visibility: hidden;
        z-index: 10;
    }
    .jsdomenuitem {
        background-color: transparent;
        border: 1px outset;
        color: #59631E;
        font-family: Tahoma, Helvetica, sans, Arial, sans-serif;
        font-size: 12px;
        padding-bottom: 2px;
        padding-left: 22px;
        padding-right: 15px;
        padding-top: 2px;
        position: relative;
    }
    .jsdomenuitemover {
        background-color: #59631E;
        border: 1px outset;
        color: #E9E6B4;
        font-family: Tahoma, Helvetica, sans, Arial, sans-serif;
```

36
5.1.2 Dynamic User Interface

The dynamic user interface implementation includes the following JavaScript files:

1. `maptools_IE.js`: This file includes functions that are called when the user clicks on the map. They are used to handle all the mouse events that take place on the map. This file is used when the browser is Internet Explorer.

2. `maptools_NS.js`: This file has the same functions as those in `maptools_IE.js` but these functions are used when the browser is Netscape.

3. `layer_scripts.js`: This file has JavaScript functions that are used for handling the events on the layers that are created for the different windows. Some of the functions that are used are:
   - `MM_draglayer()`: This function is called when a layer is dragged by the user.
   - `layerShow()`: This is used to show the hidden layers (windows) as per the user activities.
   - `startResize()`, `resizeDrag()`, `endResize()`: These functions are used to resize the different layers (windows).

4. `other_scripts.js`: This file includes functions that are used for different purposes. Some of the functions that are used are:
• `showimagelist()`, `showRefMap()`, `showNavigation()`, `showlegend()`: These functions are used in order to show/hide the image list window, reference map, navigation toolbar, GIS layer window respectively, according to the user actions.

• `accountmenu()`, `aoimenu()`: These functions are called in order to open the account or the AOI window when the user selects the respective options from the dynamic menu.

5. `progress_bar.js`: This file includes JavaScript functions for drawing, starting and stopping the progress bar.

6. `slide.js`: This file included JavaScript functions that are used for creating a slide show of the images selected by the user from the image list.

### 5.2 Server–Side Modules

The server side modules are implemented by using PHP. The file that is called, the first time the application is loaded is `index.php`. A hidden “iframe” is declared in `index.php`. The source of this hidden “iframe” is `hide.php`. A PHP callback file is associated with every user activity. This file has callback functions that are called when the user performs different activities like logging in, logging out, selecting images, downloading images etc. The target of these PHP files is the hidden “iframe”. The file `hide.php` is included at the end of all of these files. For example, the `login.php` file is called when the user clicks the login button after entering the username and password in the login window. In `login.php`, after the user authentication is done, some flags are set and the file `hide.php` is included at the end. Now, in the file `hide.php` according to the
flags that are set, the menu options are changed and a different set of menu items is displayed based on the user permission level. In the similar way the different callback files are called and some flags are set in each file, which are then accessed in the hide.php and then the action is taken accordingly. Following is the list of the all the callback functions and the PHP files associated with them:

- **User login – login.php:**
  This file is activated when the user logs in. The username and password are verified and the permission level is set for the user. In hide.php, based on this permission level different menu items are added to the menu dynamically.

- **User logout – logout.php:**
  This file is activated when the user logs out. In hide.php, based on the permission level of the user different menu items are removed dynamically from the menu.

- **Image Selection – imglist.php:**
  This file is activated when the user selects an image from the image list. The information about the image selected is set in the session. Image flags are set at the end of the file, depending on which the functions are called by hide.php.

- **GIS Layer Selection – legend.php:**
  This is activated when the user selects a layer from the layers list. The information about the layers selected is set in the session. Image flags are set at the end of the file, depending on which the functions are called by hide.php.

- **Band Combination selection – bandcombo.php:**
This file is activated when the user selects the band combination for the original image. The band information is set in the session and the image flags are set at the end, depending on which the functions are called by hide.php

- **Color Scheme selection – scheme.php:**
  This file is activated when the user selects the color scheme when the product type is NDVI, Green NDVI, and Sugar Beet Yield. The color scheme information is set in the session and the image flags are set at the end, depending on which the functions are called by hide.php

- **Product Selection – product.php:**
  This file is activated when the user selects a product type from the image control and download window. The band, color scheme, and product type information are set in the session and the image flags are set at the end. Based on these flags, the functions are called by hide.php.

- **Create AOI – creatAOI.php:**
  This is activated when the user wants to create a new AOI. After the user has entered the AOI information, it is stored in the database. The AOI list for the user is set in the session. The AOI flags are set, depending on which the functions are called by hide.php

- **View AOI – viewAOI.php:**
  This is activated when the user wants to view AOI. After the user has selected the AOI from the list, the map extension is set in the session. The image flags and AOI flags are set, depending on which the functions are called by hide.php

- **Show Dynamic Help – help.php:**
This file is activated when the user selects the DNGP help option from the help menu. It is used to display the help according to the current context of the system. A global JavaScript variable named HELP_ID is declared in index.php and is initialized to “help_all”. The value of the variable is changed according to the different functions selected by the user. After the user selects a function from the system either a JavaScript function or a callback PHP file or both are called. The value of the variable HELP_ID is set in these files. When the user clicks the DNGP help option from the menu the value of the variable is HELP_ID is set according to the function selected by the user and it is send as a Query String to the file help.php In the file help.php the value of the variable is checked and the contents of the help file are generated dynamically depending on that value. Example code for one of the callback files, imglist.php is shown in below.

```php
<?php
    session_start();
    header("Cache-control: private");
    ob_start();
    include_once("predefine.php");
    include_once("functions.php");

    if ($_POST[IMAGE_NAME] === 'sensor') { // a new sensor is selected
        $sensor = unserialize ($_SESSION['Ssensor']);
        $sensorList = unserialize ($_SESSION['SsensorList']);
        $sensor['id'] =$_POST[SELECT_SENSOR];
        $sensor['type']=$sensorList['type'][array_search($sensor['id'],$sensorList['id'])];
        $_SESSION['Ssensor']=serialize($sensor);
        $SetExt = true;
    }
    elseif(strpos ($_POST[IMAGE_NAME], '.tif') !== false) { //an
        $ImageName = $_POST[IMAGE_NAME];
        if (!file_exists(IMAGE_PATH.$ImageName)) {
            $error_message = "Sorry, the image you chose does not exist. Please select another one."
        }
    } else { ResetImage ($ImageName);
```
This file is activated when the user selects an image from the image list. The information about the image selected is set in the session. Image flags $SetImage, $IsImageLoaded are set at the end of the file, depending on which the functions are called by hide.php.
CHAPTER VI
SOFTWARE VERIFICATION AND VALIDATION

The standard definition of verification goes like this: "Are we building the product RIGHT?" i.e. Verification is a process that makes it sure that the software product is developed the right way. The software should confirm to its predefined specifications, as the product development goes through different stages, an analysis is done to ensure that all the functions are implemented correctly and all required specifications are met. Validation is a process of finding out if the product being built is right? The software product should functionally do what it is supposed to, it should satisfy all the functional requirements set by the user. Validation is done during or at the end of the development process in order to determine whether the product satisfies specified requirements. Software testing is a very important part of the software development process. After the software is implemented, it is mandatory to test it thoroughly to uncover as many errors as possible before it is delivered to the customers. The following testing techniques and strategies have been used in this project: (i) Unit testing of Web pages, (ii) integration testing, (iii) testing for overall functionality and content delivery, and (iv) testing for usability.

6.1 Unit testing of Web pages

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. When Web applications are considered the concept of unit testing changes. Each web page encapsulates content, navigation links and processing
elements (forms, scripts). It is not always possible or practical to test each of these characteristics individually. In many cases the smallest testable unit is a Web Page. Unlike testing of the conventional software, page level testing for Web applications is driven by the content, processing, and links encapsulated by the Web page. All the callback PHP files and the different JavaScript files have been tested, to make sure that they perform the functions as stated in the specification.

6.2 Integration testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The strategy for integration testing for web applications depends on the architecture that has been chosen for the application. The architecture used for this application is three tier client-server architecture. So every time a new function was added to the system, it was tested to make sure that it worked properly and no side effects occurred. Adding a new function in our system involved adding new files which included the callback methods associated with that function. A PHP callback file is associated with every user activity. This file has callback functions that are called when the user performs different activities like logging in, logging out, selecting images, downloading images etc. The target of these PHP files is the hidden “iframe” declared in index.php. The file hide.php is included at the end of all of these files. For example, the login.php file is called when the user clicks the login button after entering the username and password in the login window. In login.php, after the user authentication is done, some flags are set and the file hide.php is included at the end. Now, in the file hide.php according to the flags that are set, the menu options are changed and a different set of menu items is displayed based on the user permission level. In the similar way the different
callback files are called and some flags are set in each file, which are then accessed in the hide.php and then the action is taken accordingly. So whenever a new function was added to the system, all the methods were tested in the callback files to make sure that it worked properly and no side effect occurred.

6.3 Testing for overall functionality and Content Delivery

Like conventional validation, the validation of the Web-based systems focuses on user-visible actions and user recognizable output from the system. The use cases that are defined in the use case diagram provide scenarios that uncover errors in user interaction requirements. The system has a dynamic user interface and a dynamic menu. There are changes in the interface depending on the activities performed by the user. The different test cases used, in order to test if the system behaved as per the user requirements, are given below:

1. User Logs in.

   Expected Results:
   
   • If login is successful display menu with options for the user according to the permission level of the user.
   
   • Hide the login button and show the logout button.
Figure 23: The menu before user logs in.
Figure 24: The interface when user clicks LOGIN.
2. User logs out.

   **Expected Results:**
   
   - Hide the menu options that are displayed for the registered user.
   
   - Hide the logout button and show the login button.

3. User selects the **Select GIS Layers** or **Select DNGP Layers** or **Reference Map** or **Toolbar** from the View menu.

   **Expected Results:**
   
   - Hide the GIS layer window if visible or vise versa.
• Hide the DNGP layer window completely if visible or vise versa.
• Hide the reference map window if visible or vise versa.
• Hide the map toolbar on the side if visible or vise versa.

Figure 26: The interface with GIS Layer, DNGP Layer and Reference Map visible.
Figure 27: The interface when GIS Layer, DNGP Layer and Reference Map are invisible.

During testing it was found out that the combo box and the image icons were still displayed even when other items of the GIS Layer window were hidden. That bug is fixed now and everything in the GIS layer window is hidden properly.

4. User selects an Image from the Image list window.

Expected Result:

- Enable the **Product type** combo box in the **Download and Image Control Window**.
• If the product type selected is Natural color, do not display the color scheme and band control options.

• If the product type selected is NDVI or Green NDVI or Sugar beet yield display the color scheme options and hide the band combination options.

• If the product type selected is Original then hide the color scheme options and display the band combination option.

Figure 28: Download and Image Control window (when product type is Original Image).
While testing it was found that the color scheme option was still displayed if “Natural color” was selected as product type after selecting “NDVI”. In one of the cases, after selecting product type as “NDVI”, the color scheme option was displayed and once one of the available color schemes was selected the color scheme option was not visible. Both of these bugs are fixed and the system works as per the specifications.

5. User clicks on the stretch icon on window and drags the mouse.
   
   Expected Result: Window should be resized when the left mouse button is pressed down on the stretch icon and moved.
Figure 30: GIS layer and DNGP images layer are resized.

It was found that this function worked properly on the Internet Explorer but was not working as expected in Mozilla Firefox. A different JavaScript function was then written and it functioned properly in both browsers. The bug is now fixed.

6.4 Testing for Usability

Usability is literally the “ease of use” or understanding it takes to make something work. In this case, Web Site usability is the understanding of how an individual user navigates finds information and interacts with the Web Site. Unlike online surveys or focus
groups, usability testing is a one-on-one process in a “watch and learn” approach. The results of the sessions are used to improve user’s experience. Having the development team watch the testing and witness the results helps resolve most internal issues in an undisputed manner.

There are two approaches for the usability testing: Formal and Informal testing.

1. Formal testing usually takes place in a formal facility with an expert in human factors engineering moderating and running the testing process. Participants are prescreened and selected from a large pool of applicants. There are viewing facilities through a two-way mirror and often complex video monitoring practices.

2. Informal testing usually takes place in the participant’s own work environment, or in a casual office setting. Participants are often friends, family or co-workers. A simple test plan and task list is prepared, which is observed and noted by an impartial moderator.

The informal approach is used in this project. During the initial development stages (while I was working on the dynamic user interface and the dynamic menu), I carried out some one-one-on usability testing sessions with the users. The feedback that I received from the users helped me a lot in designing the interface. Some of the feedbacks obtained after consulting users of the system in one-on-one sessions are:

1. Initially the Image Control and Download were separate menu items. Image Control was under the View menu and Download was under the Map tools menu. Since the users usually manipulate the images before downloading, it was really frustrating for them to go back to the menu and select the Download option, after they had already selected the Image control option. So I decided to put the Image Control and
Download options together in the same window and hide and show the different options as per the user activities. This was easier to use and more useful.

2. Initially, when the system was loaded for the first time, and if the user clicked on the map by mistake, a pop up box use to appear saying that no function was selected. The users of the system found this really annoying. After a one-on-one session with users, it was clear that zoom-in was the most commonly used Map tool function. So I decided to make it the default function that will be selected when the system starts.

3. Initially the view AOI option was in the Image Control and Download window. We already had some AOI functions like Create AOI and Delete AOI in the User menu which is displayed after the user is logged in. One of the users suggested putting it in the User menu option. Now the View AOI functions is available with the other AOI functions in the User menu.
CHAPTER VII

CONCLUSION

Digital-NGP or DNGP is an online geographic information system (GIS) for archiving, searching, and delivering remote sensing images. It is developed and sponsored by the Upper Midwest Aerospace Consortium at the University of North Dakota. This system has been developed in phases with several developers working on it. The previous version suffered from several architectural drawbacks. For example, developers usually had problems in understanding the changes made by earlier developers which in turn made the future development and maintenance difficult. Also, there was still room for improvement of system interface such as user friendliness.

This final project meets all the requirements requested by the client, Dr. Xiadong Zhang at the UMAC. The project was implemented by using software engineering principles and methodologies. The system now has a windows-like user interface, with which most users are familiar. We have a dynamic menu which is used to access the functions provided by the system. The menu can have unlimited number of items and of any depth which also makes it easier for the developers to add new functions to the system. It is now easier to manage the content, layout, and overall appearance of the system independently. The suggestions given by my client and the end users of the system have been very helpful in the development of this project. Sometimes it was really frustrating when I had to rewrite a lot of
code to meet the changes in the requirements mentioned by my client, but the regular
discussions with him helped me a lot in developing the final system.

The system is implemented by using PHP and JavaScript. In PHP files, the PHP code
to generate the dynamic HTML content is mixed with the static HTML code. If the file has a
lot of PHP code mixed with the static html code, it is difficult to debug and add new code.
The future developers may consider developing the entire system using JSP and Java Beans.
Instead of embedding HTML in programming code, JSP lets you embed special active
elements into HTML pages. These elements look similar to HTML elements but behind the
scenes they are actually componentized Java programs that the server executes when the user
requests the page. The MVC (Model-View-Controller) [8] design model can be used with
JSP, where the model corresponds to business logic and data, the view to the presentation
and the controller to the request processing. The application data structure and logic (Model)
is typically the most stable part of an application and can be implemented using Java Beans,
while the presentation of the data (the View) changes fairly often and can be implemented by
using JSP. This would make the system easy to debug and maintain.
REFERENCES

[1] Upper Midwest Aerospace Consortium


[4] IBM Rational Software Modeler
   Retrieved on June 2, 2006


[6] W3C- Cascading Style Sheets
   http://www.w3.org/Style/CSS/ Retrieved on June 2, 2006

[7] PHP

[8] Sun Developer Network

Digital-NGP or DNGP is an online GIS for archiving and delivering remote sensing images and geo-spatial information. Users can retrieve high resolution satellite imagery and interactive maps easily from our system but you may not be familiar with all the features for manipulating the images, exploring the maps, their themes and the data contained in the map layers. This document describes the features involved in our system and how you can use them to effectively retrieve the data and information from our system.

The Digital-NGP website features a modularized interface design resembling the Windows environment that most of you are used to. There are user navigation buttons and seven independent windows for map, reference map, image list, layers, and controls; each of them can be independently moved, resized and minimized.

**Map Window**

The Map window is for users to view the images, maps and corresponding legend. The right side of the window consists of interface icons which help in navigating the map, retrieving the map and getting information about specific features. The icons are black in color by default; when clicked the icon becomes active and will be highlighted in maroon color. The bottom side of the window displays the scale at which the map is displayed using scale bar and also displays the corresponding cursor position on the map in Latitude and Longitude.

**Zoom-In Icon**

This button lets users view part of the map in greater detail. You can either just click on the map at the desired location or you can hold the mouse key down to drag a box over an area of interest.

**Instructions**

1. Select the Zoom-in Icon by clicking on it or by selecting zoom-out from the Navigation Tools sub-menu option.
2. Position the cursor over the part of the map you want to see in greater detail and click.
3. Or Click on a point and drag the mouse to draw a rubber band box defining your area of interest.
4. The map will be redrawn to show 2 times more detail when clicked on the map. Or the map will be redrawn to show your area of interest when a rubber band box is drawn.
**Zoom-Out Icon**

This button lets users view a larger area with less detail.

**Instructions**

1. Select the Zoom-out Icon by clicking on it or by selecting zoom-out from the Navigation Tools sub-menu option.
2. Click on a map spot.
3. The map will be redrawn to show a greater extent centered on the mouse position.

**Full Extent Icon**

This button lets users view the entire region covered by UMAC (North Dakota, South Dakota, Montana, Wyoming, Idaho and Minnesota).

**Instructions**

1. Click the Full Extent button or select View Whole UMAC from the Navigation Tools sub-menu option.
2. The map will be redrawn showing the entire UMAC area. The difference between this button and Home button is the Full Extent button will just change the map coverage to view the entire UMAC area without making any other changes whereas Home button will reset all the settings to default and restart the website.

**Query-Map Icon**

This icon shows the attributes associated with the enabled layers in the Layer Window when a location on the map is clicked.

**Instructions**

1. Select the Query-Map Icon by clicking on it or by selecting Query Map from the Map Tools Menu.
2. Select the layers in the Layer Window on which you want to perform the query.
3. Click on any point of map.
4. The attributes of the layers at the point clicked will be displayed in a new window.
Print-Map Icon

This icon lets you print the map as it is in the map window on a paper. You can print what you see in the window.

Instructions

1. Click the Print Map button or select Print Map from the Map menu.
2. The system default Print dialog pops up and helps you through the printing process.

Save-Map Icon

This icon lets users save the map as it is in the map window as an image file (only in PNG format). You can save what you see in the window.

Instructions

1. Click the Save Map button or select Save Map from the Map menu.
2. The system default Save dialog pops up and helps you through the saving process.

Pan Icon

This icon allows users to pan (shift) the map to the desired direction.

Instructions

1. Select the Pan Icon by clicking on it or select Re-center from the Navigation Tools sub-menu option.
2. Click on any corner or side of the map you wish to pan.
3. The map will pan to the desired direction.
4. The Pan icon and the small direction icons (in yellow/orange color) around the map perform the same function of panning the map in a desired direction. The only difference is that with Pan icon you will have to select the icon first by clicking on it and then you have to click on some corner of the map to pan in that direction whereas with the direction icons, clicking on any of those direction icons will pan the map towards that direction.
This icon on click displays the Slide Control Window. The Slide Control can also be open by selecting Slide show from the Map tools menu.

**Instructions:**

1. Display the first image.
2. Click 'Add' button on the slide control window.
3. Repeat the previous 2 steps to add the rest of images.
4. Click 'Delete' to delete the last image just added.
5. Click 'Clear' to clear the animation or start a new animation.

**Layer Window**

The layer window (Figure 4) features many layers which convey information about various spatial features. Multiple layers can be overlaid on one another to make a map according to your needs and can be viewed in the Map window, or printed and saved. The layers are categorized into different groups which can be minimized or maximized by clicking on the group names. The layers DOQ, DRG and Shaded Relief are Scale-dependent; they can be viewed only at a particular zoom level. The Shaded Relief layer cannot be viewed at greater zoomed level. The DOQ and DRG layers are obtained from the Microsoft TerraServer using Web Map Service which requires you to view them at moderate zoom levels. The Layer Window can be resized, minimized or moved in the web browser according to your convenience. It can also be hidden if visible by clicking Select GIS layer from the view sub-menu option or vise-versa.

**Instructions**

1. Enable the layers which you want to display by clicking in the checkboxes next to them
2. To disable the layers, uncheck the boxes next to those layers
3. Click the Select Layers button on the top of the Layer window
4. The selected layers will be displayed in the map window

**Image Window**

The DNGP Layer window (Figure 5) contains a list of Raster images in GeoTIFF format. Different images cover different areas on the map. The image list varies according to the map extent. At any instance, the list shows images that are covering the area that is viewed in the map window. Each image is shown with a name which describes the Acquisition Date, Sensor type and Coverage information. If an image covers the current map window completely then is shown as Full otherwise it shown as Partial. The image list can be refined by selecting a particular sensor type in the Select box on the top of the window to view images only from that sensor covering the displayed area in the map window. Default Sensor type shows all the images from all the sensors. The Image Window can be resized,
minimized or moved in the web browser according to your convenience. It can also be hidden if visible by clicking Select GIS layer from the view sub-menu option or vise-versa.

Instructions:

1. Select the image you would like to view
2. Click on the Select Image button at the top of the Window
3. The image shows up in the map window

Reference map Window

The Reference Map window gives a reference of the location that is being viewed in the Map window with respect to the entire UMAC coverage by highlighting with a red colored pointer.
You can change the current location displayed in the Map window by clicking on any other location on the reference map. The Reference Map Window can be hidden if visible by clicking Reference Map from the View Menu or vise versa.

Image Control and Download Window

This window can be viewed by clicking on Image Control and Download from the View menu option. This feature allows the user to manipulate and download the chosen raster image. The buttons over the image control window will be enabled only if an image is selected and displayed in the map window. The Control Window can be resized, minimized or moved in the web browser according to your convenience.

• Band Combination

These buttons are enabled when an image is selected and displayed in the map window. If the image contains more than 3 bands, then the Red, Blue and Green bands (4,2 and 1 bands) are selected by default. You can change the band combinations depending on the number of bands the image contains to suit your needs. The corresponding bands of the image will be displayed in the map window.

• Brightness

This button is enabled when an image is selected and displayed in the map window. The brightness of the image can be increased or decreased from 0.1 to 100 times the original value. Default is 1.

• Color Scheme

This button is enabled upon selection of the specific products like NDVI, GREEN NDVI and Sugar Beet Yield. You can select the grey or color scheme. The image in the map window changes accordingly with legend integrated in the map.

• View a Product
This box is enabled when an image is selected and displayed in the map window. You can select from a range of products obtained from the image including Natural color, false color, NDVI, GREEN NDVI, Sugar beet yield. Upon selecting the required product, the image in the map window changes accordingly with the corresponding legend integrated in the map. Then you can either download the current view of the image in the map window or print or save the map.

- **Download**

This button is enabled when an image is selected in the map window. You can download either all the bands or the current view of the raster image in the map window (with selected band combinations; default is RGB) in GeoTIFF format.

**Note:**
After selecting an image, if you refresh the webpage using the Browser Refresh button then the control window buttons would be disabled. To enable them you will have to reselect the image in the image window.

**Navigation Buttons**

The Login Button takes users to the login screen, where you can enter username and password to logon to your account. Upon success the login Button changes to Logout button.

**Instructions**

1. If you don't have an account with Digital-NGP then you need to create an account by clicking the Register Button.
2. If you already have an account with us, Click the Login Button.
3. A new window pops up and lets you enter the username and password.
4. Upon successful logon, you can use the DNGP AOI navigation button to manage your AOI and DNGP ACCOUNT navigation button to manage your ACCOUNT information.
5. The DNGP LOGIN button changes to DNGP LOGOUT. You can successfully end your session by clicking the LOGOUT button or by closing the web browser or by refreshing the page.
6. If you forgot your password, then you can follow the FORGOT PASSWORD? link on the login interface where you will need to answer your Hint Question created at the time of Registration. If you answer correctly, then we will let
you create a new password for your account. If you have any problem with the Login, please contact us through the DNGP QUESTION navigation button and we would be very happy to help you promptly.

**NOTE: Once you are logged in to our system and if use the web browser's Refresh button or the DNGP Home button then you will be required to Login again.**

The Register Button takes user to the registration screen.

**Instructions**

1. Click on the Register Button
2. A new window pops up where you can Enter the registration details and click Register
3. The rows marked * are required and have to be provided before the process is complete.

The Home button returns to the home page.

**Instructions**

1. Click the Home Button
2. The Digital-NGP website restarts with the default settings.

Only after logging into the Digital-NGP, this option appears in the User menu. The sub menu options available are create AOI, Delete AOI, View AOI, Request AeroCam, Request Agcam

**Create AOI**

1. Click on Create AOI link menu option.
2. Enter the name of AOI. Enter the values in DMS format and should be either in clock-wise/anti-clockwise format.
3. To enter longitude values you can either enter the values with the hemisphere, for eg., 106 25 34.56 W with a space between degree, minute, seconds and direction. For Latitude the value is 44 29 68.09 N.
4. Alternatively, the values can also be entered with a negative sign in front of it without the hemisphere. For eg., -106 25 34.56 with a space between the degree, minute and seconds but NO space between the '-' sign and the degree value
5. AOI will be created which can be accessed from the View AOI menu in the Control Panel

Create AOI

1. Click on Create AOI with Map link.
2. Zoom in to the desired Area of Interest.
3. Enter the AOI Name and click Create AOI Button.

Delete AOI

1. Click on Delete AOI link.
2. Select the AOI Names you want to delete and click Delete AOI.
3. AOI values deleted are removed permanently from the database and are irrecoverable.

View AOI

1. Click View AOI menu option.
2. Select AOI from the drop down menu.
3. Selected AOI is displayed on the map.

Request AeroCam or AgCam

Users can send request for AeroCam and AgCam image acquisition. Must be a registered user to access. Once login, under the AOI menu, there are two options, one for the AeroCam and the other for the AgCam. Follow the instruction then.

Account

Only after logging into the Digital-NGP, this button appears. This button allows you to manage your account.

Instructions

1. Click the Account from the User sub-menu option
2. A drop down menu appears with the Edit my account, change my password and change my secret question entries
3. Make the selection and click on it to go to desired function

Edit my account

1. Click on Edit my account link.
2. A new window pops up showing your current account information. You can change the information in the window

**Change my password**

1. Click on change my password link.
2. A new window pops up to let you create a new password but it requires you to enter your current password

**Change my secret question**

1. Click on change my secret question link.
2. A new window pops up to let you change your secret question and answer which would be required if you forget your password
APPENDIX II

SOURCE CODE

Please contact Dr. Xiaodong Zhang (zhang@aero.und.edu) at the Upper Midwest Aerospace Consortium (UMAC) for the source code of this project.