

DEVELOPMENT
OF PROCESSES FOR ONLINE
GIS DATA SELECTION, EXTRACTION,
AND AGGREGATION USING ARCIMS AND .NET
TECHNOLOGY

By

VARUN CHUDIWALE

Bachelor of Engineering

B.D.C.O.E, Sewagram

India

1999

Submitted to the faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirement for
the Degree of
MASTER OF SCIENCE
December, 2005

DEVELOPMENT
OF PROCESSES FOR ONLINE
GIS DATA SELECTION, EXTRACTION,
AND AGGREGATION USING ARCIMS AND .NET
TECHNOLOGY

Thesis Approved:

Dr. Johnson P. Thomas

Thesis Advisor

Dr. John Chandler

Dr. Allen Finchum

Dr. A. Gordon Emslie

Dean of the Graduate College

Acknowledgement

The purpose of this thesis is to provide different ways of GIS data extraction over the internet using ArcIMS and .Net technologies. The process of completing this thesis involves a significant amount of planning, design, drafting and revision, not to mention timing, luck and perseverance.

I would like to thank Oklahoma State University and the Department of Computer Science. I wish to express my appreciation to Dr. Johnson Thomas for his expert guidance and advice. In addition, I want to thank the Department of Geography for providing the computing resources for the development of this thesis. In particular, I wish to express my grateful thanks to Dr. Allen Finchum for his time, support, and for guiding me in the right direction.

Most importantly, I would like to thank my parents and my brother Ashutosh and Ashish and my friends for always being there for me throughout this exciting, sometimes stressful, long and gratifying journey. Your support and understanding have led to this becoming a reality.

TABLE OF CONTENTS

<u>TITLE</u>	<u>PAGE</u>
<u>1. INTRODUCTION</u>	<u>1</u>
<u>2. GIS CONCEPTS AND REQUIREMENTS</u>	<u>6</u>
2.1 GEOGRAPHIC DATA MODEL	6
2.2 GIS SOFTWARE	8
2.2.1. DESKTOP GIS	9
2.2.2. INTERNET GIS	9
<u>3. PROBLEM STATEMENT AND AIMS OF RESEARCH</u>	<u>12</u>
3.1 PROBLEM STATEMENT	12
3.2 AIMS OF THE RESEARCH	13
<u>4. BACKGROUND AND LITERATURE REVIEW</u>	<u>17</u>
4.1 LITERATURE REVIEW	17
4.2 INTRODUCTION TO CREATION OF CUSTOMIZED ARCIMS WEBSITE	19
4.3 ARCIMS ARCHITECTURE	21
4.2.1 ARCIMS SPATIAL SERVER	21
4.3 ARCSDE	22
4.4 ARCXML	23
4.5 C#.NET	24
<u>5 METHODOLOGIES AND IMPLEMENTATIONS</u>	<u>26</u>
5.1 METHODOLOGY	26
5.2 IMPLEMENTATION	28
5.2.1 IMPLEMENTING EXTRACT FUNCTION FOR DOWNLOADING RASTER AND VECTOR DATA MODEL	29
5.2.2 IMPLEMENTING ‘TRACING ROUTE’	34
5.3 CHALLENGES	39
<u>6 RESULTS AND CONCLUSIONS</u>	<u>40</u>

<u>ABBREVIATIONS</u>	<u>42</u>
----------------------	-----------

<u>REFERENCES</u>	<u>43</u>
-------------------	-----------

<u>CODE</u>	<u>54</u>
-------------	-----------

<u>LIST OF FIGURES AND TABLE</u>	<u>PAGE</u>
<u>FIGURE 2.1 STRUCTURE DIAGRAM OF ARCIMS, ARCSDE AND ARCXML</u>	<u>11</u>
<u>FIGURE 3.1 BUFFERED ROUTE WITH SELECTED BLOCK GROUP USING ARCVIEW 3.2</u>	<u>14</u>
<u>FIGURE 3.2 BUFFERED ROUTE WITH SELECTED BLOCK GROUP USING ARCGIS 8.3</u>	<u>15</u>
<u>FIGURE 3.3 DISPLAY THE CREATED BUFFER WITH RESULTANT DATA FROM DIFFERENT TABLES OF SPATIAL AND NON SPATIAL DATA BASE</u>	<u>16</u>
<u>FIGURE 4.1 DEFAULT TOOLBAR CREATED BY ARCIMS WEBSITE</u>	<u>20</u>
<u>FIGURE 4.2 THREE TIER ARCHITECTURE OF ARCIMS</u>	<u>21</u>
<u>FIGURE 5.1 FRAMESET OF CUSTOMIZED ARCIMS WEBSITE</u>	<u>26</u>
<u>FIGURE 5.2 FLOW DIAGRAM OF REQUEST AND RESPONSE FROM ARCIMS EXTRACT SERVER</u>	<u>29</u>
<u>FIGURE 5.3 FLOW DIAGRAM OF EXTRACTION OF RASTER DATASET ALONG WITH VECTOR DATA</u>	<u>30</u>
<u>FIGURE 5.4 WORK FLOW DESIGN FOR THE IMPLEMENTATION OF THE DOWNLOAD OF RASTER DATA FUNCTION</u>	<u>31</u>
<u>FIGURE 5.5 SKETCH THE ROUTE AND DISPLAY BUFFER AROUND THE ROUTE</u>	<u>32</u>
<u>FIGURE 5.6 ATTRIBUTE DATA THAT INTERSECTS THE BUFFERED REGION IN THE ARCXML FORMAT</u>	<u>33</u>
<u>FIGURE 5.7 BIND SPATIAL AND NON SPATIAL DATA TO THE DATAGRID CONTROL</u>	<u>34</u>
<u>FIGURE 5.8 WORK FLOW DESIGN FOR THE IMPLEMENTATION OF ‘TRACE ROUTE’ FUNCTION</u>	<u>36</u>
<u>FIGURE 6.1 CUSTOMIZATION OF THE ARCIMS WEBSITE FOR ‘TRACE ROUTE’ FUNCTION</u>	<u>42</u>

<u>FIGURE 6.2 ZOOMING IN THE DESIRED AREA AND THE INPUT BOX FOR THE BUFFER WIDTH SPECIFICATION</u>	<u>43</u>
<u>FIGURE 6.3 TRACED ROUTE WITH THE BUFFERED REGION</u>	<u>44</u>
<u>FIGURE 6.4 RESULT FROM THE BLOCK GROUP LAYER'S REGION FROM THE BUFFERED REGION</u>	<u>45</u>
<u>FIGURE 6.5 RESULT FROM THE SPATIAL DATABASE IN THE DATAGRID FORMAT</u>	<u>46</u>
<u>FIGURE 6.6 MAP SHOWING QUADRANGLES OF DESIRED REGION</u>	<u>47</u>
<u>FIGURE 6.7 READY TO DOWNLOAD - RASTER IMAGE WITH VECTOR DATA OF DESIRED AREA</u>	<u>48</u>
<u>FIGURE 6.8 MAP WITH A LINK TO DOWNLOAD RASTER AND VECTOR DATA TOGETHER</u>	<u>49</u>
<u>TABLE 4.1 DIFFERENT TASKS WITH THE IMPLEMENTATION TOOLS</u>	<u>25</u>

Chapter 1

Introduction

Geographical Information Systems (GIS) is a computerized data management system designed to capture, store, retrieve, analyze and report geographic information [Opadeyi, 1995]. The information in a GIS relates to the characteristics of geographic locations or areas. In other words, a GIS answers questions about where things are located or about what is located at a given location on the earth surface. GIS uses the concept of geographical mapping that can be programmed using simple code allowing for future modification when necessary. GIS technology enhances the understanding and evaluation of geographic data by creating graphical outputs in the form of maps. GIS information is represented by a series of geographic datasets that models geography by a simple, generic data structure. This data structure incorporates various graphical features with tabular data also called attribute data in order to understand and analyze real-world problems. Geographic datasets are use to create sets of intelligent maps. This maps show features and feature relationship on the earth's surface using x,y coordinates. In developing a GIS application, real world features need to be translated into simplified representations that can be stored and manipulated in a computer. Different types of dataset models available to create a GIS application [Troelson, 2001]. The two most commonly used data models

for internal representations of geographic information are the vector data model and the raster data model. The vector data model is used to symbolize discrete features such as houses, roads or districts, and the raster data model, which although is most often used to represent continuously varying phenomena such as elevation or climate, is also used to store pictures or image data from satellites and plane based cameras.

Data is the core element of all successful geographic information systems and is the single largest investment in any GIS system. The simultaneous collection of raster and vector data models has always been a challenging task for GIS professionals. Even though there exist many well established facilitation techniques, numerous challenges continue to confront those attempting to manage the process and make these data models available together in one click on the internet in a user friendly way. The manipulation of both raster and vector data over the internet is a big problem because the available technology supports downloading vector data type only. There is no supporting technology that allows users to download both raster and vector data type. The selection and extraction of geographical data in a user friendly environment is difficult because the available technology only supports extraction of geographical data using the standard file transfer protocol (ftp) which means a user cannot visualize data in a geographical map format.

This thesis highlights two unique functionalities created by combining GIS and Microsoft .Net technology. The first functionality, named 'Extract Data', explores the collection and manipulation of vector and raster data together over the internet. The Extract Data function helps to download two GIS data models, namely vector and raster, together in a compressed zip file format on a user's hard drive over the internet. This is

important because the downloaded file is available in one compressed zip format. The zip file contains all the necessary files that can be decompressed to view these data models as a geographic map on a computer. Furthermore, we provide new mechanisms for the manipulation of vector and raster data together over the Internet. These manipulations include viewing raster and vector data in one map projection. A map projection is a system in which locations on the curved surface of the earth are displayed on a flat sheet.

In this thesis we also introduce a second functionality, named 'Trace Route'. The Trace route function helps user to select some region and trace a route on a map on a computer screen by mouse clicking, for extracting the data relevant to the traced route.

Our proposed approaches are novel because both these functionalities are currently not provided by GIS systems. Existing GIS tools only provide downloading vector data model over the internet. None of the available GIS tools support extracting both raster and vector data together. In order to download both the GIS data models in the computers hard drive vector data has to be downloaded first from one http website and the raster data from another ftp website. With this approach the user cannot therefore get a WYSIWYG (what you see is what you get) environment.

Our proposed functionalities are of particular benefit to the construction and transportation industries. For the people in these industries, it has always been a difficult task to "draw" or simulate a new or existing highway route in order to gather socioeconomic and demographic data related to the traced route. Different applications and methodology have been developed to access such functionality. But all of them are available only for stand alone desktops. In other words , in order to use this function, the user needs to have the required GIS software installed in their desktop. The 'Trace Route'

functionality brings such application over the internet in a fast and user friendly environment.

Both the 'Extract Data' and 'Trace Route' applications were designed and developed to test the above mentioned concepts over the internet and to demonstrate the ability to implement such tools by integrating GIS and Microsoft .Net technologies. Running this application on the internet required setting up as Internet Map Server - Microsoft IIS, GIS Web publishing application - ArcIMS, server side programming language - C#.Net and database- Microsoft SQL Server together. The map of Oklahoma State was used as a case study for the demonstration of this application. Screenshots of the demonstrated site are available in Appendix A.

In chapter 2, we introduce the technical terms of GIS software that are used in this thesis. The chapter discusses the different GIS data formats and GIS software used in this thesis.

In Chapter 3, the problem statement and aims of the thesis are stated

Chapter 4 highlights the background and literature review. This chapter focuses on previous work done related to 'Extract Data' and 'Trace Route' functions. This chapter also defines the architecture of the GIS web mapping software called ArcIMS and its supportive software called ArcSDE (spatial database engine) and ArcXML (XML for GIS software). Later this chapter discusses the advantages of Microsoft .Net technology and its integration with GIS software.

Chapter 5 describes the methodology for implementing 'Extract Data' and 'Trace Route' functions. This chapter also highlights the different challenges faced during the implementation process of this tool.

Finally chapter 6 provides the results and conclusion of the work done in this thesis.

Chapter 2

GIS Concepts and Requirements

A geographic information system (GIS) is a system for the management, analysis, and display of geographic information. Geographic information is represented by a series of geographic datasets that model geography using simple, generic data structures.

2.1 Geographic data model:

A GIS database is a computer-based representation of the real world. GIS software provides the tools for organizing information about spatially defined features. Spatially defined features are somehow related to the surface of the earth using x,y coordinates. The basic organization principle of a GIS is the data layer. Rather than storing all spatial features in one place, as on a paper map, groups of similar features are combined in one of a number of these data layers. A comprehensive GIS database will include layers of physical features such as roads, rivers and buildings, as well as layers of defined features such as administrative boundaries or postal zones which cannot be observed on the ground [Web 03]. Other layers may include a climate layer, soil layer etc. as required. These data layers can be represented using vector and raster data models. Each representation has its own methods of presenting and using maps and the data associated with the maps.

Vector data: The Vector data model represents real-world features using a set of geometric primitive points, lines and polygons. A point is represented in a computer database by an x,y coordinate called a point feature. Points may be connected to form lines called line feature. A line feature is a sequence x,y coordinates, whereby the end points are usually called nodes and the intermediate points are termed vertices. Polygons are represented by a closed series of lines such that the first point equals the last point of the loop. Points might be used to represent houses, wells or geodetic control points; lines describe such features as roads and rivers; and enumeration areas or districts, for example, are represented by polygons [Figure 1.2]. The point, line and polygon features of vector data are stored in a special structure file format that can be readable by different GIS applications; this special structure file format is known as a ShapeFile. A ShapeFile stores the location, shape, and attribute information of geographic features.. Where shape can be either a point, line or polygon. A ShapeFile stores the geometry and attribute information for the spatial feature in a data set. This dataset is a combination of 3 other files, the other files are called main file, index file, and a dBASE file. The main file is a direct access, variable-record-length file in which each record describes a shape (point, line, and polygon) with a list of its vertices. In the index file, each record contains the offset of the corresponding main file record from the beginning of the main file. The dBASE table contains feature attributes with one record per feature. bBASE file is similar to database file which is indexed with other files in a ShapeFile.

Raster Data: The Raster data model is more like a photograph than a map. In fact when a

picture is stored in a computer the raster data model is used. In this model a grid of cells is overlaid on a map or picture. On a computer display this is a matrix/grid of pixels. Each cell's location is identified and has a value. The most common types of raster data include satellite images and scanned aerial photographs. Different types of raster data available and used in this research include Digital Raster Graph (**DRG**), Digital Ortho Quads (**DOQ**). A DRG is a scanned image of a United State Geological Survey (USGS) standard series map. The image inside the map neat line is geo referenced to the surface of the earth. DOQ is a digital image of an aerial photograph in which displacements caused by the camera and the terrain have been removed. It combines the image characteristics of a photograph with the geometric qualities of a map. The standard digital ortho photo produced by the USGS is a black-and-white, or color infrared images. The DOQ's and DRG's are photographs taken from different sources. To geo-reference these photographs on the earth surface 'World Files' are used. **World Files** contains the geographical information, which is basically the X, Y coordinates of DOQ and DRG. A world file is a small text file that accompanies the image file. For example, an image called **Oklahoma.jpg** that is an ordinary graphics file in JPG format might be accompanied by a small file called **Oklahoma.jpgw**, which is the world file. Without the world files DOQ's and DRG's are similar to scanned photographs.

2.2 GIS Software:

A modern GIS software system comprises an integrated suite of software components, including end user applications, geographic tools and data access components. The Environmental Science Research Institute (ESRI) is the leading developer of geographic

information systems software. ESRI's GIS software packages could be classified into two groups based on the functionality and type.

2.2.1. Desktop GIS

A Desktop GIS focus on data use, rather than data creation, and provides excellent tools for making maps, reports, and charts. A well-know example is ESRI's ArcGIS desktop.

ArcGIS desktop is used to perform many GIS tasks like mapping, geographic analysis, data edition and compilation on a desktop environment. Users can also develop their own extensions to ArcGIS desktop by working with ArcObjects, the ArcGIS software component library. Users develop extension and custom tools using standard window programming interfaces such as Microsoft's VBA (Visual Basic Application), Java, C++. These programming Interfaces are used to create stand alone executables or components that can plug into existing GIS application.

2.2.2. Internet GIS

Internet GIS focuses on display and query applications, as well as mapping. Examples include ESRI's ArcIMS (Internet Mapping Server).

ArcIMS, an internet mapping server, is used for delivering dynamic maps and GIS data over the Web. Its primary focus is Web delivery of geographic data and maps. It provides a highly scalable framework for GIS Web publishing that meets the needs of corporate Intranets and the demands of worldwide Internet access. Typically, ArcIMS user's access GIS web services through their Web browsers using HTML, Java, or .Net applications that are included with ArcIMS.

ArcSDE, a spatial database engine, is the GIS gateway to relational databases for ArcIMS. It allows users to manage geographic information in any of several DBMS and serve their data openly to all ArcIMS applications. When it needed a large multi-user geodatabase that can be used simultaneously by many users, ArcSDE adds the necessary capabilities to GIS system by a DBMS.

ArcXML is ESRI's flavor of XML and stands for Arc eXtensible Markup Language. ArcXML is the language used for sending requests and receiving responses through ArcIMS' spatial server. The openly published XML language for ArcIMS is named ArcXML. ArcIMS uses XML for its communication and interactions. It provides access to all ArcIMS functions and capabilities. All client requests and server responses in ArcIMS are coded in ArcXML. ArcXML support series of connectors that enables Web development to use standard tools, including C#.Net for Microsoft application, Java Server Pages (JSP) to build Web application using J2EE. For this thesis C#.Net environment is used as Web development tool.

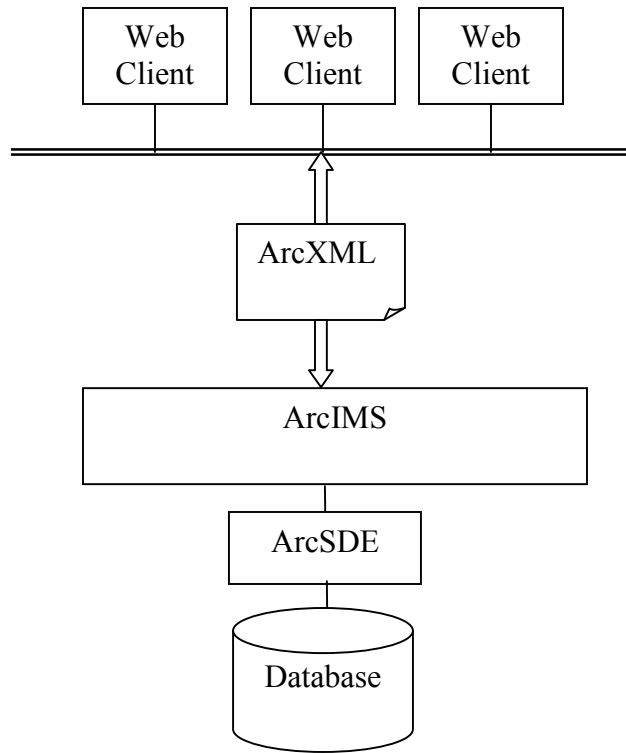


Figure 2.1 Structure diagram of ArcIMS, ArcSDE and ArcXML.

Chapter 3

Problem Statement and Aims of Research

3.1 Problem Statement:

ArcIMS allow users to extract and export geographical data into ShapeFiles format. The extraction is currently done at the vector data level only. Any vector data fully or partially visible in the view extent of the web browser is included in the download. However, available technologies do not support extracting raster data. For downloading both the GIS data model into the user's hard drive, vector data needs to be extracted from ArcIMS website and raster data from ftp web site.

The drawbacks of this process are,

1. A large Number of links have to be followed to download raster data.
2. Only one data type, either raster or vector, can be downloaded at one time.
3. The User cannot get a 'what you see is what you get' environment;

No available technology supports downloading vector and raster data together online. It is therefore essential to create a functionality that allows users to download both vector and raster data online in faster and user friendly way.

Secondly, there is no available technology that facilitates tracing routes on the computer screen over a map and simultaneously views business data relevant to the traced route.

Similar functionality is developed using ArcGIS desktop [Figure 3.1 and 3.2], that allows user to define a new highway route on the map on the computer screen and extract data for the desired area. As this is created using ArcGIS desktop, it can be only used as a stand alone desktop application. To use this function, the user needs to have ArcGIS installed and this is available only for a single user. It is therefore necessary to make this stand alone desktop application available over the internet and make it available for multi users.

3.2 Aims of the Research:

The over all vision of this research is to provide unique ways of collection of different GIS data models. The two novel functionalities that have been achieved in this research make it easier for the user to extract different GIS datasets over the internet. The extracted data also includes world file. The world file is used to display the location on the curved surface of the earth on a flat sheet over the computer screen.

The specific aim of the first functionality of the research is to provide GIS professionals, not only the ability to extract vector data, but also raster data along with world file in an online user friendly environment. The world file makes this data ready to use in ArcGIS, ArcView, or any other GIS software that supports these file types. The second function develops a method that allows user to trace a route on the computer screen over a map by mouse clicking and download the data relevant to that route. From an academic viewpoint, a further aim of this research is also to offer a better understanding of integration of Modern .Net and GIS technologies.

ArcIMS uses XML for its communication and interactions. .Net development

environment provides strong support for XML throughout the .NET Framework. The .NET XML framework provides support for all of the W3C-stamped XML specifications including XML schema, XSLT, Namespaces in XML. The entire .NET Framework is available to any ASP.NET application. This research aims to develop reusable COM components created using .Net framework for functions like download raster data and trace route.

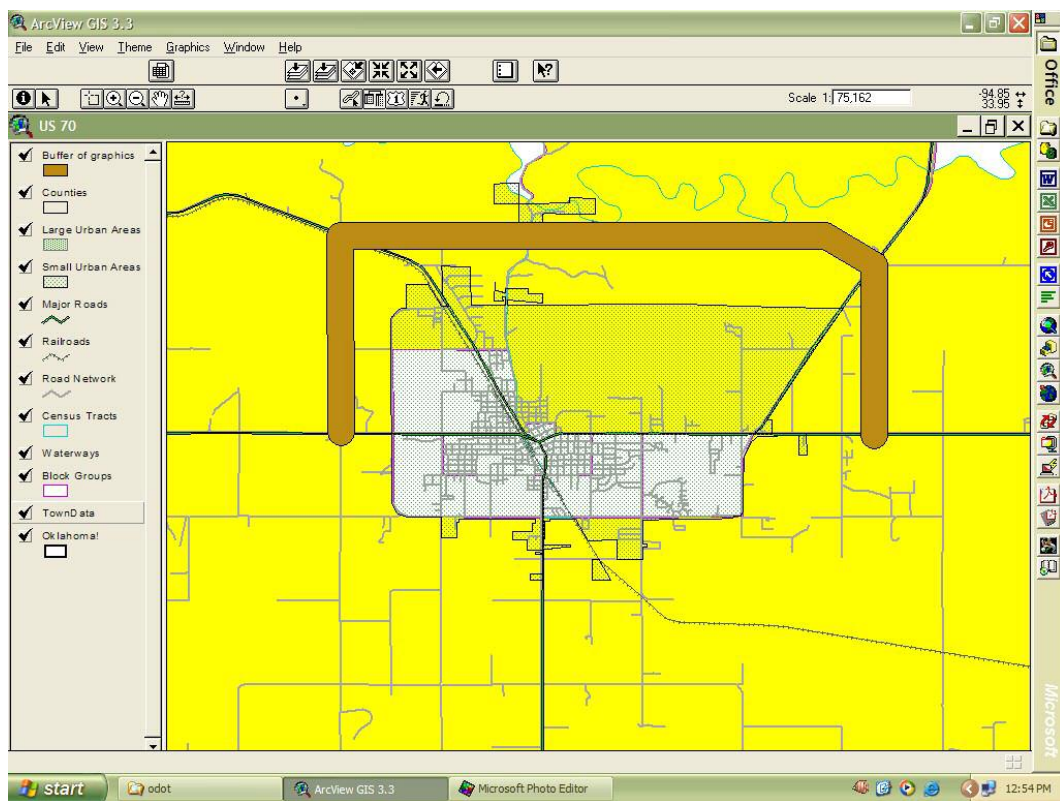


Figure 3.1 Buffered route with selected block group using ArcView 3.2

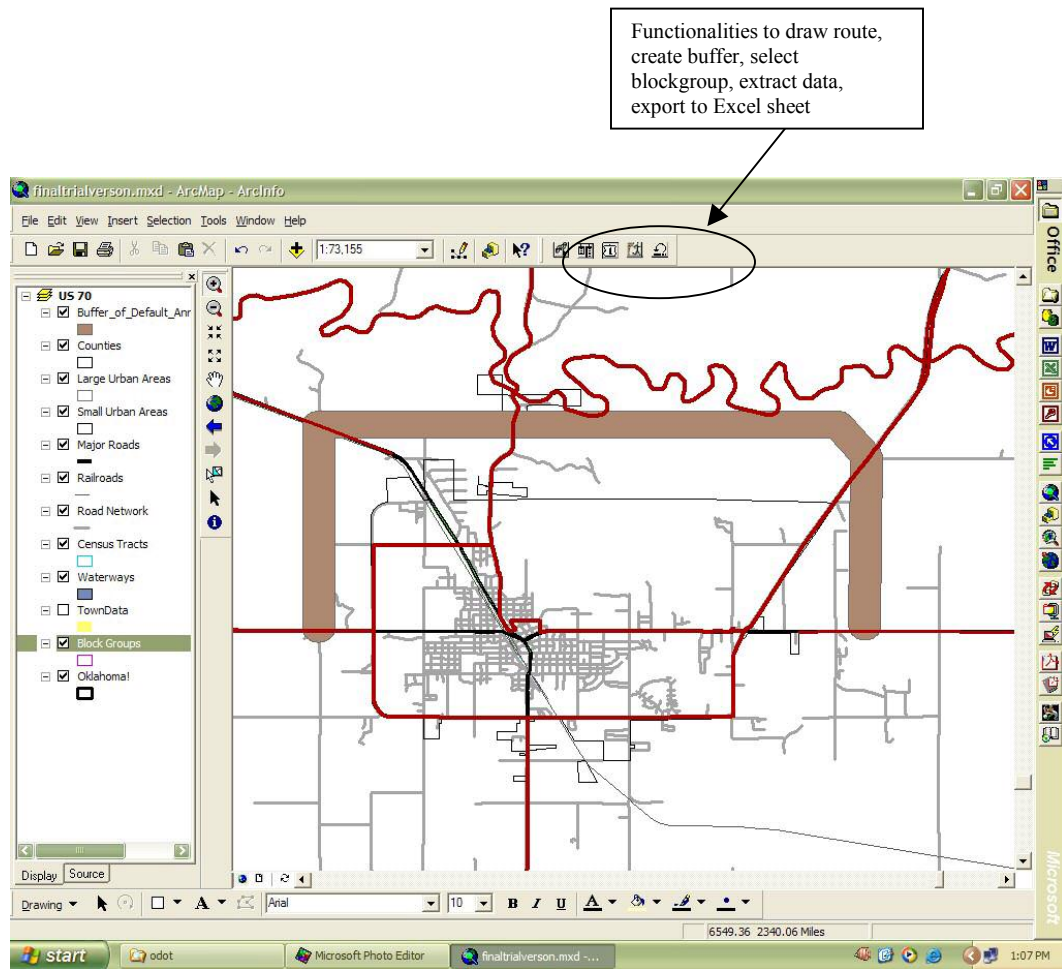
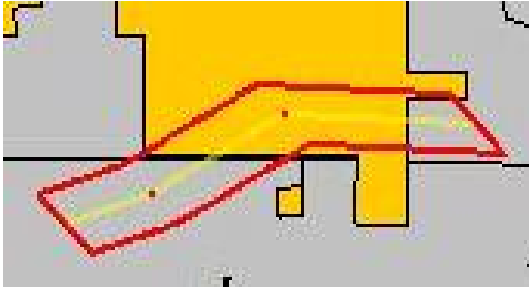


Figure 3.2 Buffered route with selected block group using ArcGIS 8.3



- ✓ Extract town data covered under buffer
- ✓ Extract block group data intersect the selected town



Spatial Data	Non Spatial Data
Town data	SIC data
Block Group Data	
County data	NASIC data
Other data	

Figure 3.3 Display the created buffer with resultant data from different tables of spatial and non spatial data base.

Chapter 4

Background and Literature review

This chapter primarily focuses on the background work on web GIS applications and further describes the usability of common components under web based GIS. The review helps define the variables and provides the framework in proceeding further (Sekaran, 2000).

‘ESRI’s Koeppel (Koeppel 2001) states that ‘ArcIMS revolutionizes the way user's access and interact with Internet mapping’. It is the only software that facilitates users to join together local data sources with data sources on the internet for display, query, and analysis using a web browser [Knowles, 2002]. ArcSDE provides a gateway to relational database for multiuse and distributed GIS systems, enabling ArcIMS to work directly with spatial data managed in RDBMS. Our literature review shows that before ESRI announced ArcIMS, there was a lack of technology to produce maps online.

4.1 Literature Review:

As far as we are aware there is no functionality provided by ESRI in ArcIMS to download Raster data. Although there are sources that let raster data to be downloaded, none provide the raster data satisfying the WYSWYG property. The only way to extract

Raster dataset is using file transfer protocol (ftp). The GIS data is located in the ftp server and link of the server is provided to the user. Download GIS datasets using ftp is similar to download any other nonGIS file from ftp server. It does not involve map viewing facility. User can just see the file names and has to guess the actual map just by the file name of the GIS data. After the release of ArcIMS, the extraction of GIS dataset is possible in WYSWYG environment. Where user can actually see the map and request the ArcIMS server to download the data in proper format. But ArcIMS doesn't solve the problem of downloading the entire GIS datasets. ArcIMS is only capable of downloading Vector format of GIS dataset. None of the technology is available today to download Vector and Raster dataset together over the internet.

Although a wide range of research has been done to implement the 'Trace Route' function discussed earlier in this thesis, all the research has focused on implementing this function to run on standalone applications. No work has been reported on implementing 'Trace Route' over the internet. ESRI's ArcView 3.2 and ArcGIS 8 have been used to implement standalone 'Trace Route' desktop application. This application was initially implemented using ArcView 3.2, the previous version of ArcGIS 8.0. Figure 3.1 shows the implementation of 'Trace Route' function using ArcView 3.2. Later the same application was enhanced using the ArcGIS 8.0 environment, as depicted in Figure 3.2. This enhancement in the ArcGIS 8.0 environment facilitated further customization and better performance, as ArcGIS uses Visual Basic Application (VBA) for ArcGIS and COM with ArcObjects as the scripting platform, rather than the proprietary Avenue language used in ArcView 3.2. But both the applications designed were desktop standalone application. The next challenge is to implement and make this application available

over the internet. This could be put into action using the finer features of an enhanced application ArcIMS, later launched by ESRI. Though the launch of ArcIMS simplified the implementation of several functionalities, it could not support the basic requirements for the 'Tracing Route' application

4.2 Introduction to creation of customized ArcIMS Website:

ArcIMS establishes a common platform for the exchange of Web-enabled GIS data and services. ArcIMS is more than just an Internet mapping solution—it is a framework for distributing GIS capabilities over the Internet. ArcIMS provides four customizable clients—HTML, Java, ActiveX, and Cold Fusion. The default website created by ArcIMS is known as HTML viewer website. This research is based on HTML viewer, because HTML is the most widely accepted and supported language on the Web for defining page contents. It does not require the Java 2 plug-in or Applet support.

The ArcIMS default website comes with simple and user friendly features like ZoomIn, ZoomOut, Pan, etc. as shown in Figure 4.1 This default ArcIMS website is built with a plain appearance and simple functionalities, which can further be customized by including additional required features like trace route and a download data model.

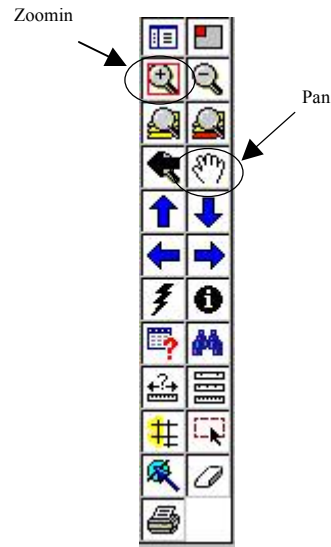


Figure 4.1 Default toolbar created by ArcIMS website

The creation of a basic default ArcIMS website requires the following steps,

- Create .axl file to generate map configuration files. These files are written in ArcXML and hosted as ArcIMS web services in the ArcIMS server. ArcIMS Author tool, which comes with the ArcIMS package, is used to create this map configuration file. This .axl file holds parameters to access GIS data models Map configuration files can also be created and edited using an XML editor.
- A second management task is to publish and administer this configuration file using ArcIMS Administrator. ArcIMS Administrator is another tool that comes with the ArcIMS package. The published file in ArcIMS Administrator is called as MapServices. For accessing this MapServices request in ArcXML, it needs to be sent to the ArcIMS server.
- The last task is to create the ArcIMS website and access the MapServices ArcIMS Designer tools, the third tool that comes with the ArcIMS package, is used to create the default ArcIMS website.

4.3 ArcIMS Architecture:

ArcIMS has a multitier architecture consisting of presentation, business logic, and data tiers. In addition, ArcIMS has a set of applications for managing a Web mapping site.

- The presentation tier includes the ArcIMS client viewers for accessing, viewing, and analyzing geographic data.
- The components in the business logic tier are used for handling requests and administering the ArcIMS site.
- The data tier includes all GIS data sources available for use with ArcIMS.

The ArcIMS site management applications provide access to components in the business logic tier for authoring maps, administering ArcIMS services and Spatial Servers, and designing Web sites. [ESRI 04]

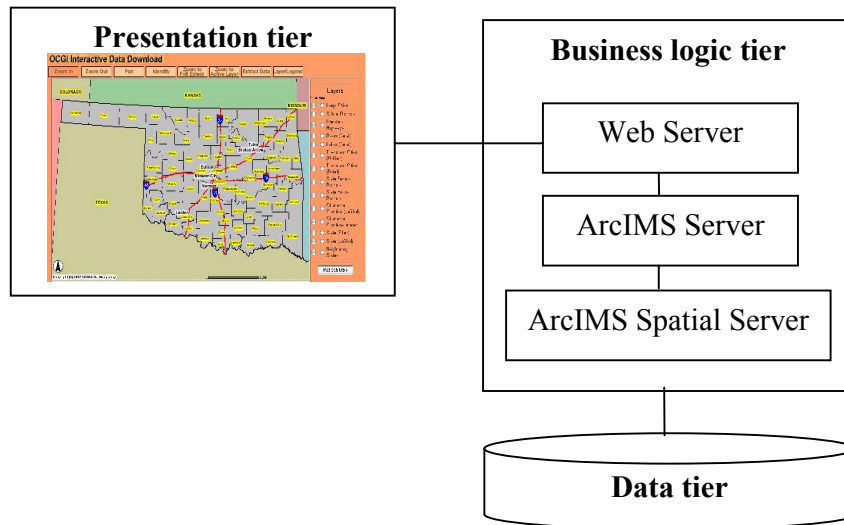


Figure 4.2 Three tier architecture of ArcIMS.

4.2.1 ArcIMS Spatial Server:

The ArcIMS Spatial Server processes requests for working on maps and related information [ArcIMS, 2001]. There are many variations of ArcIMS Spatial server but our discussion focuses only on the two basic spatial servers the image server and the extract server.

- **Image Server:** Generates and send maps to Web browsers as JPEG, PNG, or GIF images
- **Extract Server:** Extracting or "clipping" data to create a subset that can be sent back in ShapeFile format. The standard ESRI based system permits the service provider to allow the user to download customized shape files of geospatial datasets for vector and GRID based data [FINCHUM, 2004]. When using an extract server a zip file of all the visible layers shape files is placed in one default directory called output directory. The dBASE table in .dbf file in ShapeFile contains feature attributes with one record per feature.

4.3 ArcSDE:

ArcSDE, the spatial database engines software's primary role is to provide gateway access in a multi-user GIS environment to spatial data stored in a database [Longley, 2001]. Using ArcSDE, ArcIMS can work directly with spatial data managed in a database. ArcSDE provides the infrastructure required to manage multiple users editing the same spatial database with long transactions. This also includes editing of spatial database over the internet through ArcSDE where as the Non-spatial data in the SQL Server can displayed or modified directly with out the ArcSDE interface. It is easy to convert both raster and vector data into a geodatabase. File based data, such as shape files

and raster data files are difficult to manage as they are comparatively slow on internet when accessed by multi-users. ArcIMS and ArcSDE work together as an integrated back office solution for fast Internet access to vector, raster, and survey data stored in a relational database [ESRI 04].

4.4 ArcXML:

ArcXML is a GIS extension to standard extensible markup language (XML). It is the protocol for communicating with the ArcIMS Spatial Server. An ArcIMS Spatial Server is the backbone of ArcIMS and provides the functional capabilities for accessing and bundling maps and data into the appropriate format before sending the data back to a client. ArcIMS clients and servers communicate using ArcXML.

As an example, the following is a request sent to the Image Server of ArcIMS spatial server to generate a map using ArcXML.

Request to the ArcIMS server

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<ARCXML version = "1.1">
  <REQUEST>
    <GET_IMAGE>
      <PROPERTIES>
        <ENVELOPE minx = "-125" miny = "25"
          maxx = "-67" maxy = "50" />
      </PROPERTIES>
    </GET_IMAGE>
```

</REQUEST>

</ARXML>

This request is generated by ArcIMS HTML Viewer using Java script and is sent using java connector. ArcIMS Spatial Server processes a request and responds to the request in ArcXML format program in response. The map that comes on the browser while accessing ArcIMS is actually the image in jpg format

Response from the ArcIMS server

<RESPONSE>

<IMAGE>

<ENVELOPE minx = "-87.5" miny = "30.0"

maxx = "-59.5" maxy = "50.0" />

<OUTPUT file = "C:\ArcIMS\output\

usa-image-MYCOMPUTER2953026.jpg"

url = "http://mycomputer.domain.com/output/

usa_image_MYCOMPUTER2953026.jpg"/>

</IMAGE>

</RESPONSE>

4.5 C#.NET:

Visual C# .NET is a sever side language which has to be implemented on Microsoft .Net environment [MSDN 05] It includes an interactive development environment, visual designers for building Windows and Web applications, a compiler, and a debugger. Visual C# .NET is part of a suite of products, called Visual Studio .NET. The C#

programming environment is basically used to display results extracted from spatial and non spatial data base. These results are displayed as a single table using Datagrid functionality of C#.Net. Furthermore, C# program can be used to query two or more database requests from a single page. Also, the raster data extraction algorithms have to be formulated using the C#.Net environment. Below is a table showing different tasks and their implementation tools.

Tasks	Implementing Tool
Create Image Services (the form of Web services)	ArcIMS Author, XML editor, Notepad/WordPad
Publish and administer ArcIMS Services	ArcIMS administrator
Design Web pages	ArcIMS Designer
Customize ArcIMS Website	Java script, Adobe Photoshop
Request for drawing route and buffer on the computer screen.	ArcXML
Display the extracted results on the browser.	C#.net
Algorithm for downloading Raster based data	C#.net

Table 4.1 Different tasks with the implementation tools.

Chapter 5

Methodologies and Implementations

This chapter outlines the phases of this research's methodology. In addition, this chapter also describes the framework for project implementation.

5.1 Methodology

The methodology is sub divided into the following steps:

A) System setup and software requirement:

The system setup required installation of ArcIMS, ArcSDE on windows environment.

Below is the detailed list of software used for the implementation.

Operating system – Windows 2000 server

Web server - IIS 5.0,

Internet mapping server - ArcIMS 4.01

Servlet engine - Tomcat 4.0

Server side programming language – C#.Net

Database server - SQL Server 2000

Spatial database engine – ArcSDE 8.0

B) Collection and Conversion of Spatial Data into a Geo database

After installation of the required software mentioned above, collection of different types of GIS datasets was the next stage of this research. The different GIS datasets include vector and raster data. All the required data was provided by the Department of Geography, Oklahoma State University. The maps of Oklahoma were used as the test dataset. This data was converted into enterprise Geodatabase using ArcSDE. SQL Server 2000 was used as the database server. ArcSDE served as the gateway between ArcIMS and SQL Server 2000.

C) Creation of ArcIMS Website

After the installation of ArcIMS and GIS data collection, the next step is to design ArcIMS. The ArcIMS HTML viewer website was designed using ArcIMS Designer. As the default HTML viewer is not user friendly, it needed customization to make it user friendly. This customization involved changing the look and feel of the website and adding the new functionalities that were created during this research. The customized website consists of a set of HTML files, Java Script files and graphic images. Customization was achieved by the modifications to the graphic images and changes to the HTML and Java Script files. Additional functions such as Draw Highway, Extract Data, and Display attribute, etc. were also included in the customization. This customization involved only client side scripting. The additional functions needed to be handled on the sever side.

5.2 Implementation:

Implementation of these applications required extensive programming expertise in ArcXML, C#.Net, SQL Script, Java Scripts and graphic designing. The implementation involved client side, server side and SQL database programming. The client side scripting involved,

- 1) Designing ArcIMS website using ArcIMSs designer
- 2) Customizing the look and feel of the default website by changing HTML frameset of the main page. The main page of the website consists of several frames. These frames need to be loaded on the client's browser in sequential order. The Hidden frame needed to be loaded first because it declared the global variable used in the application. A new Java Script function was added to handle client side request. The new Java Script functions were listener of mouse click, mouse move and mouse up events. Hidden fields were used to pass parameters from one page to another.

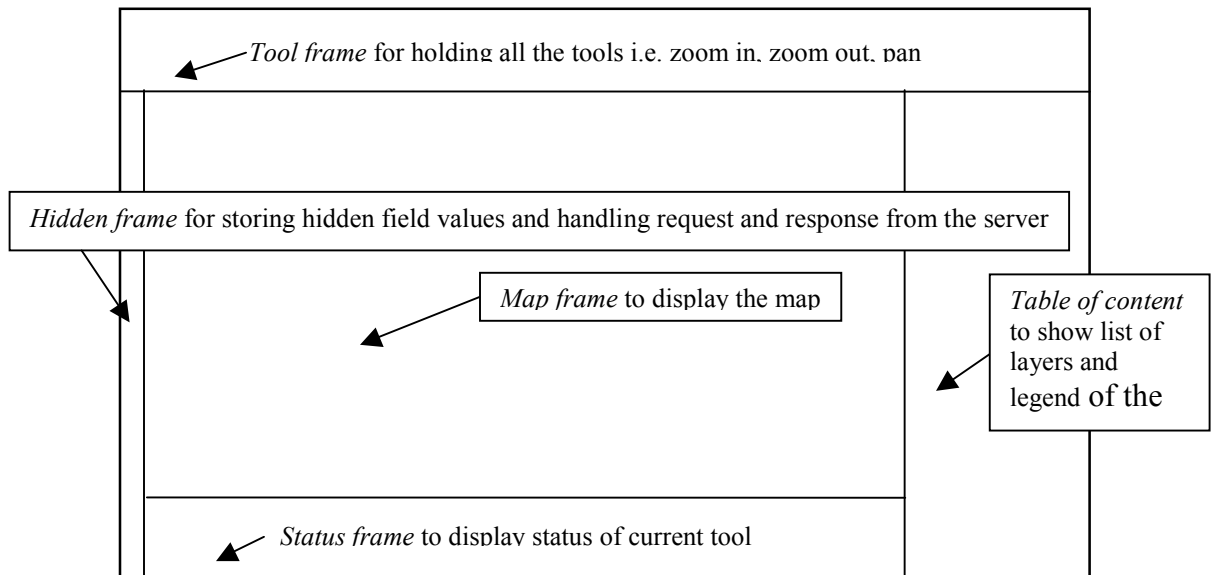


Figure 5.1 Frameset of customized ArcIMS website.

5.2.1 Implementation of Extract Function for Downloading Raster and Vector data model:

The objective is to design a function to download vector and raster data in compressed file over the Internet. The ArcIMS website uses its Extract server to download vector data set. After the extraction, ArcIMS server creates one compressed file (.zip) consisted of vector data in ShapeFile format. The compressed file is then moved to a default virtual directory called output directory. Each ShapeFile is the combination of at least three other files, the main file - for storing feature type (point, line or polygon), index file and .dbf file. The .dbf file consists of a table in dBASE format. This table contains all the feature attributes with one record per feature. The Extract server extracts all the visible ShapeFile including the 'Download Raster' ShapeFile. The 'Download Rasters' .dbf file contains one unique ID row called 'Raster ID'. All the raster data images are stored in the server. The Raster data images must be named to match exactly with 'Raster ID' of the 'Download Rasters' .dbf file.. A unique algorithm is written in C#.Net for implementing the below steps mentioned.

- 1) The compressed folder stored in the output directory needs to be uncompressed. and all the files then need to be stored in one temporary folder on the server.
- 2) The .dbf file is read to access the unique identification number of Raster data files and stores the numbers in the memory. Retrieve all the Raster images from the server by matching their names with identification number accessed from the .dbf file.

- 3) All the Raster images are collected in a temporarily folder on the server and the folder is compressed. Now the compressed folder containing all the requested Raster data and vector data is available on the server.
- 4) The compressed folder is placed in a virtual directory and the link is passed to the user as a response.

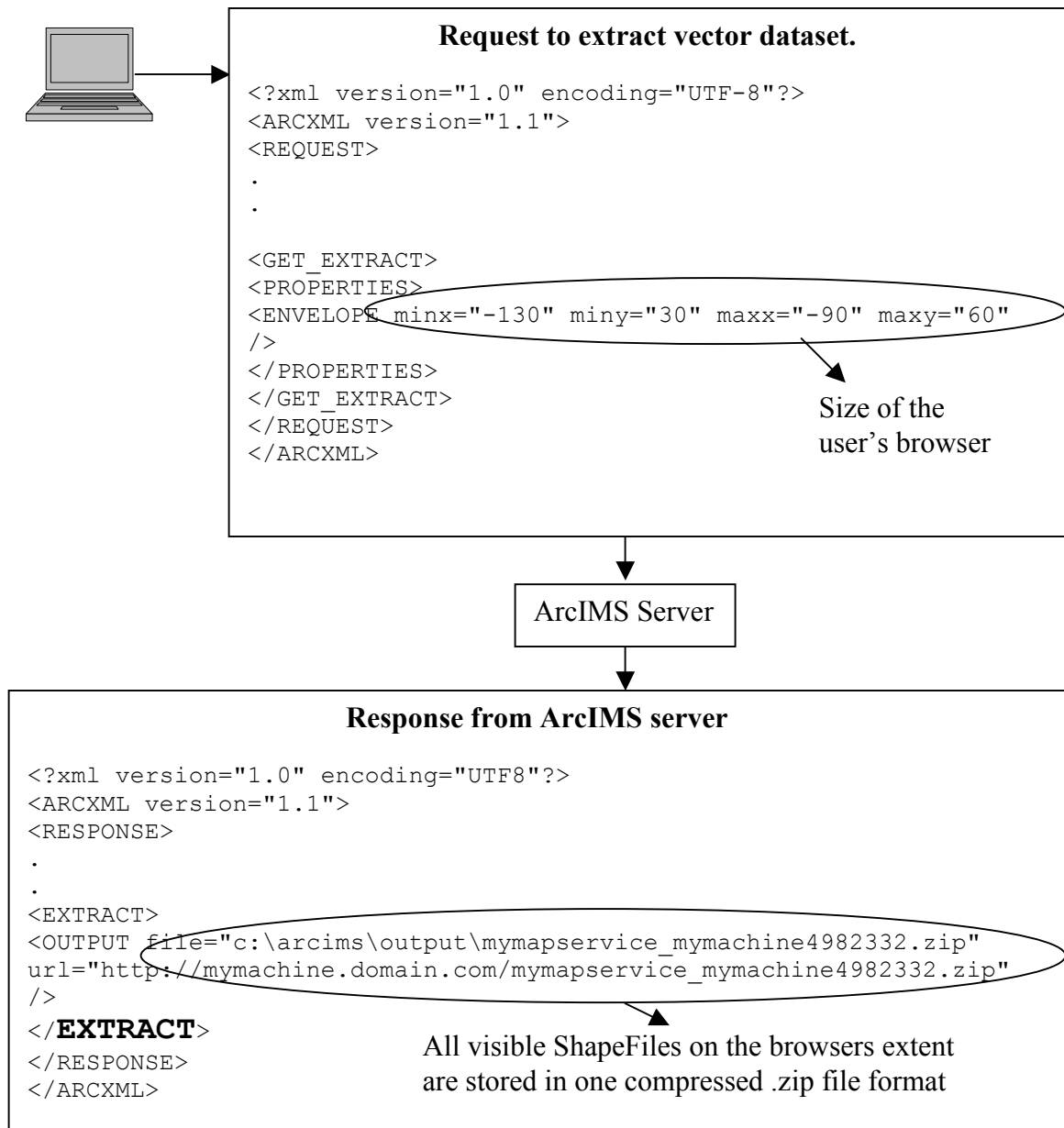


Figure 5.2 Flow diagram of request and response from ArcIMS Extract server.

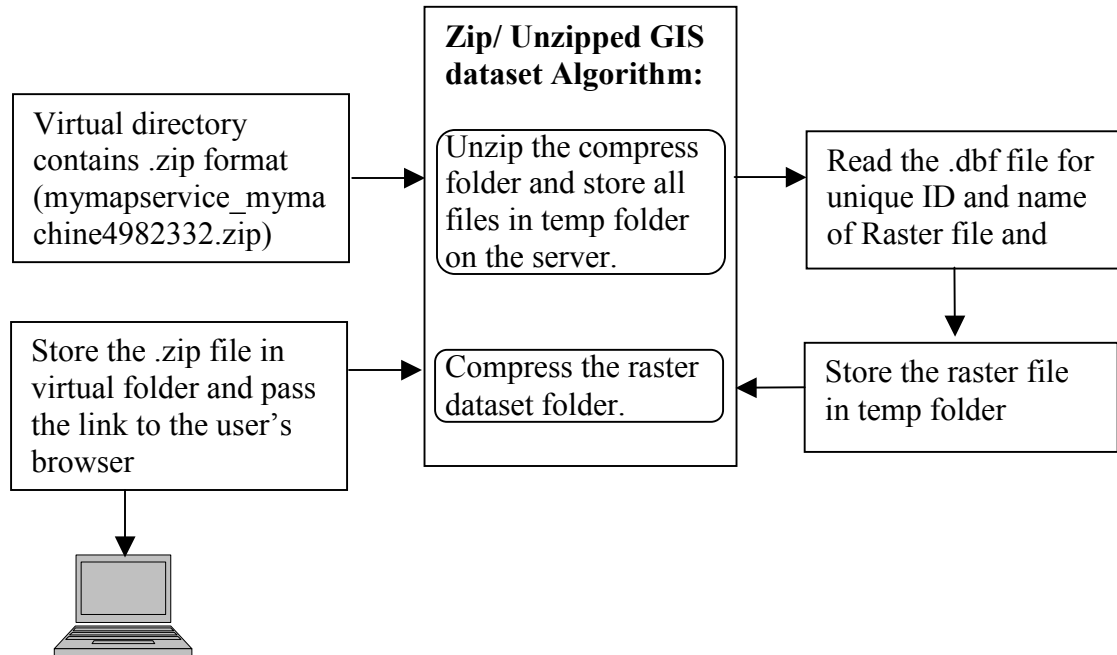


Figure 5.3 Flow diagram of extraction of Raster dataset along with Vector data.

Dynamic Linked Libraries (dll) were created for this application. This dll can be reused in any application that is written in any language (i.e. VB.Net, C#.Net or JScript) in the .Net environment.

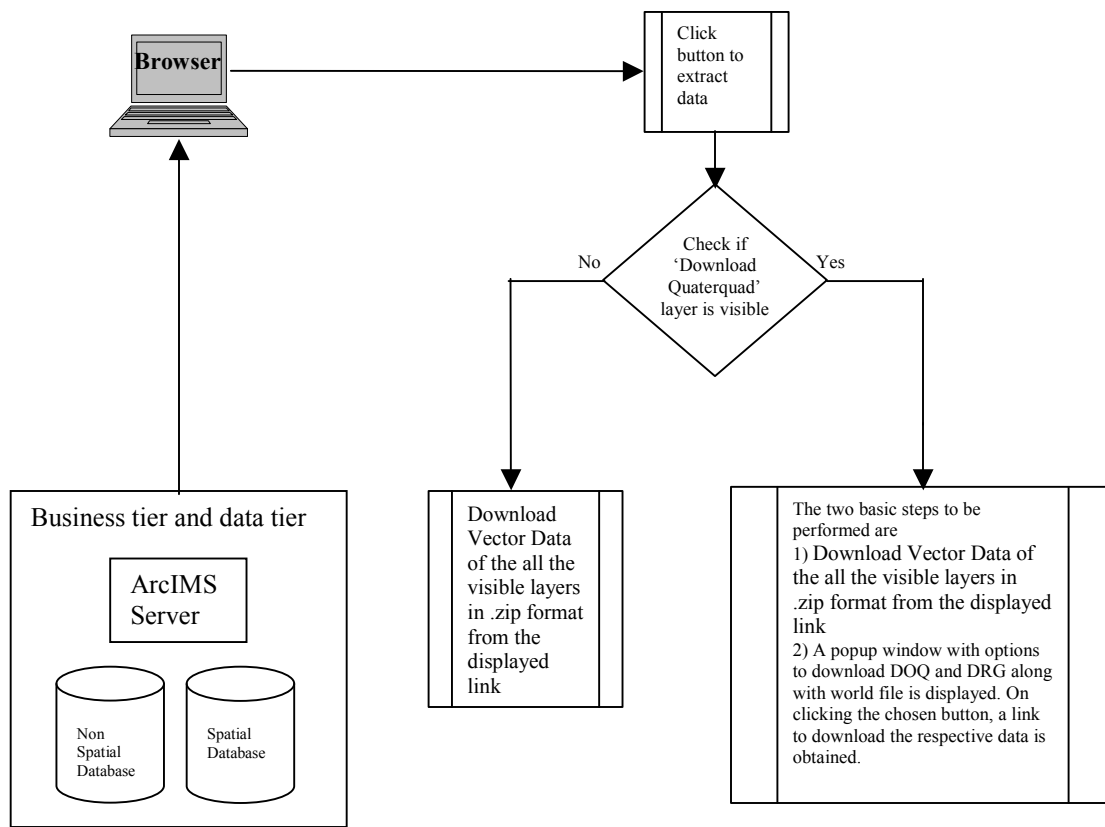


Figure 5.4 Work Flow design for the implementation of the Download of Raster data function.

5.2.2 Implementing ‘Tracing Route’

The objective here is to create a tool for drawing a line (line is called ‘Drawing Highway’/ ‘Tracing Route’ in this thesis) by clicking the mouse on the computer screen. The line needs to be buffered with a user defined buffer size All the vector data that intersects the buffers needed to be shown in the datagrid component of C#.Net. There should also be an option to download the data in excel sheet format.

The implementation of sketching a route on a map and creating a buffer around the route on the computer screen is achieved using ArcXML. The X, Y coordinates are collected on each mouse click on the map. All the points are then converted into real map points. The real map points are the actual location of the points on the earth’s surface. An

ArcXML script is generated for requesting the ArcIMS server the process new image with the portrayed route which is overlaid on the existing map. The request to the server contains all the X, Y coordinates and the request type (requesting for image or features). The ArcIMS server processes the request and in response ArcIMS sends back a new image (or attributes of the ShapeFile if it requested for feature types) to the browser. The newly displayed image consists of the new route overlaid on the existing map image. After new map with route is displayed on the browser, one input window pops up to enter the size of the buffer. A new ArcXML script is generated for the new request to show buffer around the route.

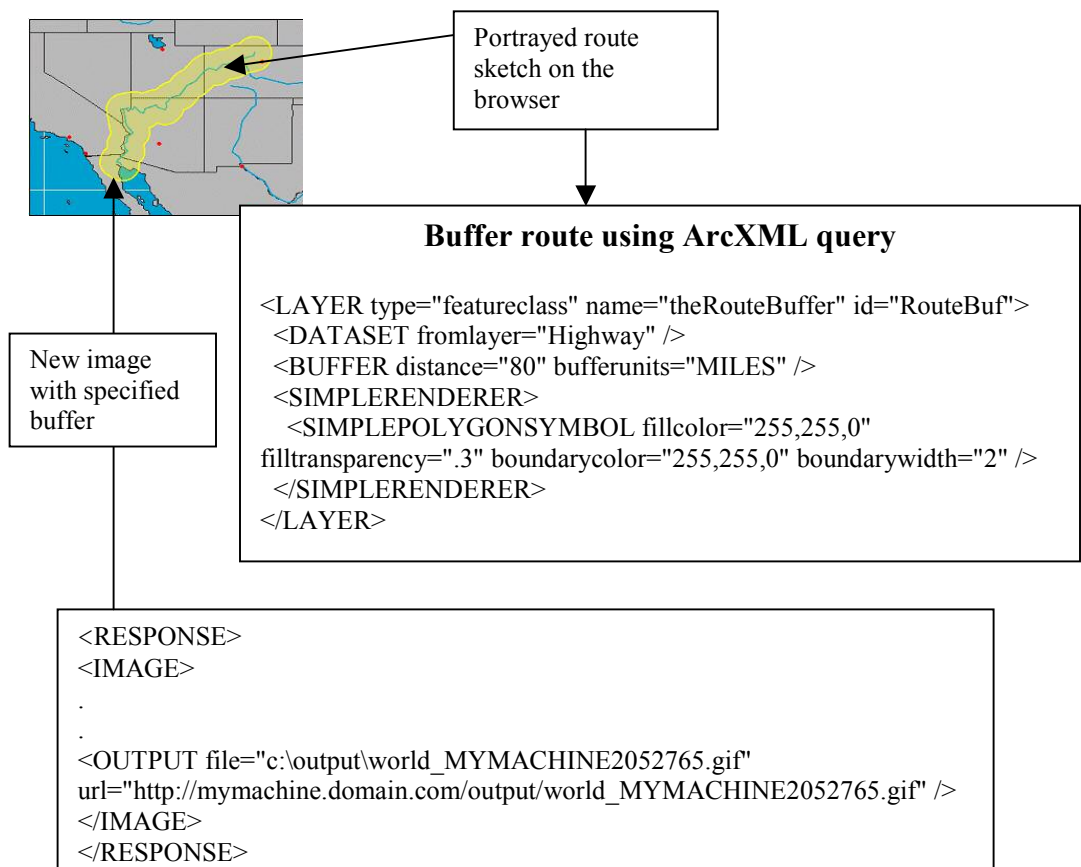


Figure 5.5 Sketch the route and display buffer around the route.

The next part is to extract all the attribute data that intersects the buffered region from the different ShapeFiles. ArcXML requests ArcIMS to get all the attribute data of the ShapeFiles. Only the data that comes within or intersects the buffer region gets extracted. The ArcXML response is later parsed using C#.Net to show the extracted attribute data on the browser in the grid format.

```
<RESPONSE>
<FEATURECOUNT count="32000" hasmore="true" />
<FEATURES>
<FIELDS NAME="Hall of fame Ave" PERIMETER=".041" #SHAPE#="[Geometry]"
/>
</FEATURE>
<FEATURE>
<FIELDS NAME="Cleveland street" PERIMETER=".011" SHAPE#="[Geometry]" />
</FEATURE>
.
.
.
</RESPONSE>
```

Figure 5.6 Attribute data that intersects the buffered region in the ArcXML format.

The data is extracted from more than one table. Each table contains hundreds of rows extracted from both spatial and non spatial data. As each table comprises of a large number of rows, the table property of HTML is not a good option to display the large amount of extracted data. The data is displayed in a simplified and user friendly manner using datagrid COM object of C#.NET [Figure 6.1.5]. Apart from displaying the data, the same data is also available to download from the web page.

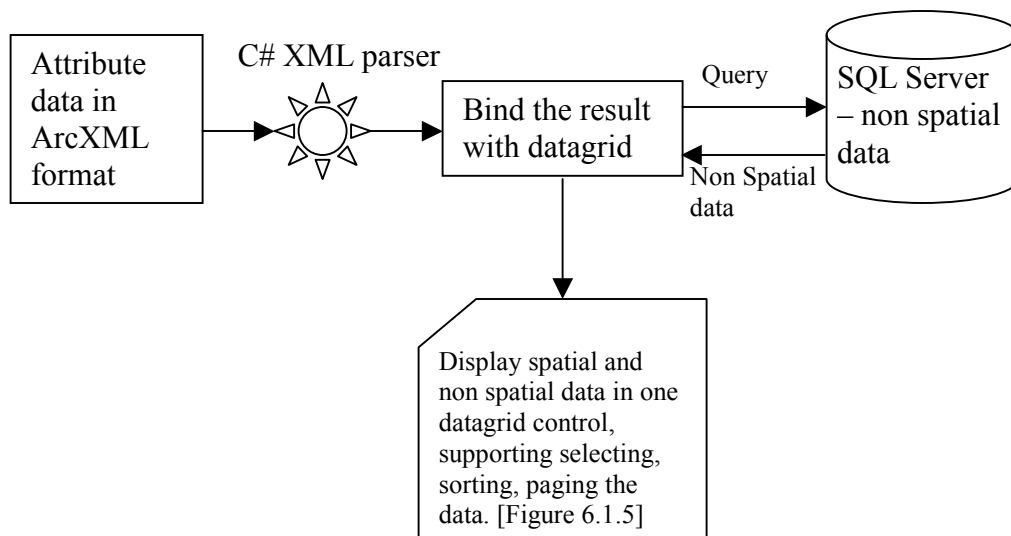


Figure 5.7 Bind spatial and non spatial data to the datagrid control.

The following steps are involved in the complete process.

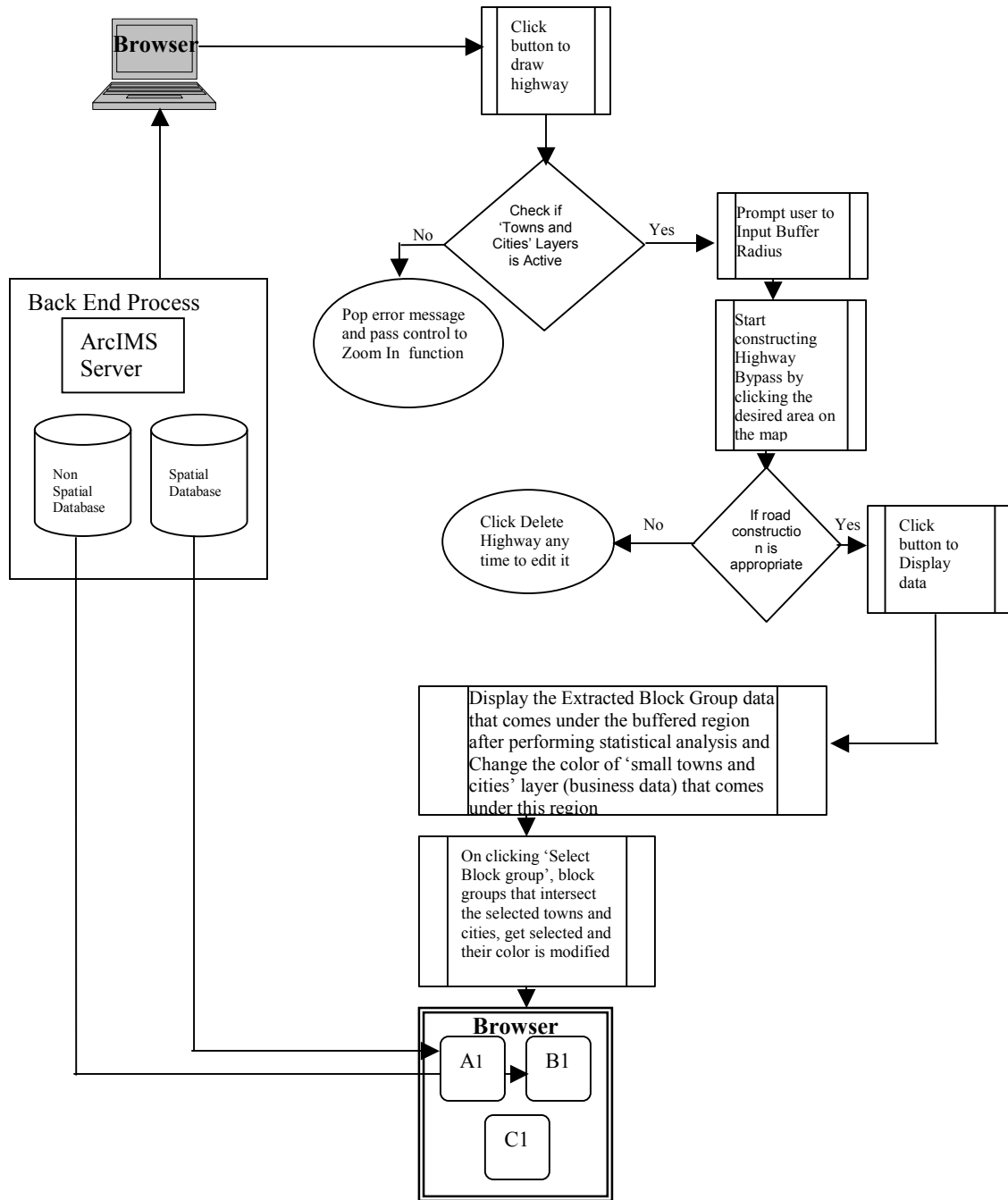
- 1) Connect C#.Net page with the HTML viewer of ArcIMS.
- 2) Extract spatial and non spatial data from two different databases at the same time.
- 3) Display data from different tables on a single C# webpage using datagrid object.
- 4) Allow the displayed data to be downloaded from the webpage in Excel sheet format in the user's hard drive.

Programming in Java Script:

By using the library functions and the global variables of Java Script in the HTML Viewer, it is convenient to customize the Web sites. The request from the browser to the server side application is sent using Java script in the form of a string. This string is

a dynamically generated ArcXML program. The main functionalities handled by Java Script in this project are

- 1) Client side functions such as popping up a window to display the alert messages.
- 2) Accept user input to determine the radius of the buffer.
- 3) Send the request and receive the response from the server in the form of a string.
- 4) Calling C#.net server side function to parse the response and display on Web browser.



A1- Selected Block Group Data, the Spatial Data comprising of geographical information, extracted from the 'block group' table of SDE database.

B1- Non Spatial Data queried which is from another database

C1- The data described in A1 and B1 above is available to download, and also to view as an Excel sheet.

Figure 5.8 Work flow design for the implementation of 'Trace Route' function.

5.3 Challenges:

After reviewing the previous work on topics related to these fields it has been noticed that it is comparatively difficult to develop these functionalities on the web rather than stand alone applications. Some of the challenges faced during the design and implementation are

- Creation of buffer around the route.
- Display the various extracted results in a single window.
- Make all extracted data downloadable in excel form
- The conversion of the compressed .zip file to uncompressed file to access.dbf file and reading the essential attributes.
- The Creation of a temporary folder to store the required Raster images and compress this folder in .zip file format to place it in a virtual directory.
- Deletion of the temporary folder from the virtual directory after the completion of users' session.

Note:

During a conversation with the team of ESRI in the 'Virginia GIS Conference 2003, Richmond' held on 18th October 2003, is the ESRI team observed that no such software with these functionalities has been used or developed by them till date. This means that the work reported here is novel. 'Trace Route' and Download Raster Data on the World Wide Web are new and developing fields where a substantial amount of work is being done.

Chapter 6

Results and Conclusions

The implementation of both the application has been successfully tested. The Oklahoma State data has been used while testing the applications. Both the mentioned application can be reused by plugging them in to any other GIS application running on .Net environment. Final testing of both the application gives the results mentioned below

- The Customized ArcIMS website is now providing a map based retrieval function for downloading Raster data model and Vector data model [Figure 6.7]. This website allows ‘what you see is what you get’ downloading of both Raster and Vector based geographic files directly to the user’s computer via compressed (.zip) aggregated file [Figure 6.8]. After the desired .zip files are downloaded to the user's computer, the various shape and image files in each .zip file can be extracted and used in ArcGIS, ArcView GIS, or any other GIS software that supports these file types.
- It is now possible to trace route on a desired area on the map online over the ArcIMS website [Figure 6.1]. The buffer around the route with user defined distance is also drawn to collect the data around the buffer [Figure 6.2]. While

doing so, the ArcXML request with specified distance for drawing buffered route is made to the ArcIMS spatial server. The response in the form of an image comes to the browser and is overlaid on the existing map [Figure 6.3]. The buffered region is then converted into an envelope and the envelope is sent to the browser for requesting the feature data that comes under the buffered region. As a response the feature data for different ShapeFiles comes as an ArcXML dynamic file. Using JavaScript the feature data is received and statistical analysis is done to display on the browser in a user friendly way. While loading the page, C#.net program makes connection with spatial and non spatial data base and displays the relevant data using datagrid component of the ASP.Net [Figure 6.4 - Figure 6.5].

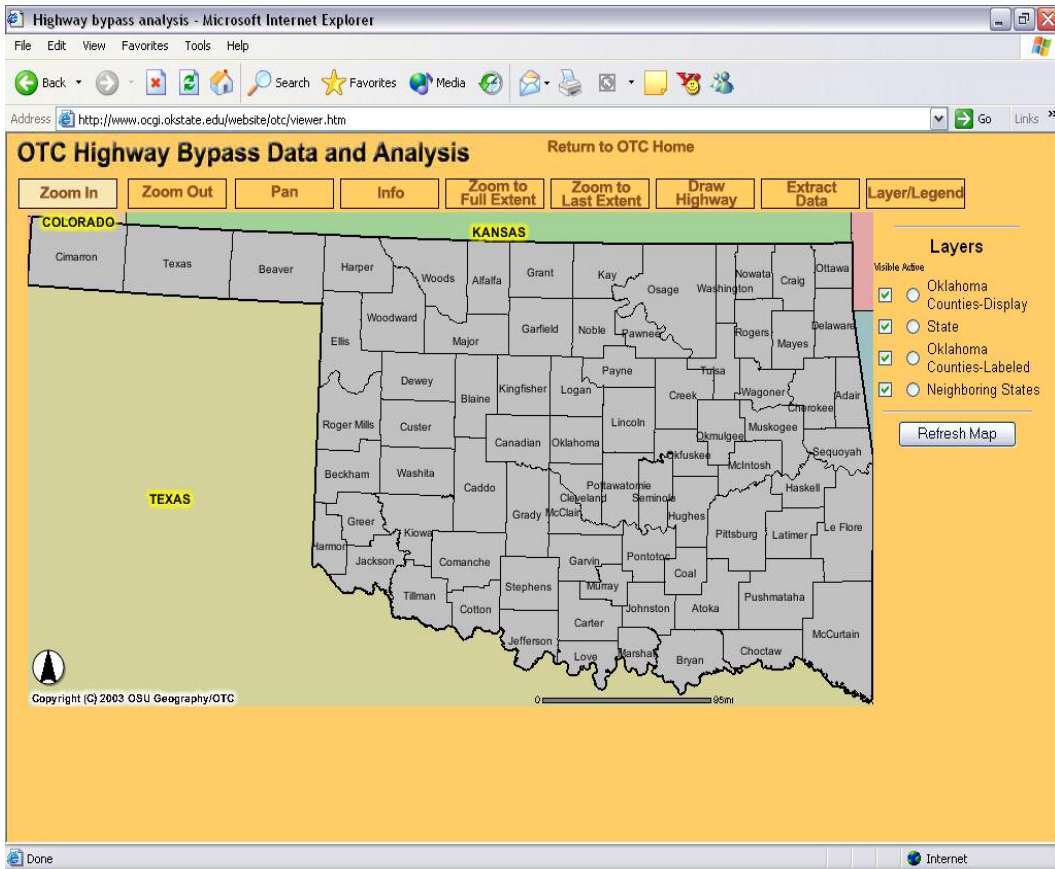


Figure 6.1 Customization of the ArcIMS website for 'Trace Route' function.

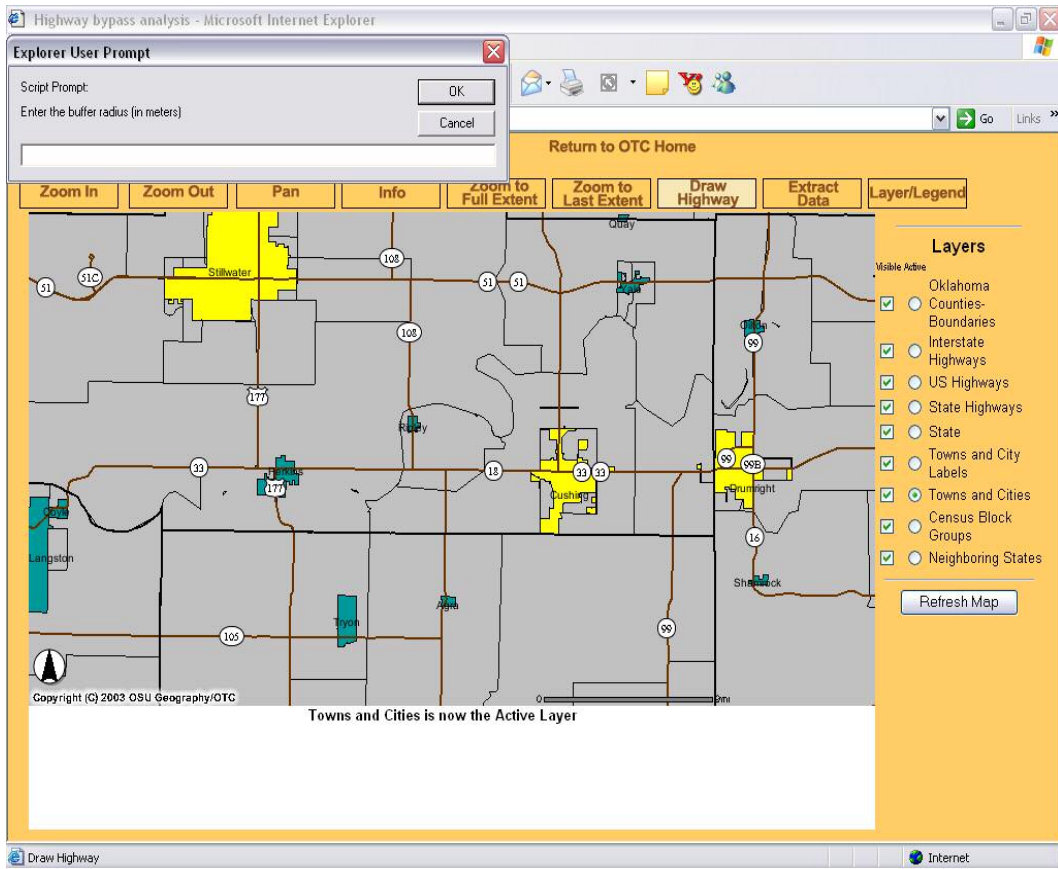


Figure 6.2 Zooming in the desired area and the input box for the buffer width specification.

Light yellow line is the 'Trace Route' and the red box covered to the yellow line is the buffer

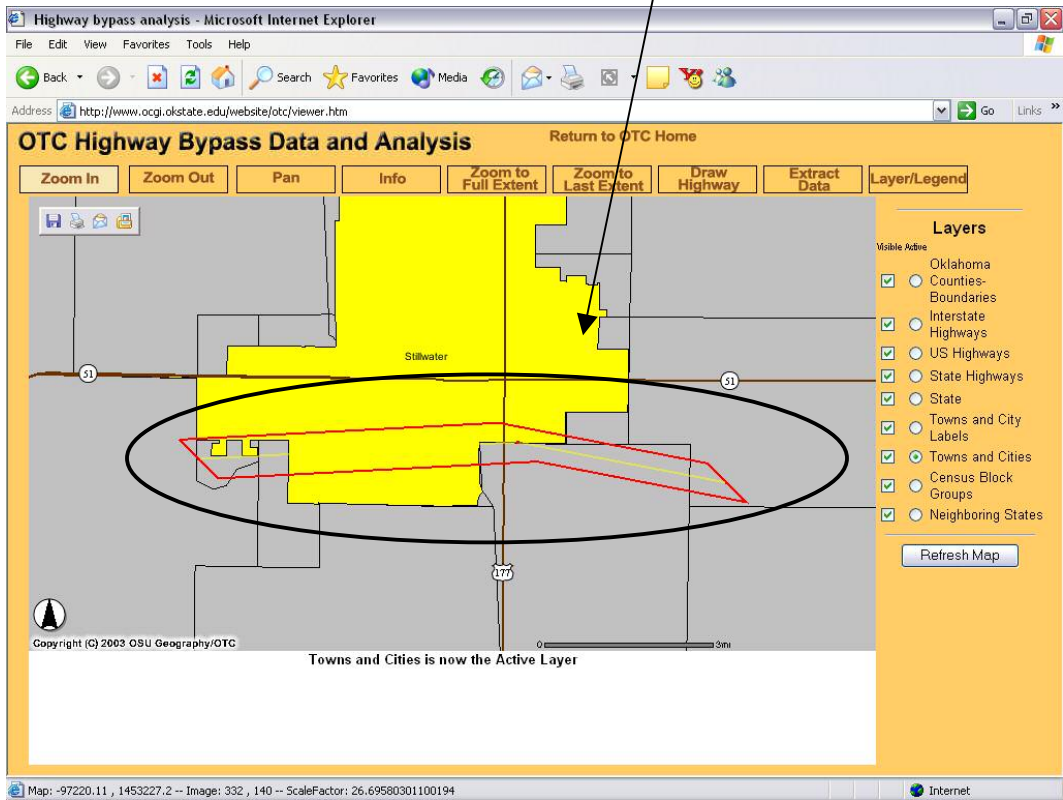


Figure 6.3 Traced Route with the buffered region.

Selected Block Group Data - Microsoft Internet Explorer

Block Group Data

	1990 Data	2000 Data	Changed Data	Changed Data in %
Total Population	3993	5191	1198	30
Total Male Population	1944	2554	610	31
Total Female Population	2050	2637	587	28
Total White Population	3564	4363	799	22
Total Black Population	54	1	-53	-98
Total Native Population	335	301	-34	-10
Total Asian/Pacific Population	0	20	-	-
Total Other Population	40	102	62	155
Total Hispanic Population	107	166	59	55
Total 65 age & over Population	339	477	138	40
Total Per capita Income	12865	22065	9200	-NA-
Total Units	1606	1906	300	18
Total Occupied Housing	1411	1790	379	26
Total Vacant House	195	116	-79	-40
Total Multibunit Housing	-	404	-	-

[Zoom to this property](#)

Figure 6.4 Result from the block group layer's region from the buffered region.

NAICS and SIC data Display - Microsoft Internet Explorer

Address: http://www5.geog.okstate.edu/website/webapplication2/result.aspx?fips=1780016400177850101012960019050190501

SIC DATA NAICS DATA TOWN DATA

Download SIC Data Download NAICS Data Download Town Data

Previous Results Next Results

Geography	1997 NAICS code	NAICS Industry	Number of establishments	Sls, shps, rcpts, rev (\$1,000)	Annual payroll (\$1,000)	Number of employees	Estab02	Estab07nbp	Estab07bp	FIPS Code	Census Code	Category	Bypl
Bixby, OK	451	Sporting goods, hobby, book, and music stores	1	D	D	0 to 19 employees	4	3	6	06400	0195	Large	No
Bixby, OK	452	General merchandise stores	3	1846	183	35	4	5	7	06400	0195	Large	No
Bixby, OK	453	Miscellaneous store retailers	4	D	D	0 to 19 employees	9	12	17	06400	0195	Large	No
Bixby, OK	454	Nonstore retailers	2	D	D	0 to 19 employees	3	3	5	06400	0195	Large	No
Bixby, OK	513	Broadcasting and telecommunications	3	D	D	20 to 99 employees	4	4	9	06400	0195	Large	No
Bixby, OK	53	Real estate and rental and leasing	11	2355	567	50	16	20	34	06400	0195	Large	No
Bixby, OK	531	Real estate	9	D	D	20 to 99 employees	11	13	23	06400	0195	Large	No
Bixby, OK	532	Rental and leasing services	2	D	D	0 to 19 employees	5	6	10	06400	0195	Large	No
Bixby, OK	54	Professional, scientific, and technical services	17	3289	1155	56	32	38	60	06400	0195	Large	No
Bixby, OK	56	Administrative and support and waste management and remediation	10	1821	884	99	16	20	24	06400	0195	Large	No

Figure 6.5 Result from the spatial database in the datagrid format.

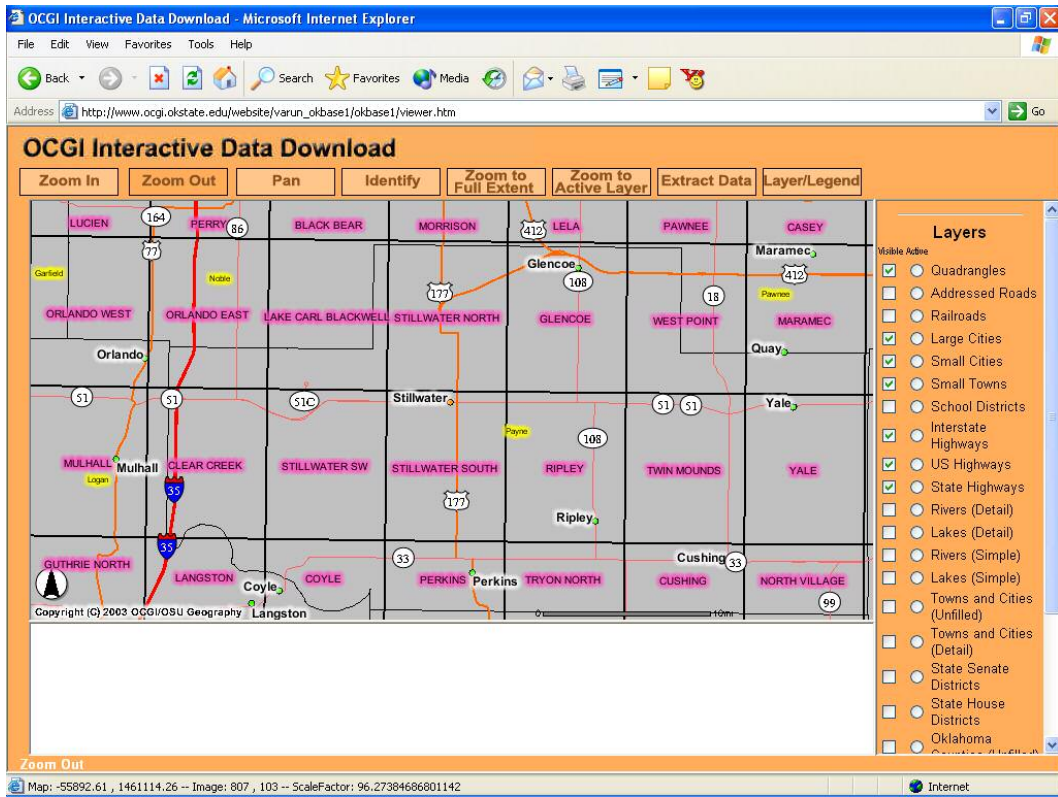


Figure 6.6 Map showing Quadrangles of desired region.

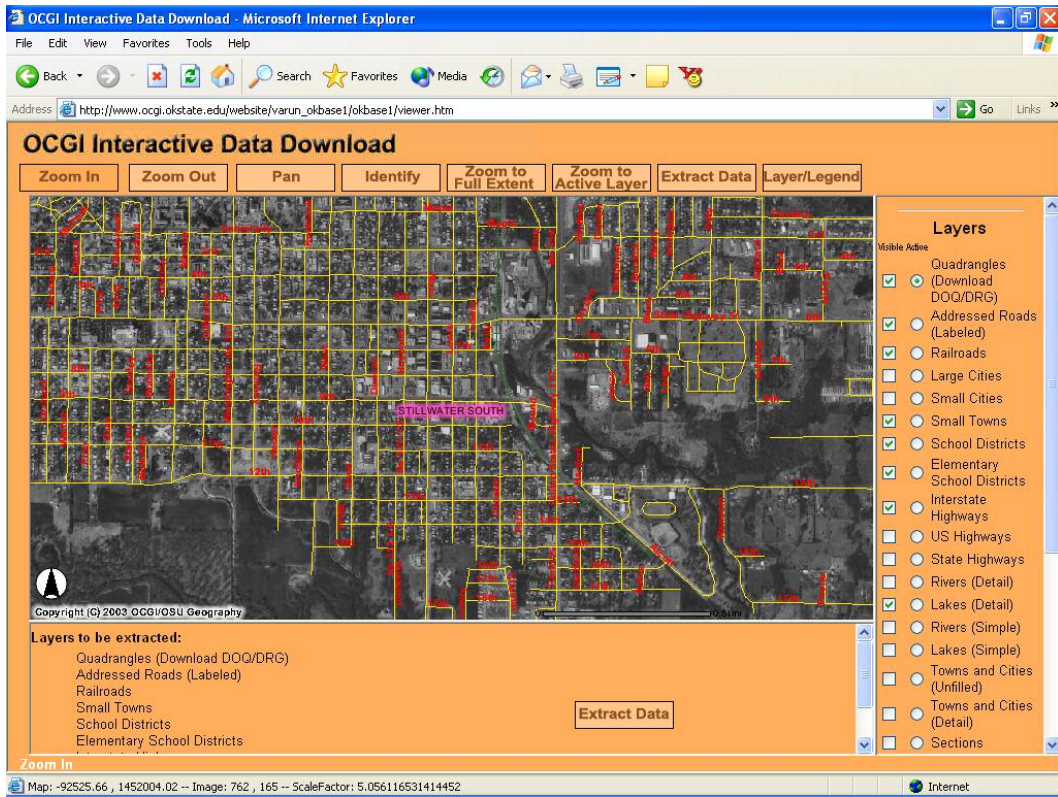


Figure 6.7 Ready to download - Raster Image with Vector data of desired area.

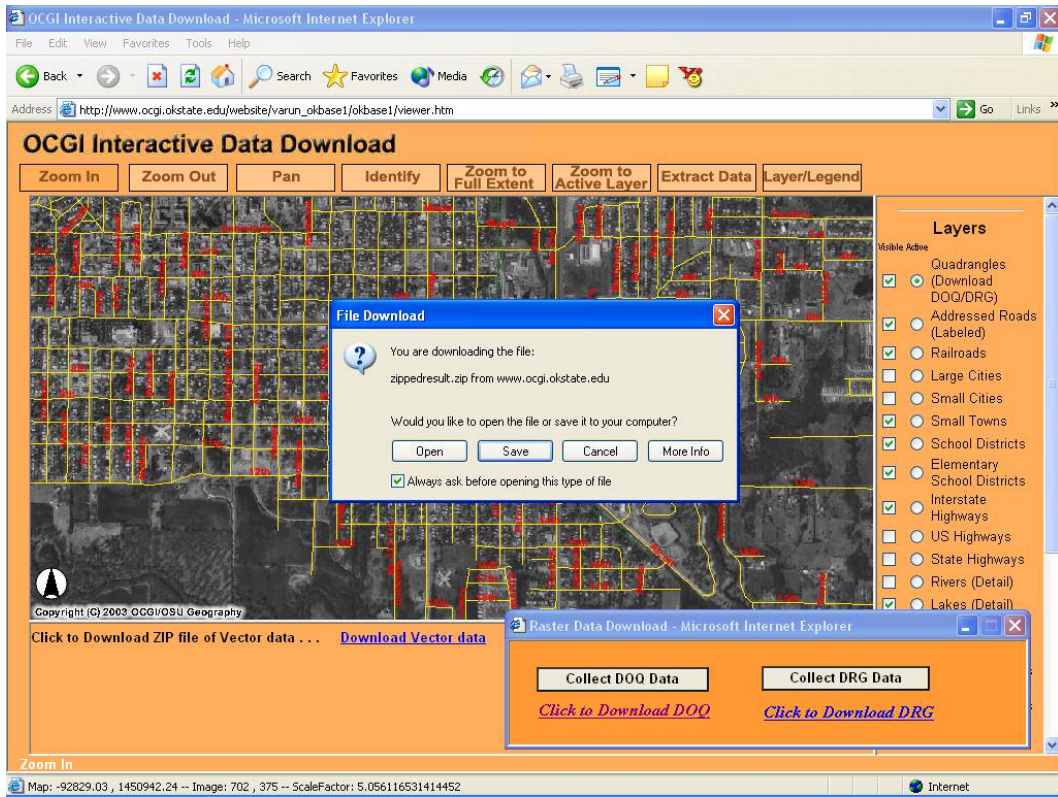


Figure 6.8 Map with a link to download raster and vector data together.

Abbreviations

ESRI – Environmental Science Research Institute

ArcIMS – Internet Mapping Services

ArcSDE – Spatial Database Engine

DOQ – Digital Ortho Quads

DRG – Digital Raster Graph

DEM – Digital Elevation Model

MrSID - Multi-resolution Seamless Image Database

SIC - Standard Industrial Classifications

NAICS -North American Industry Classification System

References

- [ArcIMS, 2001] *Using ArcIMS*, ESRI Press; Published, July 2001
- [Comer, 2000] Comer, Jonathan C., Finchum, G. Allen, and Coleman, Amanda K. "A Methodology Using Geographic Information Systems to Evaluate Socio-economic Data Concerning Impacts of Highway Bypasses in Oklahoma." *Proceedings of the Oklahoma Academy of Science* Vol. 80: pp. 79-89
- [ESRI 05] ESRI, The GIS Software Leader,
www.esri.com/library/whitepapers/pdfs/ShapeFile.pdf , last accessed Feb 2004
- [ESRI 04] ESRI, The GIS Software Leader, <http://supports.esri.com>, last accessed Feb 2004
- [Finchum, 2004] Finchum, G. Allen and Chudiwale, Varun. "Online Raster Image Extraction Using .net Technologies with ArcIMS". *2004 Annual Meeting of the Association of American Geographers, Philadelphia, Pennsylvania*, 14-19 March 2004.
- [Knowles, 2002] Anne Kelly Knowles, *Past Time, Past Place: GIS for History*, Published April 2002
- [Longley, 2001] Paul A. Longley, Michael F. Goodchild, David J. Maguire, David W. Rhind, *Geographic Information Systems and Science*, Published June 2001
- [MSDN 04] .Net framework development center,
<http://www.msdn.microsoft.com/netframework/>, last accessed Feb 04
- [MSDN 05] Visual C# development center, <http://www.msdn.microsoft.com/vcsharp>, last accessed Feb 04

[OCGI 04] Oklahoma Center for Geo science information, <http://ocgi.okstate.edu> last accessed Feb 2004

[Opadeyi, 1995] OPADEYI, J. (Univ. of West Indies, St. Augustine, Trinidad and Tobago). "Geographic information systems: an information management tool". *FID News Bulletin*, 45(3) Mar. 1995, p.77-80.

[Troelson, 2001] Troelson, Andrew W., *C# and the .NET platform*, Apress, 2001

[Uma 2000] Sekaran, Uma (2000). "*Research Methods for Business A Skill Building Approach*". 3rd Edition. New York, NY: John Wiley and Sons, Inc.

[WEB 01] GeoExplorer | TELUS Geomatic,
http://www.telusgeomatics.com/tgmain/geoexplorer/geoexplorer_new2.htm, last accessed Feb 2004

[WEB 02] Earthpace, LLC - Environmental Consulting Services,
<http://www.earthpace.com/nepaM/emergingtechglossary.html>, last accessed Feb, 2004

Algorithm for Downloading Raster data

```
DownloadRasterButton_Click(object sender, System.EventArgs e)

//Locate the .zip in the output folder of the server and unzip the .zip file
UnzipFile(string zipFile, string destinationDirectory )

//store all the files in the .zip in one folder and store that folder in temp folder
CompressFile(ZipOutputStream s, string path, string entryPath)

//Read the unique ID from the shapefile
ReadRasterID(string filename)

//load all raster in temp folder and compress the folder
CompressDirectory(ZipOutputStream s, string path, string entryPath)

//for deleting all file other than recent .zip, ArcIMS creates bunch of other file in
temporary folder
DeleteTemp (String directoryName,String nodelete)

//Show link for the compressed folder
Showlink(String linkPath)
```

Codes for the algorithm

```
//load all raster in temp folder and compress the folder
public static void CompressDirectory(ZipOutputStream s, string path, string entryPath)
{
    if (entryPath == "")
    {
        entryPath = new DirectoryInfo(path).Name + "/";
    }
    else
    {
        entryPath += new DirectoryInfo(path).Name + "/";
    }

    string[] filenames = Directory.GetFiles(path);

    foreach (string file in filenames)
    {
        CompressFile(s, file, entryPath);
    }

    string [] directorynames = Directory.GetDirectories(path);

    foreach (string directory in directorynames)
    {
        CompressDirectory(s, directory, entryPath);
    }
}

public static void CompressFile(ZipOutputStream s, string path, string
entryPath)
{
    FileStream fs = File.OpenRead(path);
    byte[] buffer = new byte[fs.Length];
    fs.Read(buffer, 0, buffer.Length);
    ZipEntry entry = new ZipEntry(entryPath +
Path.GetFileName(path));
    s.PutNextEntry(entry);
    s.Write(buffer, 0, buffer.Length);
    fs.Close();
}

// unzip procedure
public static void UnzipFile(string zipFile, string destinationDirectory )
{
    // make sure the file exists
```

```

        // make sure a directory to put the files in exists
        if (!Directory.Exists(destinationDirectory))
Directory.CreateDirectory(destinationDirectory);
        if
(!destinationDirectory.EndsWith(Path.DirectorySeparatorChar.ToString()))
destinationDirectory += Path.DirectorySeparatorChar;
        // open the zip file
        ZipInputStream s = new ZipInputStream(File.OpenRead(zipFile));
        ZipEntry theEntry;
        // go through each entry in the file and write it out to our directory
        while ((theEntry = s.GetNextEntry()) != null)
        {
            System.Console.WriteLine("File " + theEntry.Name);
            if (theEntry.IsDirectory) Directory.CreateDirectory
(destinationDirectory + theEntry.Name);
            else
            { // write each entry out to a file
                int size = 2048;
                byte[] data = new byte[size];
                FileStream fs = new
FileStream(destinationDirectory + theEntry.Name, FileMode.Create);
                while ((size = s.Read(data, 0, data.Length)) > 0)
                {
                    fs.Write(data, 0, size);
                }
                fs.Flush();
                fs.Close();
            }
        }
        s.Close();
    }

//start downloading DOQ data
private void Button1_Click(object sender, System.EventArgs e)
{
    try
    {
        //checking for DOQ's if file is similar put that file in download
        directory
        foreach (string s2 in
Directory.GetFiles("D:\\SIDS\\DOQ\\") )
        {
            for(int i=0;i<count;i++)
            {

```

```

        if(image[i]!="0")
        {
            ss=image[i];
            String sss;
            // 12 is length of path till d:\sids\doq
            sss=s2.Insert(12,"downimage\\");

            if(s2.LastIndexOf(ss)>0)
            {
                File.Copy(s2,sss,true);
            }
        }
    } // end of for
} // end of foreach
ZipOutputStream s12 = new
ZipOutputStream(File.Create("C:\\Inetpub\\wwwroot\\okbase\\zippedresult.zip"));
s12.SetLevel(5); // 0 - store only to 9 - means best compression
CompressDirectory(s12, "D:\\SIDS\\DOQ\\downimage",
""");
    progress.Visible=false;
    HyperLink1.Visible=true;
    HyperLink1.NavigateUrl="http://www.ocgi.okstate.edu/okbase/zippedresult.zip";
    s12.Finish();
    s12.Close();
}
catch (Exception e1)
{
    Response.Write("Oooops. Caught an exception:\n{0}");
    Response.Write(e1.Message);
}
}

//for deleting all file other than recent .zip, ArcIMS creates bunch of other
file in temporary folder
public void Perform1(String directoryName,String nodelete)

//Show link for the compressed folder

Showlink(String linkPath)
{
    String[] fileNames=Directory.GetFiles(directoryName);

    // removing the http:/... till the zip file name
    String s=nodelete.Remove(0,45);

```

```

        for (Int32 i=0;i<fileNames.Length;++i)
        {
            // delete all file other than s
            if(fileNames[i].EndsWith(s)==false)
            {

                File.Delete(fileNames[i]);

            }

        }

    }

    #region Decompress code - contains the methods providing
    decompression
    public string DeCompress(string stringToDecompress)
    {

        string outString = "";
        try
        {
            if (stringToDecompress == null)
            {
                throw new
                ArgumentException("stringToDecompress","You tried to use an empty string");
            }
            //Where the conversion is done (added Trim to get over a
            particular problem with some XML - spaces can creep in!)
            byte[] inArr =
            Convert.FromBase64String(stringToDecompress.Trim());
            StringBuilder sb = new StringBuilder(4096);

            sb.Append(System.Text.Encoding.ASCII.GetString(DeCompress(inArr)));
            outString = sb.ToString();
        }
        catch(ArgumentException aEx)
        {
            return aEx.Message;
        }
        catch(ObjectDisposedException oEx)
        {
            return oEx.Message;
        }
        catch (NullReferenceException nEx)

```



```
    {  
        return nEx.Message;  
    }  
    return outString;  
}
```

VITA

Varun Chudiwale

Candidate for the degree of
Masters of Science

Thesis: Development of processes for Online GIS Data Selection, Extraction, and Aggregation using ArcIMS and .Net Technology

Major Field: Computer Science

Biographical:

Personal Data: Born in Kanpur, India, On May 3rd 1978, the son of Shri. Shrikant and Smt. Snehlata Chudiwale

Educational: Graduated from Bapurao Deshmukh College of Engineering, Sewagram, India in May 1999; received Bachelor of Engineer degree in Power Electronics. Completed the requirements for the Masters of Science degree with a major in Computer Science at Oklahoma State University in December 2004.

Certification in Geographical Information Systems (GIS) in May 2004.
SUN Certified JAVA Programmer.

Experience: Research Assistant; Oklahoma State University, Department of Geography, 2002 to May 2004.

GIS Application developer, United State Department of Agriculture (USDA), Forest Services, Salt Lake City, UT, June 2004 to present.