2003

Internet cartographic software application for the regional economies assessment database

Timothy J. Abbenhaus
The University of Montana

AN INTERNET CARTOGRAPHIC SOFTWARE APPLICATION FOR THE REGIONAL ECONOMIES ASSESSMENT DATABASE

by

Timothy J. Abbenhaus

B.S. University of Utah, 1990

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

Master of Arts

The University of Montana

May 2003

Approved by:

Paul B Wilson

Chairperson

Dean, Graduate School

6-4-03

Date

Abbenhaus, Timothy, M.A., May 2003 Geography

An Internet Cartographic Software Application for the Regional Economies Assessment Database (71 PP.)

Advisor: Dr. Paul B. Wilson PW

Internet cartography is the modern extension of traditional cartographic techniques. It is used to present a variety of data in map form. Used in the technique are modern programming philosophies and technologies for producing quality dynamic maps. This technique is applied to large dynamic datasets that until recently, due to a lack of technology, could not be shown in maps.

Object-oriented programming standards have been applied in the programming of the software application subsequently named the Internet Cartographic Software Application (ICSA). The programming environment used to program the ICSA allowed its construction to be completed quickly.

This computer functionality has since its installation on the server for the Center for the Rocky Mountain West in the spring of 2000 proved to be secure and stable presenting regional economic maps to user's requests. The same methods have been applied in building new versions of ICSA for natural resource information and installed on servers at the Montana State Library Natural Resource Information System.

The implementation of this study indicates cartographic methods employed with traditional tools can also be employed in modern programming tools. This is central to the philosophy of ICSA which shows how modern tools have been used to create dynamic maps. Programming and cartography are using the Internet to deliver economic maps. Noted is the importance of the geographer cartographer in creating an ICSA.

TABLE OF CONTENTS

LIST OF FIGURES

Figures .... Page

Chapter 1

INTRODUCTION

Beginning in the year 2000, the author developed a software application that dynamically and interactively delivers cartographic data contained in the Center for the Rocky Mountain West's Regional Economies Assessment Database[1] as map images using the Internet. The new application, subsequently named Internet Cartographic Software Application (ICSA), currently resides on servers maintained by both the State of Montana Natural Resource Information System, and the Center for the Rocky Mountain West. These sites are accessed by Internet users every day. This document describes the production of ICSA including step by step programming instructions. Recent advances in data availability have created new challenges for cartographers and programmers who generate images for geographic information system (GIS) applications[2]. Integrating up-to-date information for mapping applications requires connecting of two distinct entities, the cartographic software and the data. The purpose of ISCA is to develop capabilities for imaging economic data in cartographic

[1] ***READ*** stands for the ***Regional Economies Assessment Database***. ***READ*** is a computerized database and evaluation system designed to allow the complex structure and functioning of the economy to be systematically examined and assessed at many different levels.

[2] A glossary of terms used in this thesis appears on page 58.

applications. Interactive methods to deliver data to the Internet user has the potential to make information of this type widely available to cartographers, resource managers, regional planners, and the public. The process of developing mapping capabilities illustrates problem-solving approaches that may be useful for integrating other types of data into GIS as it is used on the Internet.

The Internet provides the opportunity for cartographers to create dynamic up-to-date renderings of topics. This thesis describes the use of modern programming techniques to present cartographic information by the Internet. It explores the problems arising in the process of developing this computer geographic capability, explains how these problems were surmounted, and establishes the advantages of using these techniques in cartography. In addition to this thesis, the project generated an Internet Cartographic Software Application (ICSA). This ICSA uses Microsoft Visual Basic 6.0 (VB6) and Environmental Systems Research Institute's (ESRI) Map Objects 2.0 (MO) software, to incorporate updated information available in large digital databases into maps.

To incorporate the information in these large databases into maps assembled and delivered via the Internet, a programmer must be proficient with programming languages used to access database contents, and proficient in GIS and Internet programming methods. The database whose contents this application was created to map, using the Internet, is known as Regional Economies

Assessment Database (READ). The READ contains census and economic survey data for individuals living in 1500 counties of 22 western states. The database also contains information, such as locations of motor vehicle, rail, and air transportation corridors and hubs, across national, regional, state and county scales. Maps assembled by the ICSA integrate information from economic and census surveys with these geographic analyses.

A database is typically comprised of individual tables, each of which contains information on particular topics. In programming to create the READ, demographic parameters such as population and income are assigned to tables. Rather than being organized on the basis of tables, the geographic database consists of Shapefiles (shp), each representing a unit of geographic enumeration.

The exploration of Internet cartography described in this thesis includes new concepts, explanation of which requires definitions of technical terms. The method used to develop the cartographic application for this project, called "object oriented programming," emphasizes three principle tenets: polymorphism, inheritance and encapsulation. In general usage, polymorphism is the ability of something to appear in many forms. In object-oriented programming, polymorphism refers to a programming language's ability to process objects differently depending on their data type or class.

Used in programming, encapsulation is one part of the process of combining existing data elements to create a new entity. This encapsulation refers to the ability to program code that can be used repetitively in the program, in the form of modules that can be used where needed. Encapsulation allows the programmer to set aside pieces of the code separate from the main program in order to facilitate clearly and concisely written routines.

Inheritance, in programming, means the repetitive use of methods or properties, by taking or receiving them from a preceding assignment in the code. This property of inheritance is associated with encapsulation, as both are attributes of modular code that ,once written, can be reinserted as needed throughout a program or programs.

All three properties are essential elements of object oriented programming, but even the most sophisticated modern programming languages do not utilize all three properties. Visual Basic, for example, is not considered a true object oriented language because it is not truly capable of polymorphism in the sense that it does not lend it self to alteration by programmers to meet the needs of all types of applications. Visual Basic does, however, provide a versatile platform, enabling programmers to use object-oriented methodologies.

A graphical user interface (GUI) provides the programmer the ability to devise and select particular "objects" that the programming language uses in the software application. An advantage of using a GUI is that it allows the cartographer to quickly program and test individual applications or modules of code early in the development process. Additional terms necessary in discussing Internet cartographic applications such as object, class, database, Active X, Active Data Object (ADO), Object Database Connectivity (ODBC), Object Oriented Programming (OOP), Standard Query Language (SQL), relational database are defined in the glossary which should be consulted as they are encountered.

The Internet Cartographic Software Application developed for this project does not require the user to be proficient in GIS techniques. Compared to less user-friendly interfaces available, the application creates the displayed message of the spatial data, and allows these data elements to be combined without a user having knowledge of the GIS software.

Online GIS can be considered as an alternative to nontraditional approaches, for obtaining useful data for geographic analyses. In online GIS applications, users must download a small executable software application. When activated, this software supplies tools enabling the use of geographic files and documents to create maps and to perform a number of analyses selected by the user. The computer completes processing on the side of the server where the map and the web page dynamically become a product ready to return information to the user.

The production of digital "soft-copy" maps has become as important as printing conventional paper maps. Available interface tools allow users to view dynamic, multi-layered, digital map data using web browser software. To employ the Internet as the medium for map display and end user inquiry, however, the cartographer must prepare a presentable and effective GUI for the user through programming. The interface is produced with the development of a cartographic software application that allows the user to view maps that are based on all of the data contained in any database the programmer chooses to include. Cartographic programming for digital, online display, enables users to access a vast array of data built into and available for extraction from a single map application. The creation of such maps facilitates intuitive browsing by end users submitting unique queries. This radically simplifies the methods of data manipulation currently available in GIS applications making information widely available.

Objective

The objective of this thesis is to generate and demonstrate a cartographic software application for use on the Internet that exemplifies online cartographic programming. The thesis explores the possibilities and pitfalls of this technique and considers possible standards for these types of software development in facilitating future computer-based cartographic techniques.

The development of this software is explained in the methodology chapter along with exploring and analyzing the Internet for mapping. The distribution with Online GIS and Internet cartography and other issues that directly relate to Internet cartographic map production and programming challenges are compared and contrasted.

Briefly reviewing recent improvements in personal computers makes it apparent that the hardware requirements for conducting cartography are now available in most machines. As the personal computer underwent change, cartography evolved into a revolutionary new tool. Utilizing the Internet as the medium of display redefines map data as now being dynamic thus widening the possibility to create maps. This widening allows data, including population, income, employment and other demographic information to be mapped in the Internet environment for states, counties, or even for sub-county areas. The ICSA, however, differs from network distributed GIS in that it provides a pre-manipulated analysis of applicable data and allows "read only" access to the database used in an ICSA. A network distributed GIS, in contrast, allows system users both "write" and "read only" access to a database. Accommodating this difference presents problems, which the modern cartographer solves using software tools specifically created and designed for this type of programming.

Generating the proposed application has required the author to navigate unexplored territory, especially in applying diverse programming concepts and techniques for creating Internet displayed thematic maps driven by data. A record of steps taken, and assessments of success or failure, has been essential to completion of this task.

The study began with research to identify potentially useful programming languages, database design elements, and methods currently available for displaying geographic and demographic data. The focus was narrowed to economic data, and emphasis was placed on making information available to Internet users with varying degrees of computer operation experience. Symbolization and color choice capabilities, for example, were optimized in the development of display methods that would both inform and appeal to diverse audiences.

To define more precisely the methodological advancement to be achieved, the development of cartographic tools in general were reviewed as was the status of GIS software and the Internet. Results of this review appear in Chapter 2, "Internet Cartographic Technique." Besides definitions, this chapter illuminates comparisons and contrasts between Internet cartography and online GIS. Based on economic data, a model for the methodological approach was possible because the project was accessible and fully supported by an economist and an associate programmer.

Programming considerations involved in maximizing hardware and software compatibility are discussed in chapter three. The process by which the connection to the READ database is discussed in chapter four.

Chapter 2

INTERNET CARTOGRAPHIC TECHNIQUE

The Internet cartographic technique has arisen as a result of several innovations in the personal computer and mainframe computer technologies. An automated Internet cartographic mapping application creates a map and conveys an intended message through the World Wide Web then provides further information upon the user's request. An automated Internet cartographic mapping application is a software application designed to handle information similar to that conveyed by a custom map set. A personal computer, established as a server, provides the necessary computations for creating maps that can embody various topics, such as economics.

A literature review has been included, chronicling the evolution of cartographic methodology and Internet applications. While differences between digital cartography and online GIS methods of accessing information initially led to divergence in software development, more recent applications generate output to both GIS and other cartographic software integrating connections to remote databases.

History of Internet Cartography

The late 1960s marked the start of Computer-based GIS, of which, extensive literature has been published describing two evolutionary stages of GIS. The first of these stages was the "Pioneering Age" from 1960 to about 1975[3]; it was named by many individual achievements that highlighted the leaders knowledge and ambitious programming. This stage was dominated by national agencies whereby abundant experimentation was begun that led to common duplication of efforts. The second stage was the "Age of Commercial Enterprise" that continues to today and dominates software development. Today, competition among vendors, standardization on open systems, and further agreements on the user's perception of what a GIS should do and look like, has produced a period of user dominance.

The history and evolution of the Internet is important since its roots are inherent in the relationship it has with the World Wide Web. In tracing the history of the Internet, Len Kleinrock's work in packet switching[4], ranked the beginning. The following year, the Advanced Research Projects Agency[5] became interested in computer networks and began supporting computer scientists around the country. The relationship was symbiotic and ARPA maintained the new-found and growing computer center. The vast number of computers and scientist involved in packet

[3] J.T. Coppock, et al. *The History of GIS, Volume 1.* (John Wiley & Sons, Inc., 1993), 183.

[4] L. Kleinrock *Information Flow in Large Communication Networks.* (RLE Quarterly Progress Report, 1961).

[5] The Advanced Research Projects Agency was formed with an emphasis towards research, and thus was not oriented only to a military product. The formation of this agency was part of the U.S. reaction to the then Soviet Union's launch of Sputnik in 1957. (ARPA draft, III-6). ARPA was assigned to research how to utilize their investment in computers via Command and Control Research (CCR).

switching lead to the idea of data sharing and subsequently the onset of networking.

The sharing of data by networking expanded and accelerated efforts for an Internet. The unique experiences lead by Kleinrock found a place in Washington D.C. in this effort. The government committed to creating the framework and infrastructure for the new technology based on "packet/packet switching" that would lead to the Internet, a network for scientists to exchange information over large distances. The ability to network computers beyond the very limited experimental laboratory or local area network of computers made an environment where scientists could exchange information and harness the power of all the computers on the network for large computations.

Data networking, among other ideas of Kleinrock, played an essential role in the preparation of a functional specification for the Advanced Research Projects Agency Network (ARPANET). ARPANET is the government supported data network that first utilized the technology known as "packet/packet switching."

The development of the World Wide Web and its growth over the past 12 years has nurtured the information age that would otherwise be much smaller and more specialized. Tim Berners-Lee is credited as the inventor of the World Wide Web and with coining the term "WWW" in 1990. Berners-Lee, while at work at CERN[6], authored a memo in 1990 that outlined the World Wide Web

[6] CERN is The European Center for Nuclear Research (European Council For Nuclear Research).

as a means of communicating using Hypertext language. The WWW is the graphical interface created by using Hypertext Transfer Protocol (HTTP) and Hypertext Markup Language (HTML) protocols[7] common to all Internet users.

Hypertext Markup Language (HTML) is the basic document model for the World Wide Web. Hypertext Markup Language determines the appearance of many of the elements, for example paragraphs of text, backgrounds, lists of information, and graphics and colors[8]. Elements such as text and graphics on WebPages can link users to topics from a huge array of networks. The Internet itself provides the means for communicating the HTML code to a personal computer or network server where the user is running Internet browser software. The Internet browser software compiles and executes the HTML sent over the WWW to produce an HTML page on the user's computer. The interpretation of each HTML page on individual personal computers can be slightly different from the page created by the original designer. The nature of HTML allows WebPages to be displayed without exact specification as to color and font. This can create an issue for cartography on the Internet, since color and font are important communication tools for mapmakers. The problem is that since Internet browser software is specific to a computer, and if the display properties are limited on the computer a substitutions

[7] T. Berners-Lee, & R. Caillian, *World Wide Web: Proposal for Hypertext,* (CERN Memo, 1990).

[8] G.P. Landow, *The Definition of Hypertext and its History as a Concept,* (Johns Hopkins University Press, 2000), 102.

will be made that are often times inappropriate. For example, font types vary for each computer and thereby limits the display adapter or monitor; when this happens, the Internet browser software substitutes that component allowing display to take place regardless of program settings. The degree of control exerted by the original page designer is dependent on the method for WebPages creation and distribution.

The creation of Internet, Hypertext Transfer Protocol (HTTP) and HTML changed the linear structure of Internet documents and allowed users to branch out from one document to another through links provided by the web page author. The public, which now has access to the WWW from home, work, or the local library, is growing accustomed to the WWW method of searching for and obtaining information.

WWW access has provided cartographers with an opportunity to dynamically serve geographic information. Cartography on the Internet started in 1993 when the Xerox Palo Alto Research Center posted a pure HTML[9] interactive world map[10]. The map and its levels of interaction where rather limited, however, since the Xerox map was based on pure HTML and every level of zoom required an individual HTML page.

[9] G.P. Landow, *The Definition of Hypertext and its History as a Concept,* (Johns Hopkins University Press, 2000), 122.

[10] C. Harder, *Serving Maps on the Internet: geographic information on the World Wide Web.* (ESRI, 1998)

Internet Cartography versus Online GIS

The Internet offers extensive, continually evolving capabilities to cartographers and GIS specialists. Studying these capabilities has been central to this project. With respect to the graphic products for display, Internet cartography uses maps and map conventions similar to the traditional methods of cartography. The map, as utilized for its presentation of the topic, does not require the user in the handling of its construction. The construction is the responsibility of the programmer analyst, yet needs the expertise of the cartographer in selecting projection and legend categories as well as conducting the preliminary analysis.

Online GIS, on the other hand, gives the user the tools to process geographic data, such as selecting various attribute data for a map to display. Online GIS assumes the user has the ability to select the levels of analysis and display. This is a primary difference between the two methods; the online GIS user receives raw data and default categories while the Internet Cartography user does not. Online GIS also provides the user with GIS-style tools to manipulate the view of the data. The online GIS method provides the user with the ability to add layers, to remove layers, and to change the projection of the layers of geographic data.

The Internet cartographic map with its programmed mapping application provides the user with completed maps. This provides finished maps that are otherwise beyond the user's technical and logistical ability to create. Although both Internet Cartography and Online GIS use the Internet to display and distribute maps and data, their properties require different responsibilities from the cartographer verses the programmer.

Online GIS allows the user to obtain useful data for mapping through browser-mapping applications such as Arc-Explorer or through data that is ready to be used in Arc-View 3.x. The online GIS approach is the corporate enterprise response to getting an organization's data online rapidly. The originator assumes that the user of online GIS has the ability to create maps and manipulate a GIS. The issues of mapping in this case are the responsibility of the user; the data and some GIS functionality are provided through the online GIS application.

The different programming tools used for Internet cartography include links and list-box choices that function as query routes to the geographic attribute database. The GIS tools use programming to hide routes and query commands. For example, the list box makes a close-up view available for any image simply by clicking on the map. The cartographic mapping application is automated and provides features such as zoom and the creation layers relevant to the map and its intended message.

The online GIS technique provides the user control over final display and analysis options. Internet cartography is analogous to a custom map set. Maps are custom and pre-made, the elements of cartography are decided, leaving the user with a map that is usable for unique queries provided by the user.

The cartographer is needed to establish criteria that makes GIS more intuitive for the user and eliminates past confusion often associated with Online GIS. READ is an example of a system based on click events that route the user to different links to convey information at various geographic scales. The map is automatically projected from geographic coordinates without actually choosing a projection. The geographic data does not need to be manipulated by the user, but becomes readily available in order to provide information for display. The burden of processing the information into meaningful display is the responsibility of the cartographer/programmer. The relationship between cartographer, programmer, and user produces quicker mapping and broadens the constituents utilizing data in map and database form.

Chapter 3

PROGRAMMING CONSIDERATIONS

Different personal computer operating systems and network server operating systems create compatibility problems for Internet Cartography, and Internet programmer in general. There are two major Internet browser software vendors in the world, Microsoft and Netscape. Both provide graphical user interfaces for personal computer users accessing the Internet. Programmers must consider the subtle differences between these two browsers when designing map applications. They must, for example, understand the individual network processes involved in each application. Problems encountered in disseminating to web pages or scripting components such as Microsoft Active X or Netscape JavaScript elements, typify these programming difficulties.

Browser software available on personal computers maximizes use of the central processor and Random Access Memory (RAM) resources. Devising and sending smaller module packages of script is one way of making use of the client's RAM. Scripting applications require the programming of smaller scripted applications that are read into the client's browser software. Assembling maps customized to users' requests and returning these new, smaller input packages to network servers, occurs concurrently with interactive selection of data elements headed for display. The network server employed for the mapping, processes these input packages, using the resources of the client's personal computer for some of the computing power necessary. This enables faster data retrieval and web browsing. The differences between using Java Script and Active X on the client or server side for map generation methods are based on differing proportions of user-side and server-side computing.

The document object model is a suggested standard in scripting, which provides a versatile view of a document's contents as well as allowing the data to be manipulated by other routines and reused with other document object models. The most popular Internet connectivity software, Internet Explorer (Microsoft), uses the Active X and JavaScript languages. Netscape Navigator software, the next most widely used Internet browser, uses JavaScript language exclusively. The Active X method only works when users accept the installation package requiring third party verification insuring the code as virus free. Required validation and compatibility are drawbacks using this approach, and Active X components can be programmed to be malicious. Current authorization signature protocols for Active X have inhibited its widespread use. The programmer attempts to avoid the hassles of authorization signatures and may prefer scripting with JavaScript and using widely accepted web programming methods instead of using Active X. In addition, programmers prefer the JavaScript method because of its resistance to writing viruses. Both Internet cartography and online GIS require the user's computer to do some of the work.

Some geographic software must be installed on the users personal computer to load these required routines. A programmer builds an Active X program with dynamic library links (.dll), and distributes them to the users' computers through the Internet. Arc-Explorer produced specifically for such data browsing and data assembling tasks, is free from ESRI via the Internet. Arc-Explorer software contains basic GIS tools necessary for downloading data, organized in Arc-Explorer project files or basic shapefiles.

Proprietary information such as that used by READ requires secure computing protocols on the side of the server. Network servers are considered secure repositories for programmed applications. Before completing programming for the Network server, processed information must be sent in parcels prior to its insertion in a Web Page. The user-side computer sends requests to the Network server, using the same process, before writing data elements to a Web Page. As programming is completed, the Network server transfers newly created images that incorporate all of data elements at the same time reducing the amount of data that must be returned to that particular users' personal computer. Furthermore, this process does not require exposure, during Internet transmission, of underlying attribute data.

These applications must be programmed to execute sequentially in steps that mirror activities taken by a GIS specialist building applications to display the same database elements the ICSA includes. Programming objects necessary for assembling a map depicting database elements identified in users' requests. The following instructions for using GIS software to produce a map such as ArcView 3.X (AV3.X) will illustrate steps the program accomplishes for each request. These instructions also illustrate how base map Shapefiles for READ were constructed. The process of generating a map customized for an individual Internet user to display information that he or she has requested follows the following six steps:

ArcView 3.x	*ICSA*
Add data to the view	Add data to the "Map Control"
Select area of interest from geographic file.	Use subset of existing geographic file, either programmatically or with GIS.
Use the join method to add and join the attribute table to the geographic file based attribute table.	In the ICSA, use one of the various Map Objects methods to add external database tables, and relate or join to the attributes of the geographic file.
Use the legend editor to set legend categories.	Use one of the Map Objects methods for rendering the themes programmatically.
Apply a map projection to the data using the "view property" projection tool.	Establish a projection for the map control, or convert all layers added to the map control to the desired projection.
Print the map product or export it to an image type file.	Call a function to output the contents of the map control to an image type file ready for use in a web page or elsewhere.

After completion of these six steps, the resulting information moves from the workspace to the "layout", or paper view. Once the Layout contains the data, adding the final effects of cartography completes the map. Throughout this process the ICSA programmer must ensure quality by applying quality control measures in keeping with traditional cartographers and their applications to produce quality cartographic outputs.

Chapter 4

CONNECTING ICSA TO READ: CONSTRUCTION PHASE

A person using a computer interfaces with that machine by viewing a series of displays on the computer's monitor. Changes in the display are made in response to selections made by the user. These displays taken together comprise the computers graphical user interface, or GUI. The GUI developed for the READ ICSA allows a user to select data from READ and view it in maps displayed on their monitor screen. This ICSA enables users with varying degrees of computing sophistication, having little or no GIS or computing experience, to optimize time spent accessing, and familiarizing themselves with demographic and economic information in READ. Furthermore, the READ ICSA represents an improvement in cartography, by providing maps by the Internet through a process that takes advantage of cartographers' training and experience, in making meaningful decisions.

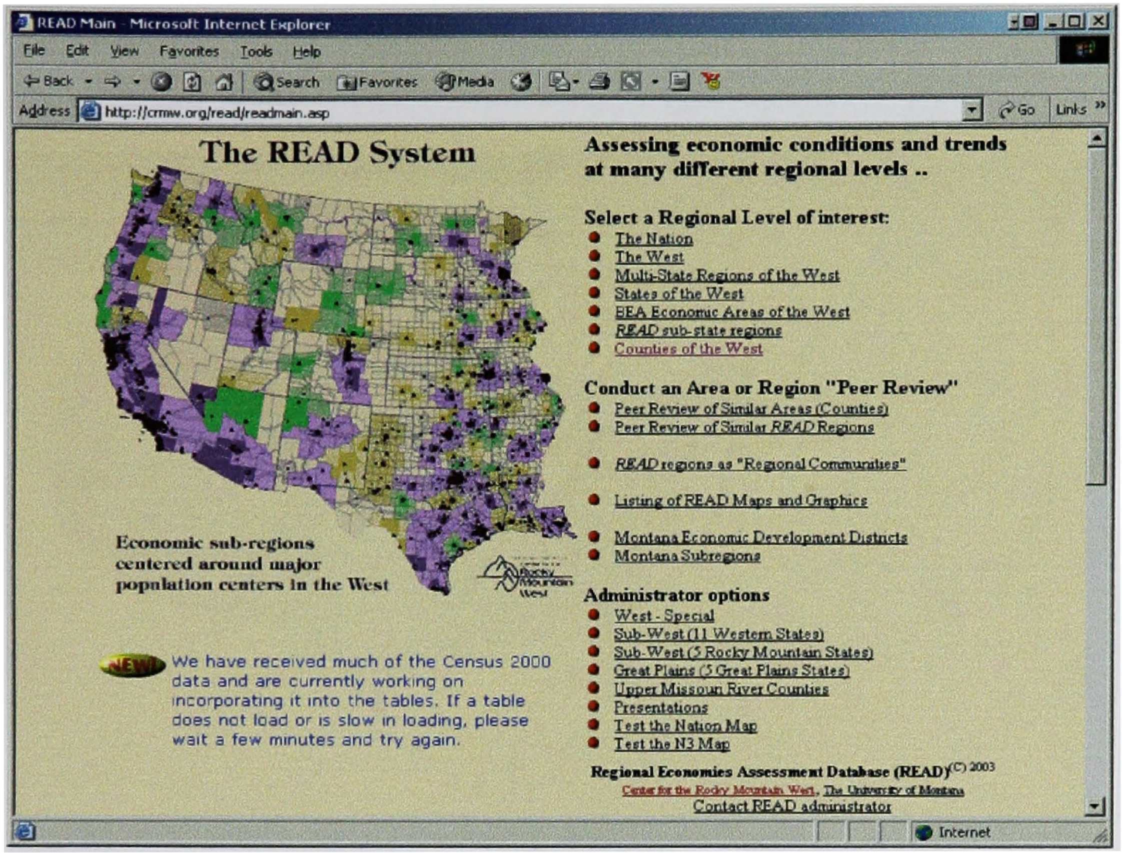
The ICSA/READ Connection

The web pages introducing the READ portion of the ICSA describe how the regional data is organized and accessed. Information for the entire nation is organized following the spatial data organization scheme established by the U.S. Census Bureau.

Creating an application to deliver information from a web site requires programming that anticipates and plans for users' needs. Planing for these requests, or queries is accomplished by analyzing the database in terms of its organizational scheme and its component parts. The READ ICSAs organizational scheme is based upon geographic principles.

The READ ICSA enables users to choose individual paths through the information, using capabilities that are standard features of Internet browsers. User paths follow a linear progression, through maps of the western USA, to maps of sub-regions, and then states or counties. Provision of READ information in multiple scales allows users to customize outputs as needed for individual study areas. Local and regional economic information is made more meaningful in this manner, as local or regional study area data is readily comparable to regional or national scale data. The Read ICSA has been programmed to facilitate these comparisons. This programming includes limitations on accepting user generated queries. This maximizes the rate at which the ICSA delivers READ information (Figure 1).

The READ system organization is based on a hierarchy of geographic scales in decreasing order beginning with the link at the top right in Figure 1 "The Nation" for national data. This ordering focuses user attention on national economic conditions and trends first, and then facilitates exploring data for smaller geographical areas. This format provides users with


READ Main - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://crmw.org/read/readmain.asp
Go Links
Internet
NEW!
Rocky Mountain West

The READ System

Economic sub-regions centered around major population centers in the West

We have received much of the Census 2000 data and are currently working on incorporating it into the tables. If a table does not load or is slow in loading, please wait a few minutes and try again.

Assessing economic conditions and trends at many different regional levels ..

Select a Regional Level of interest:
- The Nation
- The West
- Multi-State Regions of the West
- States of the West
- BEA Economic Areas of the West
- *READ* sub-state regions
- Counties of the West

Conduct an Area or Region "Peer Review"
- Peer Review of Similar Areas (Counties)
- Peer Review of Similar *READ* Regions
- *READ* regions as "Regional Communities"
- Listing of READ Maps and Graphics
- Montana Economic Development Districts
- Montana Subregions

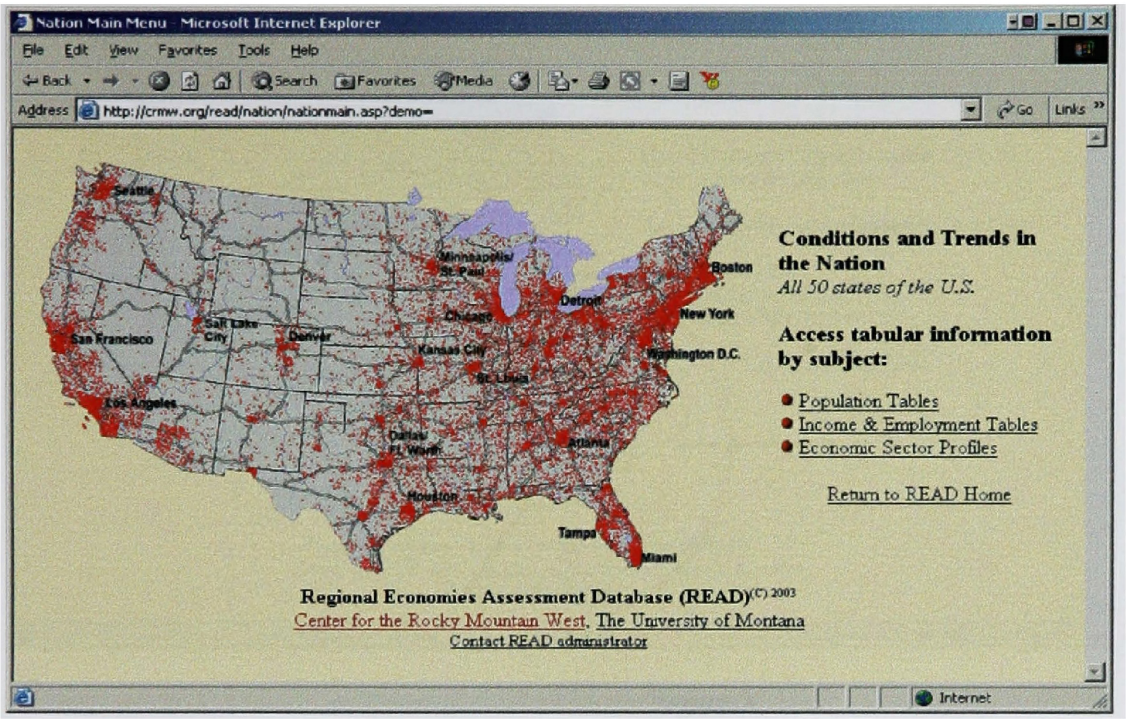
Administrator options
- West - Special
- Sub-West (11 Western States)
- Sub-West (5 Rocky Mountain States)
- Great Plains (5 Great Plains States)
- Upper Missouri River Counties
- Presentations
- Test the Nation Map
- Test the N3 Map

Regional Economies Assessment Database (READ)(C) 2003
Center for the Rocky Mountain West, The University of Montana
Contact READ administrator

Figure 1

The starting point of the READ ICSA, for assessing economic conditions and trends in the western USA, and exploring comparisons between regional and national data.

an economic framework for their individual queries. If the user does not have a specific economic query when entering, the READ system provides thoughtful arrangement of links to quickly communicate meaningful comparisons of national scale data and regional scale data. For example, when a user selects the link for nation data on the screen shown in Figure 1, the ICSA displays the screen shown in Figure 2, including a map of population distribution data collected by the 1990 U.S. Census. Then the user chooses population tables, income/employment tables, or economic sector profile tables that display data for the entire USA.



Conditions and Trends in the Nation
All 50 states of the U.S.

Access tabular information by subject:

- Population Tables
- Income & Employment Tables
- Economic Sector Profiles

Return to READ Home

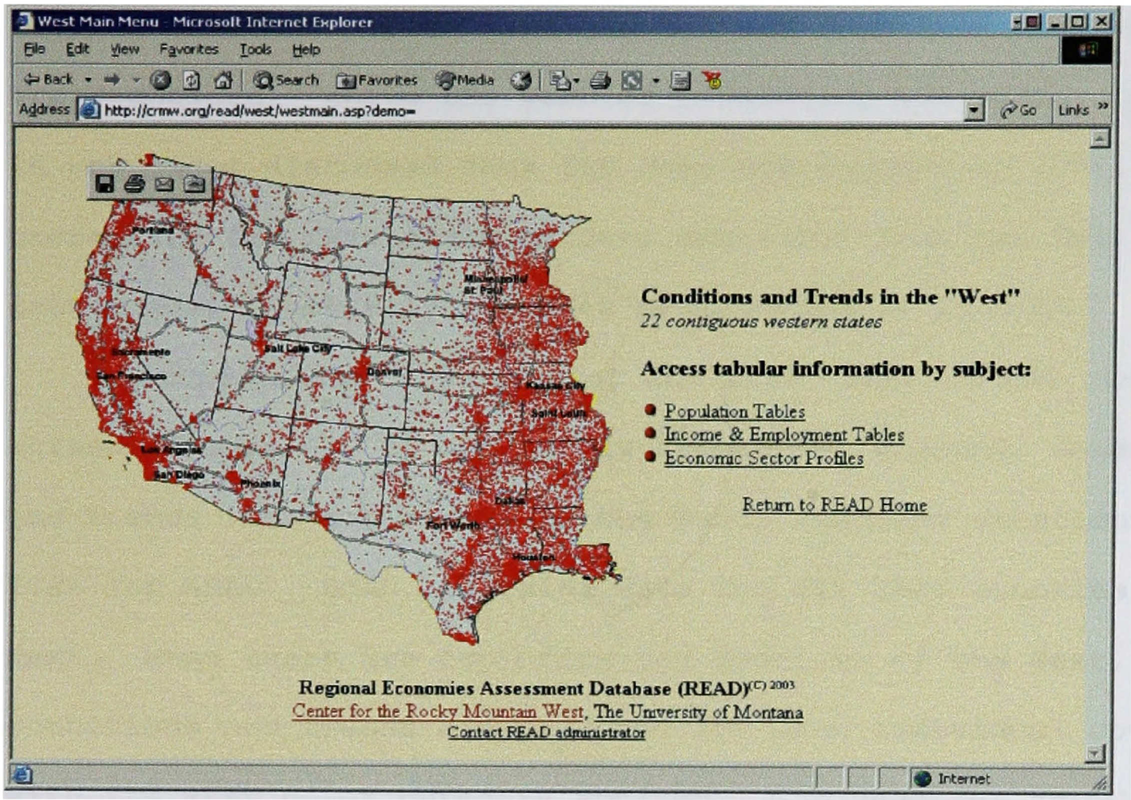
Regional Economies Assessment Database (READ)(C) 2003
Center for the Rocky Mountain West, The University of Montana
Contact READ administrator

Figure 2

After selecting "The Nation" on the screen shown in Figure1, the READ ICSA displays 1990 Census data for the U.S., and offers to display more detailed information in population tables, income and employment tables, or economic sector profiles.

Based on their selection of geographic scale for the data, users are provided further refinements in the maps on the system. After selecting "The West" on the starting screen (Figure 1), users see a population density map of the western USA. The same links provided for national scale data are also provided here, but limited to conditions and trends in the West (Figure 3).

Below "The Nation" and "The West" selection on the starting screen (Figure 1), options are provided for viewing maps of economic data for multi-state regions of the West,


West Main Menu - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://crmw.org/read/west/westmain.asp?demo=
Go Links
Portland
Minneapolis St Paul
Sacramento
San Francisco
Salt Lake City
Denver
Kansas City
Saint Louis
Los Angeles
San Diego
Phoenix
Dallas
Fort Worth
Houston
Internet

Conditions and Trends in the "West"
22 contiguous western states

Access tabular information by subject:

- Population Tables
- Income & Employment Tables
- Economic Sector Profiles

Return to READ Home

Regional Economies Assessment Database (READ)(C) 2003
Center for the Rocky Mountain West, The University of Montana
Contact READ administrator

Figure 3

Selecting "The West" from the first screen of the READ ICSA (shown in Figure 1.), allows its users access to information about conditions and trends in the western USA, by presenting this display.

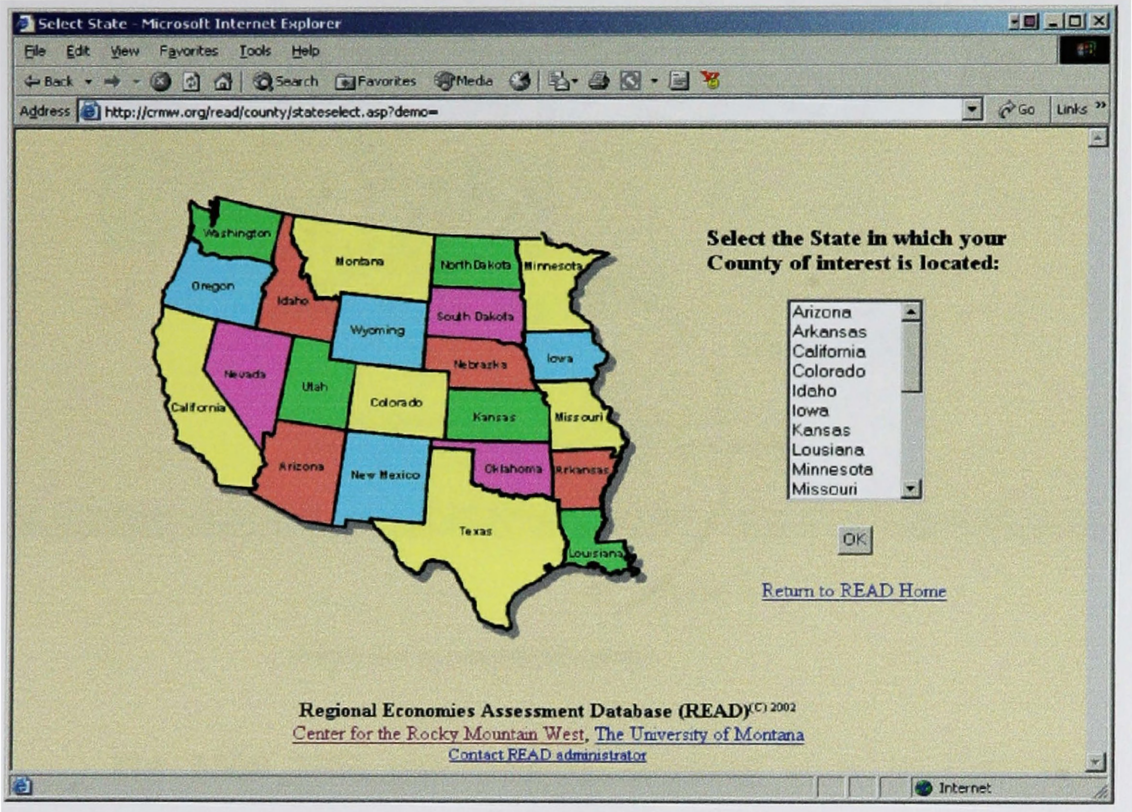
each state individually, Bureau of Economic Analysis regions, and READ sub-state regions. All levels of query in READ provide users with maps showing the geographic scope of the subsequent economic tables. The work of presenting and linking all these maps would not warrant the creation of an ICSA, as these maps could be created using existing ArcView software.

The three screen displays shown in Figures 1-3 do not require an ICSA for their creation. The READ ICSA has been programmed to create these three maps only when the data they are based upon has been updated. That is, the ICSA program checks whether the map file exists, and if it does, then the existing file is used instead of making a new map each time the web pages are requested. While the work of presenting the maps, included in web pages discussed thus far does not require an ICSA, presenting the next scale of data available from the READ, the county scale, actually requires programming of an ICSA.

Including the "Counties of the West" link on the starting screen (Figure 1) to accurately depict the economic conditions and trends for each county in the West, required creation of the ICSA for READ. READ, contains data for all 1500 counties of the West. When users are exploring the counties of the West, conditions and trends are provided for each individual county with its READ color category applied. The need for an ICSA becomes apparent, as one considers that the cartographer is responsible for creating 1500 individual images.

The READ ICSA guides users to selecting counties by first selecting states. The user selects a state, for viewing a county, by highlighting the state in which the county exists and pressing the "ok" button. Next, the user selects the specific county to be examined. At this point, the ICSA examines data files to determine if the map requested already exists, and if it does not, the ICSA builds and delivers the map to the user. This is all transparent to the user as the ICSA provides the map from either source, efficiently retrieving or creating all the maps requested. Maps of counties are not created at all until the first request is made. Then, they are saved and reused for any subsequent requests.

After selecting "Counties of the West" from the starting screen (Figure 1), the following screen display shown in Figure 4, enables users to select a state. After selecting the state, the system displays a map of the state showing its counties (Figure 5).



Select the State in which your County of interest is located:

Arizona
Arkansas
California
Colorado
Idaho
Iowa
Kansas
Lousiana
Minnesota
Missouri

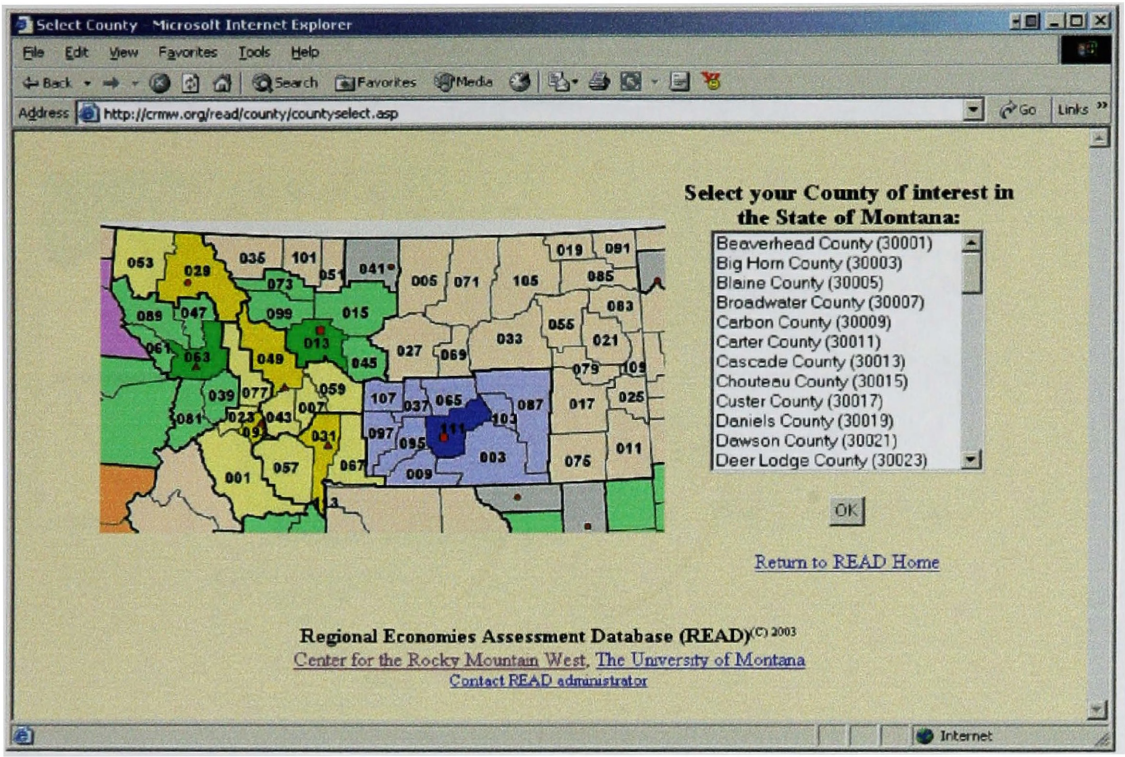
Return to READ Home

Regional Economies Assessment Database (READ)(C) 2002
Center for the Rocky Mountain West, The University of Montana
Contact READ administrator

Figure 4

To examine economic conditions and trends in a specific county users first select which state the county is in.

After a user selects the county of interest, the READ ICSA determines if it is making a new map or using a map already on file, A significant feature is that it frees cartographers from creating map images that will not be used. The next screen seen shows the individual county selected, as shown in Figure 6.



Select your County of interest in the State of Montana:

Beaverhead County (30001)
Big Horn County (30003)
Blaine County (30005)
Broadwater County (30007)
Carbon County (30009)
Carter County (30011)
Cascade County (30013)
Chouteau County (30015)
Custer County (30017)
Daniels County (30019)
Dawson County (30021)
Deer Lodge County (30023)

Return to READ Home

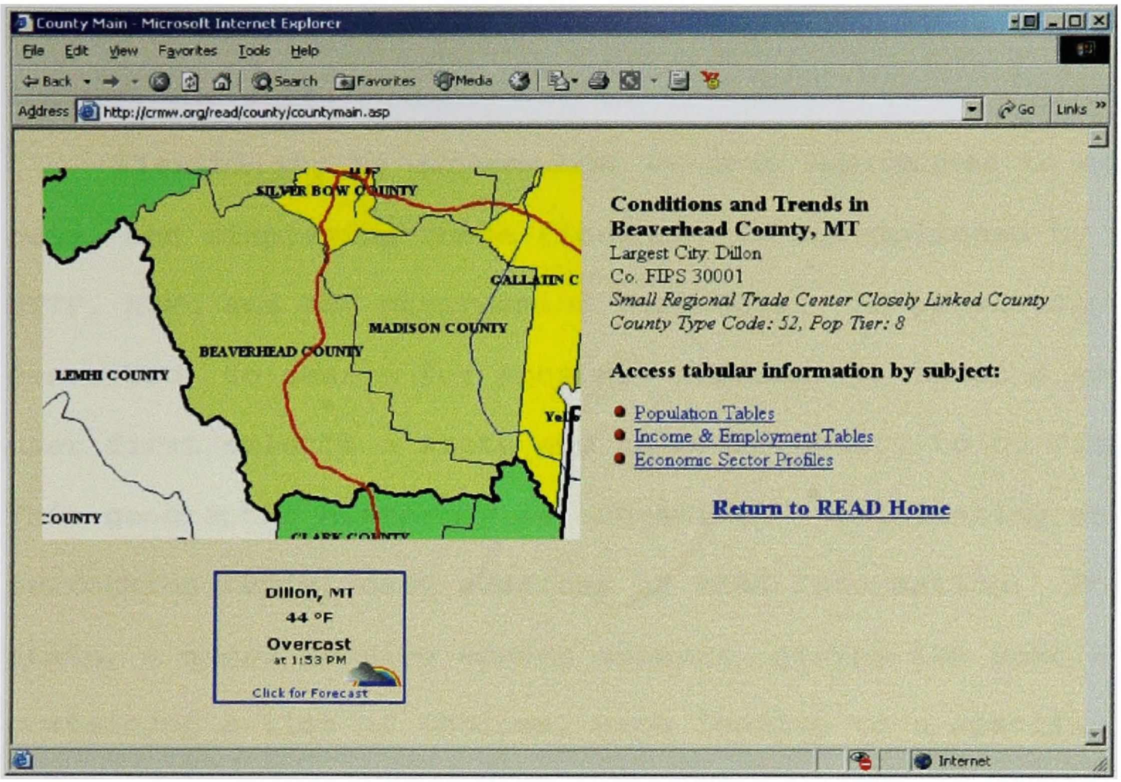
Regional Economies Assessment Database (READ)(C) 2003
Center for the Rocky Mountain West, The University of Montana
Contact READ administrator

Figure 5

A state and its component counties.

The READ ICSA was created bearing in mind that most map users have more experience with paper maps than with maps displayed on the Internet. Electronic presentation of economic and geographic information requires programmers to anticipate users' preferences for each map element, such as map coverage area or topographic contour interval. These attributes are predetermined, physical attributes of paper maps but they are variables in the creation and presentation of maps electronically.

The READ ICSA offers users several tools for accessing and displaying data, including navigation and action handles, that are standard features of Internet browsers such as Microsoft


County Main - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://crmw.org/read/county/countymain.asp
Go Links
SILVER BOW COUNTY
GALLATIN C
MADISON COUNTY
BEAVERHEAD COUNTY
LEMHI COUNTY
COUNTY
CLARK COUNTY
Yel
Internet

Conditions and Trends in Beaverhead County, MT
Largest City: Dillon
Co. FIPS 30001
Small Regional Trade Center Closely Linked County
County Type Code: 52, Pop Tier: 8

Access tabular information by subject:

- Population Tables
- Income & Employment Tables
- Economic Sector Profiles

Return to READ Home

Dillon, MT
44 °F
Overcast
at 1:53 PM
Click for Forecast

Figure 6

A county within the state chosen on the screen shown in Figure 5. The READ ICSA created the image, the map requested remains in a file on the READ server waiting for another user to pick the same county.

Internet Explorer and Netscape Navigator, that comply with the World Wide Web Consortium Document Object Model (DOM). Navigation handles include labels, icons, buttons, and image regions where users click to invoke actions, for example to open a link. The handle's effectiveness can be enhanced, as is commonly seen when different colored text is used to identify a link, or a magnifying glass is used to zoom in on a sub-region.

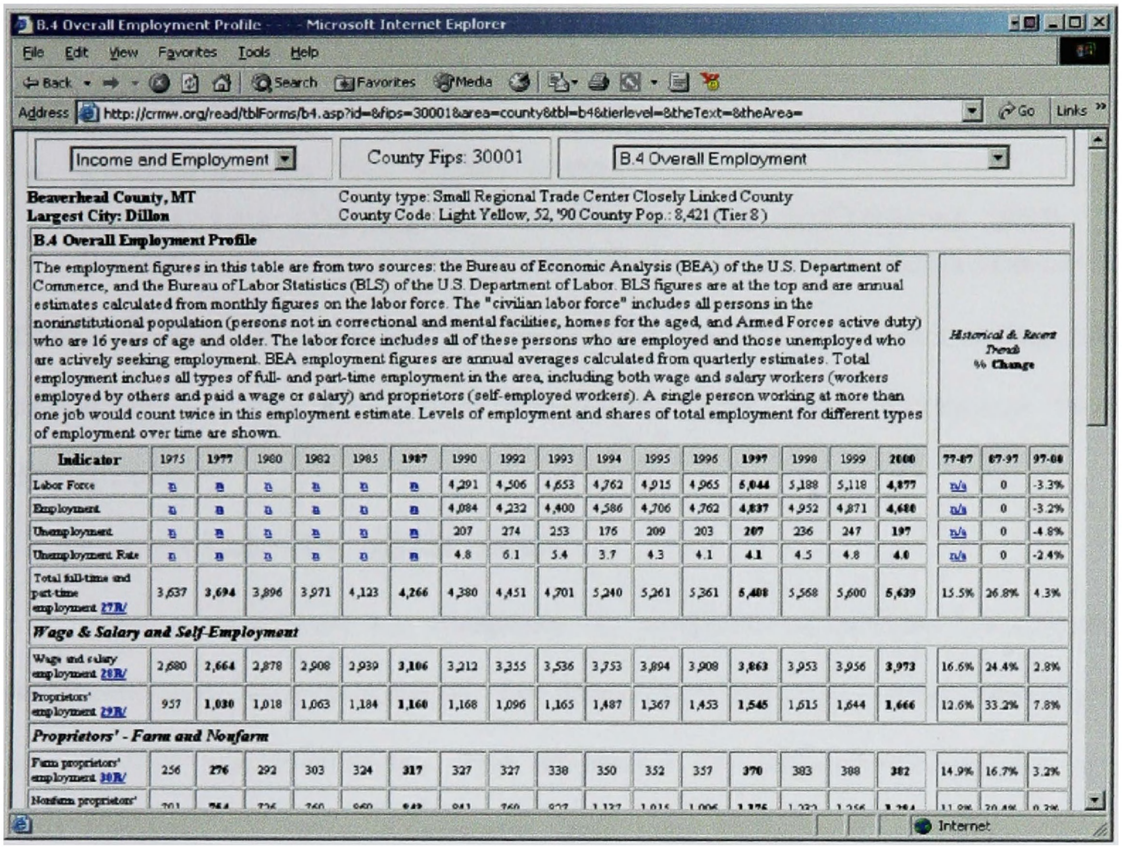
The READ ICSA presents action handles, in menus of links and list boxes, facilitate users' access to the READ information. Searches in READ are accomplished using mouse clicks and keystrokes. Each data element within the organizational scheme can be accessed by multiple links, providing numerous routes through the data.

Flexibility to accommodate various approaches to accessing data, and displaying these findings, is accomplished by using HTTP, HTML and ASP programming languages. As previously described, to search for economic information about a county, the user first selects a state and then the county to be displayed. This geographic hierarchy is convenient, facilitating users performing rapid, easy searches of READ information. For each state, a query preview screen appears, giving the user a list box containing a list of choices, each leading to a specific county within the chosen state (Figure 5). Then, expanded menu and link routes provide multiple options for economic data selection, and orient and display maps, as shown in Figure 6.

The results of their queries are then displayed to users, as shown in Figure 7, in a formatted table loaded with more than 25 years of data, programmed to provide accurate comparisons in selected categories. Economist Larry Swanson has provided formulas for extrapolating data, to be included in these tables, that are missing from the raw data from which READ is assembled. The table returned with information about topic(s) selected also provides additional information, such as text descriptions of how the data has been derived, and how those who gather the data allow for omissions in that information.

Automated Cartographic Database Design: Internet Implementation

After envisioning how READ information could be integrated


B.4 Overall Employment Profile - - - - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://crmw.org/read/tblForms/b4.asp?id=&fips=30001&area=county&tbl=b4&tierlevel=&theText=&theArea=
Go Links
Internet

Income and Employment | County Fips: 30001 | B.4 Overall Employment

Beaverhead County, MT
Largest City: Dillon

County type: Small Regional Trade Center Closely Linked County
County Code: Light Yellow, 52, '90 County Pop.: 8,421 (Tier 8)

B.4 Overall Employment Profile

The employment figures in this table are from two sources: the Bureau of Economic Analysis (BEA) of the U.S. Department of Commerce, and the Bureau of Labor Statistics (BLS) of the U.S. Department of Labor. BLS figures are at the top and are annual estimates calculated from monthly figures on the labor force. The "civilian labor force" includes all persons in the noninstitutional population (persons not in correctional and mental facilities, homes for the aged, and Armed Forces active duty) who are 16 years of age and older. The labor force includes all of these persons who are employed and those unemployed who are actively seeking employment. BEA employment figures are annual averages calculated from quarterly estimates. Total employment inclues all types of full- and part-time employment in the area, including both wage and salary workers (workers employed by others and paid a wage or salary) and proprietors (self-employed workers). A single person working at more than one job would count twice in this employment estimate. Levels of employment and shares of total employment for different types of employment over time are shown.

Historical & Recent Trend
% Change

Indicator	1975	1977	1980	1982	1985	1987	1990	1992	1993	1994	1995	1996	1997	1998	1999	2000	77-87	67-97	97-00
Labor Force	n	n	n	n	n	n	4,291	4,506	4,653	4,762	4,915	4,965	5,044	5,188	5,118	4,877	n/a	0	-3.3%
Employment	n	n	n	n	n	n	4,084	4,232	4,400	4,586	4,706	4,762	4,837	4,952	4,871	4,680	n/a	0	-3.2%
Unemployment	n	n	n	n	n	n	207	274	253	176	209	203	207	236	247	197	n/a	0	-4.8%
Unemployment Rate	n	n	n	n	n	n	4.8	6.1	5.4	3.7	4.3	4.1	4.1	4.5	4.8	4.0	n/a	0	-2.4%
Total full-time and part-time employment 27R/	3,637	3,694	3,896	3,971	4,123	4,266	4,380	4,451	4,701	5,240	5,261	5,361	5,408	5,568	5,600	5,639	15.5%	26.8%	4.3%
Wage & Salary and Self-Employment																			
Wage and salary employment 28R/	2,680	2,664	2,878	2,908	2,939	3,106	3,212	3,355	3,536	3,753	3,894	3,908	3,863	3,953	3,956	3,973	16.6%	24.4%	2.8%
Proprietors' employment 29R/	957	1,030	1,018	1,063	1,184	1,160	1,168	1,096	1,165	1,487	1,367	1,453	1,545	1,615	1,644	1,666	12.6%	33.2%	7.8%
Proprietors' - Farm and Nonfarm																			
Farm proprietors' employment 30R/	256	276	292	303	324	317	327	327	338	350	352	357	370	383	388	382	14.9%	16.7%	3.2%
Nonfarm proprietors'	701	754	726	760	860	843	841	769	827	1,137	1,015	1,096	1,175	1,232	1,256	1,284	11.9%	20.4%	9.2%

Figure 7

READ information for a county is displayed on web pages that include both navigation handles and action handles.

into an Internet cartographic application, the next step was implementation. Programming the application requires integrating several GIS software applications, including ArcView 3.2 (AV32), Map Objects 2.0 (MO2), Internet Map Server 1.2 (IMS12), and Microsoft Visual Basic 6.0 (VB6). Programming interfaces between these applications presented both challenges and opportunities through exploration of cutting edge electronic geographical capabilities. For these applications to be useful, these interfaces must function fully. Successfully integrating READ into ICSA involved three distinct challenges. The following sections discuss these challenges, and their solutions, in the following order.

- Envisioning the READ ICSA;
- Preparing the database using GIS software; and
- Programming the Internet Cartographic Application.

Then, basic programming steps are outlined, including the creation of cartographic sub-functions, and software testing is described.

Envisioning the READ ICSA

As discussed in Chapter 3, object oriented programming methodology was integral to creation of the ICSA cartographic technique advancement. The objects in object-oriented programming are groups of data elements and the inherent methods to interrelate them. Objects of traditional maps, such as map projection, legend, layout design, or other graphical items, are selected in VB6 and MO2 programming environments then directed to generate maps to display on the Internet. The cartographer then uses pre-programmed methods to manipulate map objects and other similar objects to create a unique product. In the Visual Basic programming environment, this object-oriented methodology enables users to interact with the program through a GUI. The GUI environment is easy to use presenting icons and drop down menus as interface commands that reduces time spent writing code.

A mapping window was added to extend the programming functionality of VB6 for the ICSA. The window is an "Active X" component, which is provided through the Map Objects (MO) software. The addition of the control in Visual Basic facilitates the inclusion of geographic data in the Visual Basic project. Visual Basic, when it is new from Microsoft, does not contain the ability to use geographic type data. The ability to include geographic data in the Visual Basic programming environment is achieved by the installation of MO from ESRI.

The versatility of making maps this way can be illuminated by a comparison of mapping objectives. While the ICSA produces maps using programming steps that emulate ArcView 3.2 operations, ICSA users create map legends without interacting directly with GIS software. Instead, using features programmed into the ICSA, they create a legend as an object, which is a sub-function within the ICSA. Requests for information, and resulting responses, are exchanged by means of the map control.

Online GIS for example, using ArcView software, differs from using an ICSA in that users do not receive raw geographic data at any time. The user does not need to receive any new Active X components or Java Script enabled web items to assemble and view a map from READ. ICSA users browse a constellation of custom economic maps through hierarchical levels of analysis. The program is prepared for requests which users implement by clicking on a menu item, or part of a map, depending on the web interface design. Requests are processed on the server, then returned to the user, along with more information about the topic in graphic format. Throughout this process, the map control remains linked to the tabular data represented, and finally, the user receives a screen image containing requested information. Users receive requested data in the form of HTML page(s) with map image(s) inserted. Current HTML browsers offer design possibilities for the Internet cartographer in the form of graphic backgrounds, hyperlinks to analysis of the topic, hyperlinks to databases across the World Wide Web, as well as many other, options.

Frequency categories needed for mapping population and economic data, and for creating legends with ISCA, are inherited from the READ system's mapping scheme. This results in a map showing 1500 counties in the West, either together in regional categories or individually depending on the users request (Figure 1). The READ organizational hierarchy is necessary for programming such things as frequency categories and color schemes. The READ scheme structure uses information that shows a relationship and association with each object. The scheme also outlines definitions used in conjunction with the map legend.

The methods are object-oriented operations written using VB6 language for the objects to be programmed. A method is chosen for an object according to the scheme presented by the database. For example, with the READ system, twelve colors are used for the sub-classification of regional economies in twelve categories. Colors of the READ categories are controlled in the maps, by using Map Objects' value map renderer to set unique values to specific colors. The selection of color values is done manually. The value map renderer is a programming object whose methods and properties can be used by the cartographer to enhance displays of economic information in the READ system. In the READ system of mapping, a column in the database has fields with each unique value for each category to be used. Knowing that the READ system incorporates 12 unique sub-categories allows controls to be established for each category's unique set of values.

Preparing the Database Using GIS Software

ArcView 3.2 desktop GIS software by ESRI is used for preparing the spatial data needed by ICSA for dynamic Internet mapping. This software was enhanced by using scripts and by programming application extensions as necessary to establish connections with ICSA. Basic database preparation in AV3.2 sufficed for purposes of the ICSA. This included data processing that was needed for both the economic and geographic databases used in READ. ArcView 3.2 is able to select geographic features that could be depicted by points, lines, or polygons. These, in turn, could be linked to economic attributes that are recorded in a variety of tables within the READ system.

Steps required to accomplish data preparation for Internet mapping are performed at many levels of the programming process. Each geographic unit must be selected, then linked or joined to the attribute data for display while using the READ color scheme and classification categories.

ArcView 3.2 can map database files selected from the READ system for the ICSA. Using a function within AV 3.2 known as "SQL connect," the cartographer connects to the READ system and selects the columns of data to be mapped. To use the "SQL connect" function, the cartographer needs to first establish the software drivers allowing the database connection. To achieve this, the appropriate Open Database Connectivity (ODBC) drivers need to be installed and running on the server and the personal computer with the AV3.2 software. This is accomplished in Microsoft Windows control panel using the ODBC32 manager. The READ database itself is a Fox Pro database. ICSA accesses this database using the same ODBC function in Windows control panel, once the system data sources have been specified and the appropriate drivers for the database connectivity have been installed. ArcView 3.2 automatically detects data source names from those established as data system names in the ODBC32 upon opening the software. The significance of using ODBC for the connection to the READ system is that this enables AV3.2 to access the database and create files without outside software.

The READ system contains more information than can be used at any one time. Desired information is extracted from READ by means of Standard Query Language (SQL) statements. Selected data can be mapped using only those fields, and one field that to joins them to geographic features, map generation requires a common data field for joining economic attributes to geographic features. The data for location of the counties and the other spatial elements is assembled using ESRI's shapefile format. The READ system has 1500 counties in the West, contained in 155 sub-regions. Within these sub-regions, economic variables are ranked and referenced according to population, size, and other criteria. Each county in a sub-region is also identified as a "core" or a "closely linked" county in terms of population of the largest place in the county. Providing a refined look at conditions and trends in the West based on the data obtained from the many contributing sources of data is accomplished by READ.

Cartography for READ progressed from paper-printed maps through online static images to the dynamic data type that the ICSA provides. Throughout this evolution, map legend categories became more elaborate as did their implementation in the ICSA. More complex dynamic map creation requires cartographers to perform mapping routines and layout for up-to-date maps, and the ICSA accomplishes this automatically. Once the ICSA has been programmed, the cartographer updates the economic attribute files on the web-server, then these updates are incorporated into outputs as appropriate.

Programming the Internet Cartographic Software Application

Establishing programming objective was the first essential step in creating the ICSA. These objectives included the display, function, and data security design elements of the program which are ICSA's fundamental attributes. The intention to produce economic thematic maps for the Internet required the creation of another component that uses the GIS object oriented programming methods and properties to create map image layouts. Using the GIS objects for the creation of ICSA allows a programmer to specify what GIS data is necessary for the map image. Other considerations are that while the addition of technological mapping enhancements continually update the site, familiarity with the site should only increase as procedures to retrieve data from it do not change. Standardization of menus and hyperlinks facilitates access to increasingly detailed information. The ICSA seamlessly integrates READ information making use of READ's economic themes display standards. Finally, the ICSA incorporates important safeguards into data storage and delivery routines and uses a security strategy to effectively protect both the ICSA program itself, and the READ information on a server.

Important differences exist between web display options of users' Internet browser software, such as found on Active X and Java enabled pages, as opposed to those generated using HTML and ASP data security strategies. Online GIS applications have traditionally required that users have basic GIS software skills involving the use of tools for transforming data into maps. For example, the basic GIS tools can be in the form of small executable applications sent to the users computer from the map server. Other GIS tools are those that are programmed through a variety of programming languages. For example, Microsoft uses VB6 and compiles it into the Active X format. The process encapsulates various portions of a GIS program so that they can be placed onto a user's machine as a complete application of tools.

Firstly, because only Microsoft Internet Explorer browser software accepts and runs Active X components. Proprietary data on a server is exposed in order for it to be processed on the client's computer. Secondly, Netscape Navigator Internet browser software does not run Active X components at all.

Policies established for responding to ICSA user requests for proprietary data from servers depends upon the objectives of database administrators. Aggregates of data already processed and analyzed for the user audience, such as exists in READ, are maintained and protected on a particular server. When data is not processed beyond basic geographic data transformations, that data is disseminated and undergoes further analysis by the GIS user, and the Active X and Java enabled pages are presented on users machines.

Processing requests and responses is the central process of map generation using an ICSA. The initial web page seen in READ, with basic HTML structure, is comprised of images, hyperlinks, and customized backgrounds. Active Server Pages (ASP) and HTML are used to produce and deliver images in the ICSA. User requests are narrowed using exchanges in HTML, and output is made in response to map requests by ICSA as generated in HTML. ASP is used for more explicit requests including direct database queries, script executions, and tabular displays of READ contents. ASP is used with embedded Visual Basic Script and Active Data Objects (ADO) using SQL to connect and query the READ dynamically. To produce maps on a server for Internet distribution without the component applications of Active X or Java, ICSA creates an image for the request on a web site. The map server is the only active element processing the requests. Each is responded to with an image file that can be described as a "snapshot" of some databases contents. Then the image file is inserted into the web page built by the ICSA.

BASIC PROGRAMMING STEPS:

The following outline describes manipulations necessary to incorporate GIS components into the Visual Basic programming environment in writing a program to receive Internet requests and respond by delivering maps.

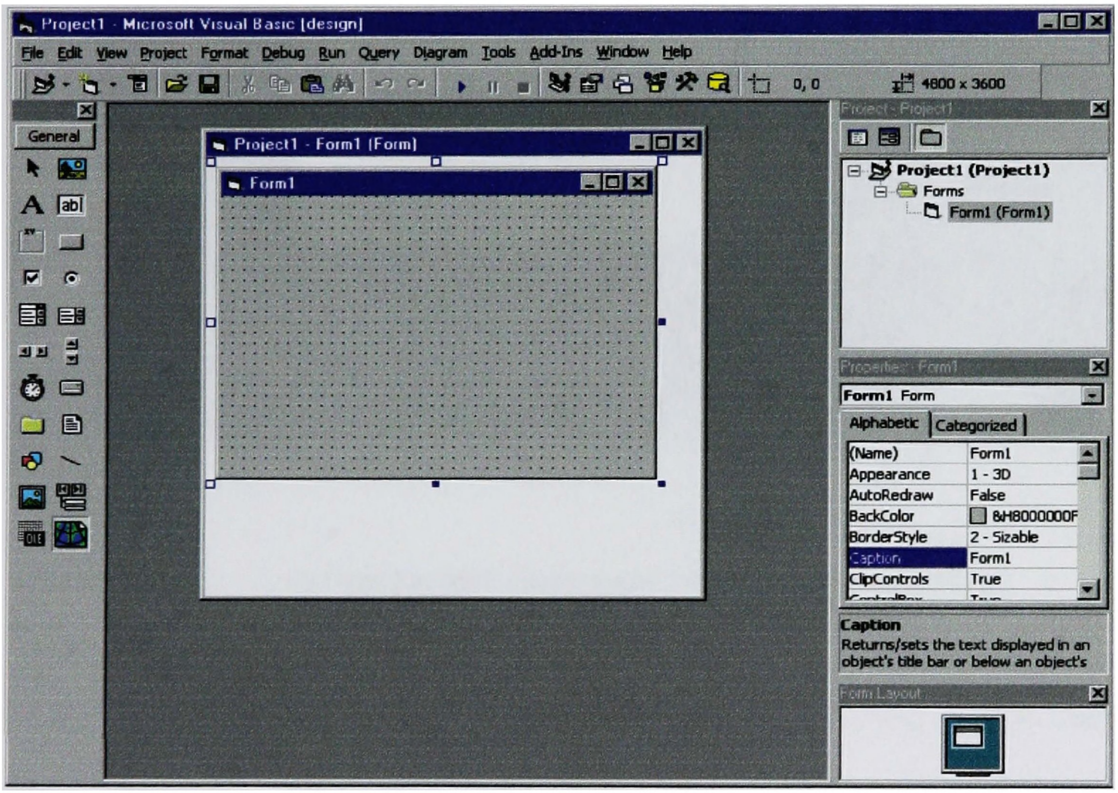
- Open Visual Basic 6.0 as shown in (Figure 8). Open the components menu item and choose to add Map Objects 2.0 and the Web-link of MOIMS.

- Open a form, and select the map control from the left menu and place it in the form (Figure 9). Save the form and save the project at this point.

- Place a Weblink in the form. This will provide communication with the esrimap.dll installed on the map server.

- Open the Visual Basic code editor, specify the option explicit (recommended to help in debugging), then declare (DIM) relevant variables, as required using the "DIM" character string (Figure 10).

The ESRI Internet Map Server (IMS) tool can be used to distribute the READ ICSA. An Active X weblink is embedded into the ICSA as a control. The weblink, which is installed from IMS, works as "middle-ware" to link the VB program as enhanced with the Map Objects (MO) to the esrimap.dll MOIMS. The MO and MOIMS software provide for communication between READ ICSA and the HTML and ASP outputs. The MOIMS is composed of several parts, each registered by the ICSA in response to requests from the user interface. In order to run, the ICSA needs a .dll file, such as esrimap.dll, and an Active X control. The .dll file is used to handle requests from users, parse them, and pass them to servers running the ICSA VB/MO program. The weblink provides a passage between the map, the data requests, and the responses on a personal computer and the esrimap.dll file on a server. The weblink receives map requests from the .dll, and then sends HTML back to the .dll, which then passes them on to the user's personal computer.



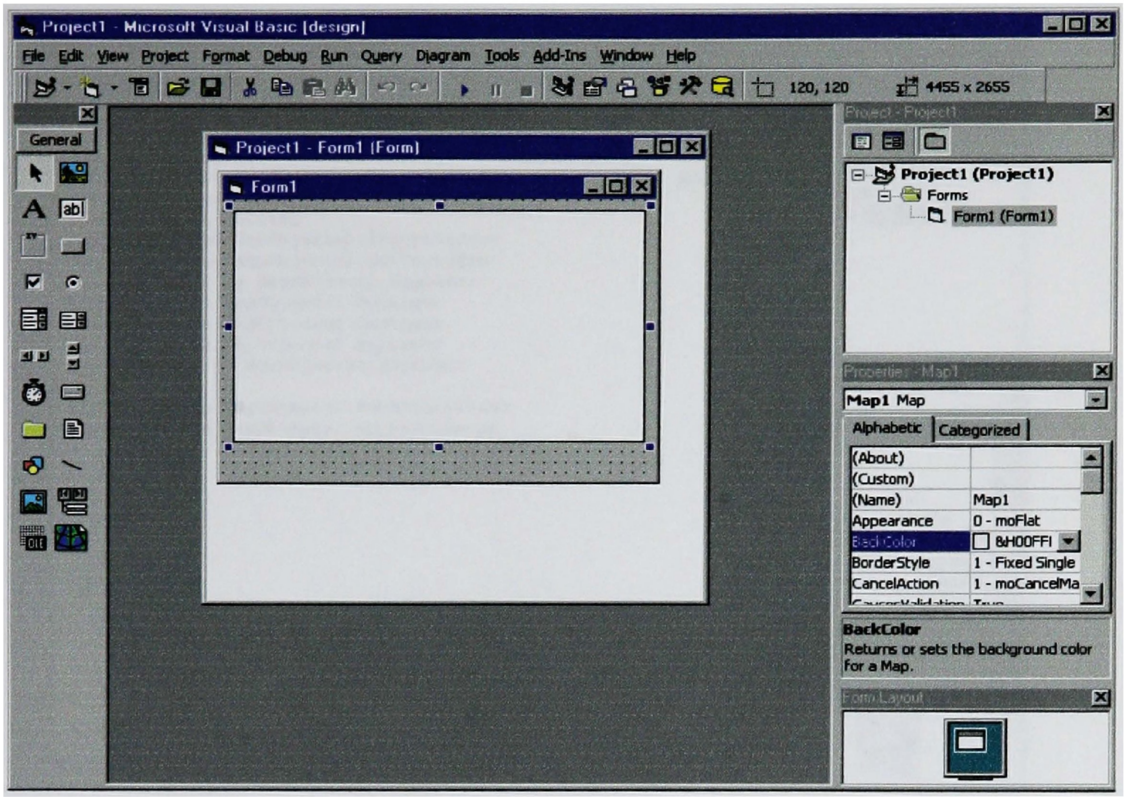
(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	□ &H8000000F
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True

Caption
Returns/sets the text displayed in an object's title bar or below an object's

Figure 8

Using the Microsoft Visual Basic 6.0, programming GUI, seen on the left is the component tool bar.

Having installed these tools in the programming environment,



Map1 Map

Alphabetic | Categorized

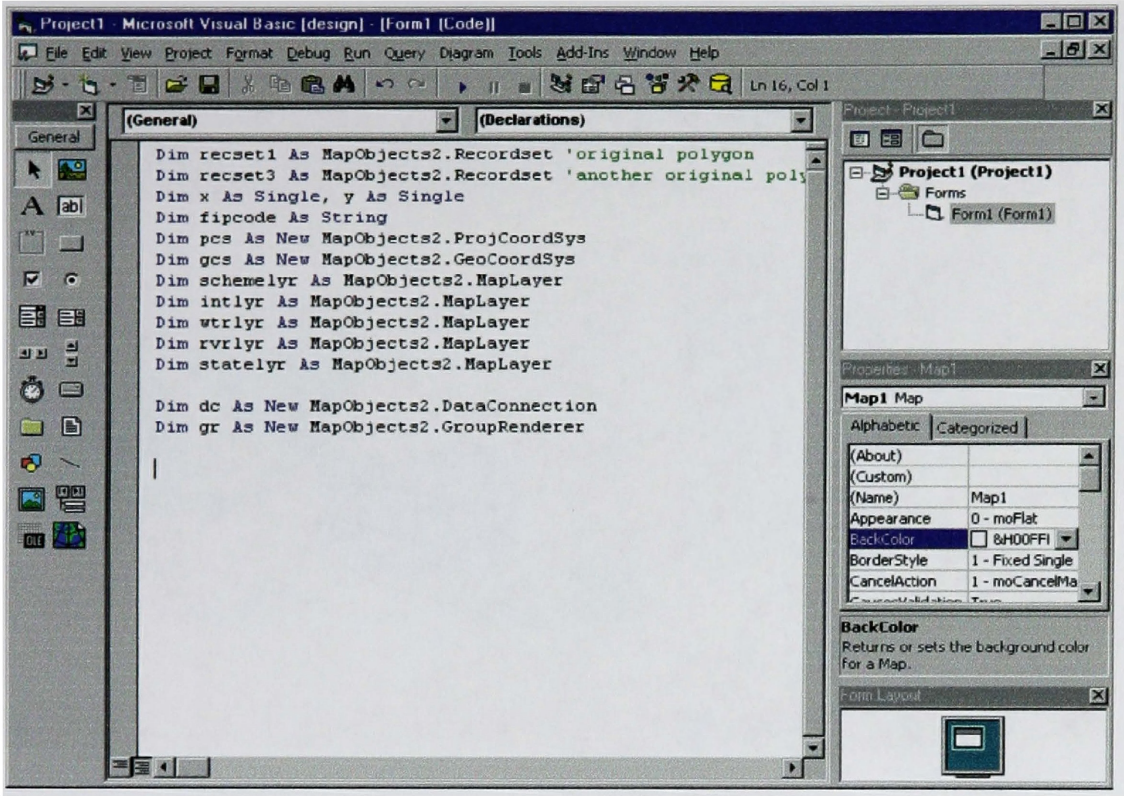
(About)	
(Custom)	
(Name)	Map1
Appearance	0 - moFlat
BackColor	□ &H00FFI
BorderStyle	1 - Fixed Single
CancelAction	1 - moCancelMa
CausesValidation	True

BackColor
Returns or sets the background color for a Map.

Figure 9

A Form has been opened a map control has been selected and placed in the form. Save the form and the project at this point.

the ICSA developer next considers the graphical user interface. The MOIMS software architecture uses a three-tiered approach. On the first tier, the user chooses a county from the menu, as shown in Figure 4. The second tier consists of the MOIMS catalog, launcher, and administrator functions. The third tier is comprised of ICSA functions that reside on the server. A user's first-tier choice of County, as shown in Figure 11, is passed to the MOIMS middle-ware, on the second tier, which receives requests from the user, parses them, and then passes them to the ICSA. The ICSA is registered and always running on



```
Dim recset1 As MapObjects2.Recordset 'original polygon
Dim recset3 As MapObjects2.Recordset 'another original poly
Dim x As Single, y As Single
Dim fipcode As String
Dim pcs As New MapObjects2.ProjCoordSys
Dim gcs As New MapObjects2.GeoCoordSys
Dim schemelyr As MapObjects2.MapLayer
Dim intlyr As MapObjects2.MapLayer
Dim wtrlyr As MapObjects2.MapLayer
Dim rvrlyr As MapObjects2.MapLayer
Dim statelyr As MapObjects2.MapLayer

Dim dc As New MapObjects2.DataConnection
Dim gr As New MapObjects2.GroupRenderer
```

(About)	
(Custom)	
(Name)	Map1
Appearance	0 - moFlat
BackColor	&H00FFI
BorderStyle	1 - Fixed Single
CancelAction	1 - moCancelMa

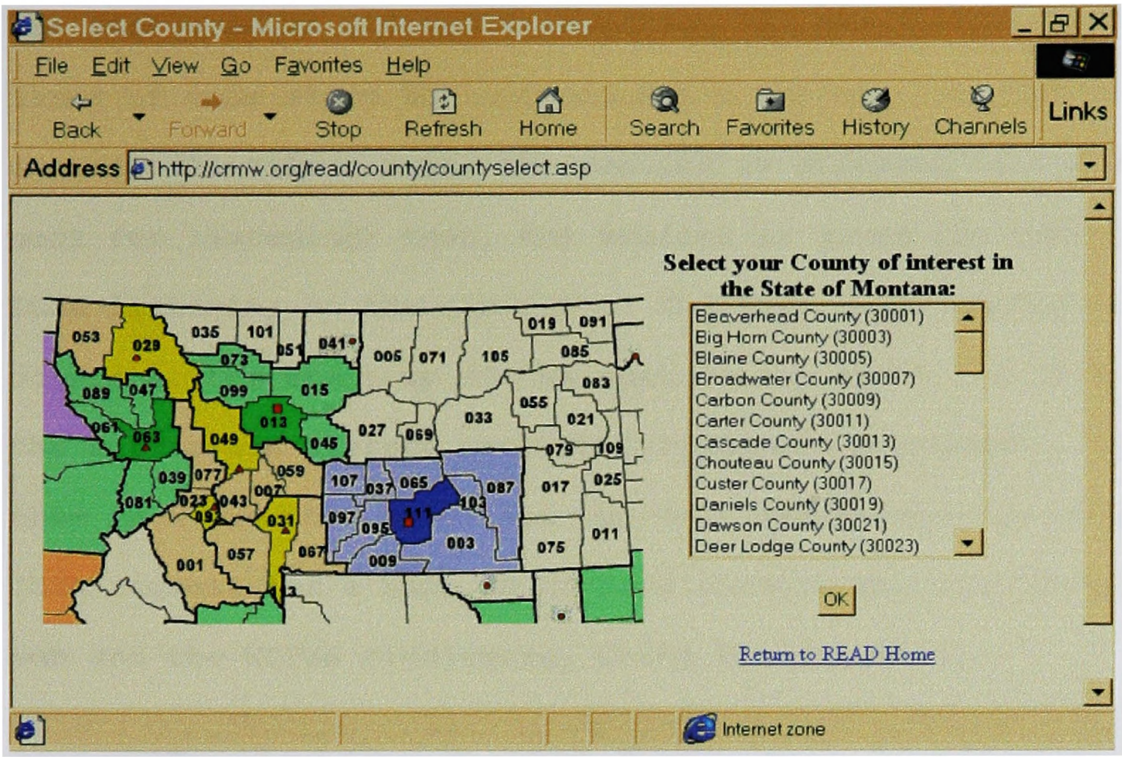
BackColor
Returns or sets the background color for a Map.

Figure 10

The Visual Basic code editor. Specify the "option explicit", then declare relevant variables using the "DIM" character string.

the server waiting for requests. The program generates the map as a server side GIS element. The application creates an image file, saves that file to the server hard drive, and almost simultaneously writes that image file to an image tag on the computer of the user that requested the image, inserting the file into an HTML page returned to the user through the esrimap.dll. This entire process is managed by the MOIMS middleware.

The Internet Cartographic Software Application residing on a server, comprises the third tier of MOIMS architecture. As mentioned, the ICSA communicates with the MOIMS through the weblink control. All middleware is installed and running



Select your County of interest in the State of Montana:

Beaverhead County (30001)
Big Horn County (30003)
Blaine County (30005)
Broadwater County (30007)
Carbon County (30009)
Carter County (30011)
Cascade County (30013)
Chouteau County (30015)
Custer County (30017)
Daniels County (30019)
Dawson County (30021)
Deer Lodge County (30023)

OK

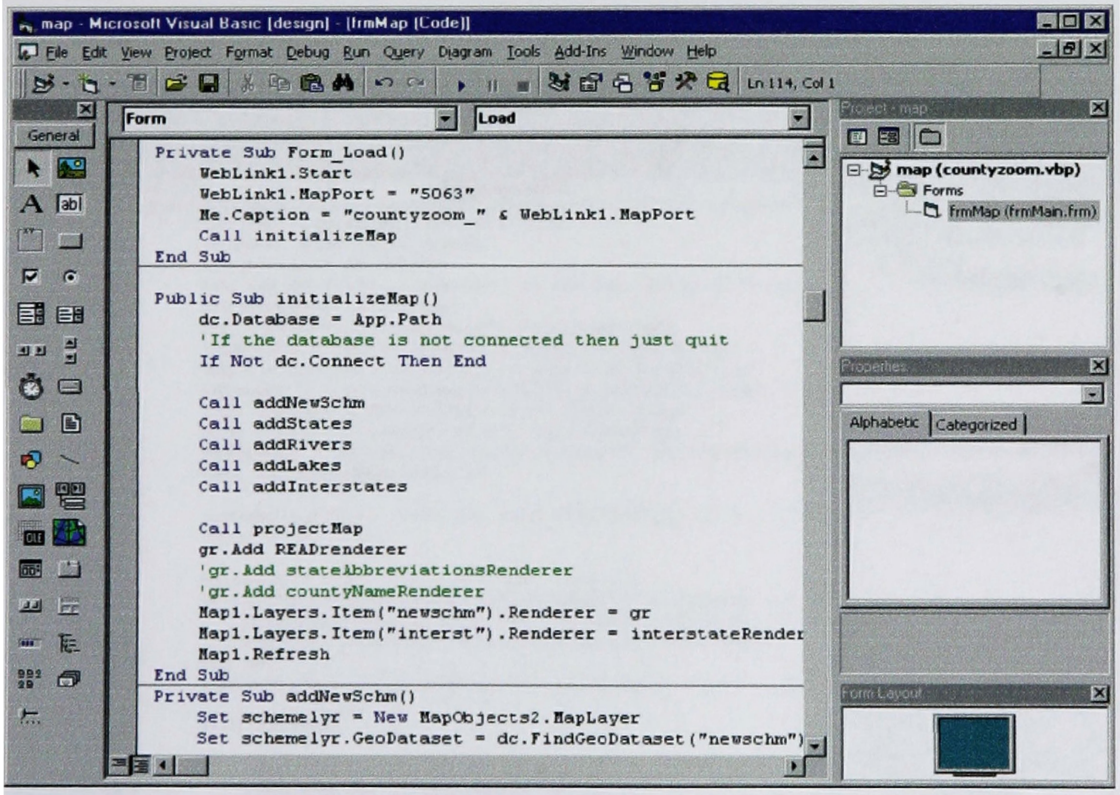
Return to READ Home

Figure 11

The READ ICSA user chooses a county. The county containing the largest population center, is shown in the darkest shade of the color scheme.

server tier constantly, providing responses to any user requests. The weblink has several available actions including Start, Stop, WriteResponseHeader, WriteString, WriteFile, BMP2JPEG, and BMP2GIF. The event is a request, initiated by the ICSA using "Start." This event begins the communication with the esrimap.dll, then "Stop" is used to end the communication.

The Internet Cartographic Software Application incorporates three sub-functions, Form-Load, Web-Link request, and Value Map Renderer. Their operations allow flexibility to accommodate various sets of user preferences, but generally, they function as follows. The Form-Load sub-function, the code of which appears in Figure 12, is one of the ICSA's weblink methods used to define the cartographic display transmitted via the Internet. These lines of code start the communication between the ICSA and the esrimap.dll file. Next, the weblink is directed to the available port for listening; then, the Weblink is given the name of the ICSA listening on the map port. The Form Load sub-function occurs at the start up of the application since the ICSA runs on the server and listens constantly. Request management, by placing them in a cue when more than one user simultaneously makes an ICSA request for a map, is handled through the weblink request sub and the MOIMS middleware, using the erimap.dll.


map - Microsoft Visual Basic [design] - [frmMap (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 114, Col 1
General
Form
Load
Project - map
map (countyzoom.vbp)
Forms
frmMap (frmMain.frm)
Properties
Alphabetic
Categorized
Form Layout

```
Private Sub Form_Load()
    WebLink1.Start
    WebLink1.MapPort = "5063"
    Me.Caption = "countyzoom_" & WebLink1.MapPort
    Call initializeMap
End Sub

Public Sub initializeMap()
    dc.Database = App.Path
    'If the database is not connected then just quit
    If Not dc.Connect Then End

    Call addNewSchm
    Call addStates
    Call addRivers
    Call addLakes
    Call addInterstates

    Call projectMap
    gr.Add READrenderer
    'gr.Add stateAbbreviationsRenderer
    'gr.Add countyNameRenderer
    Map1.Layers.Item("newschm").Renderer = gr
    Map1.Layers.Item("interst").Renderer = interstateRender
    Map1.Refresh
End Sub
Private Sub addNewSchm()
    Set schemelyr = New MapObjects2.MapLayer
    Set schemelyr.GeoDataset = dc.FindGeoDataset("newschm")
```

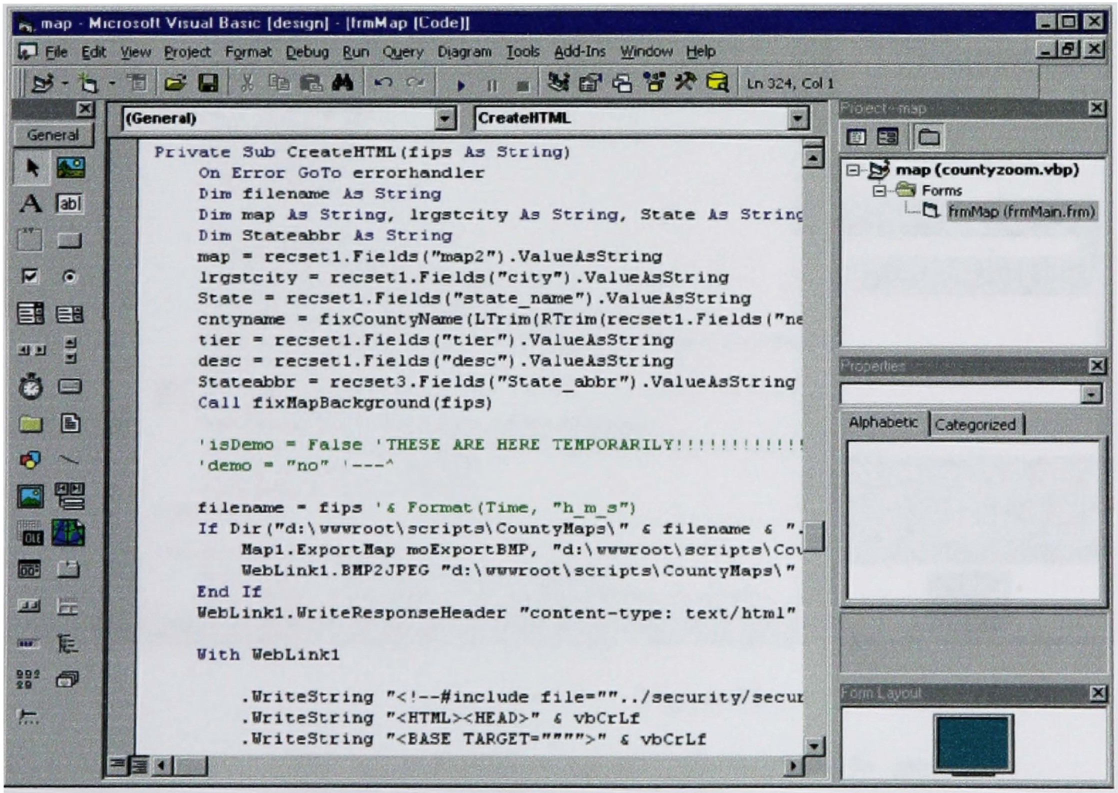
Figure 12

The Form-Load sub routine. The first sub routine to execute is the Form Load.

The weblink has several methods and properties important to the

ICSA. In addition to providing the link to the web application created for delivering maps requested, it also provides the actual graphics interchange file (GIF) generation

The weblink is also used to create the image of the requested map. The ICSA running on the server has a full complement of GIS functionality, allowing dynamic selecting, zooming, labeling, and categorizing. Once the server-side GIS has been completed, the ICSA uses the weblink to capture the image as a bitmap (BMP), and then convert it to a standard web image format such as (JPEG) or (GIF) file. Finally, the BMP image is saved to the server's hard drive briefly



```
Private Sub CreateHTML(fips As String)
    On Error GoTo errorhandler
    Dim filename As String
    Dim map As String, lrgstcity As String, State As Strinc
    Dim Stateabbr As String
    map = recset1.Fields("map2").ValueAsString
    lrgstcity = recset1.Fields("city").ValueAsString
    State = recset1.Fields("state_name").ValueAsString
    cntyname = fixCountyName(LTrim(RTrim(recset1.Fields("ne
    tier = recset1.Fields("tier").ValueAsString
    desc = recset1.Fields("desc").ValueAsString
    Stateabbr = recset3.Fields("State_abbr").ValueAsString
    Call fixMapBackground(fips)

    'isDemo = False 'THESE ARE HERE TEMPORARILY!!!!!!!!!!!!
    'demo = "no" '---^

    filename = fips '& Format(Time, "h_m_s")
    If Dir("d:\wwwroot\scripts\CountyMaps\" & filename & ".
        Map1.ExportMap moExportBMP, "d:\wwwroot\scripts\Cou
        WebLink1.BMP2JPEG "d:\wwwroot\scripts\CountyMaps\"
    End If
    WebLink1.WriteResponseHeader "content-type: text/html"

    With WebLink1

        .WriteString "<!--#include file=""../security/secur
        .WriteString "<HTML><HEAD>" & vbCrLf
        .WriteString "<BASE TARGET="""">" & vbCrLf
```

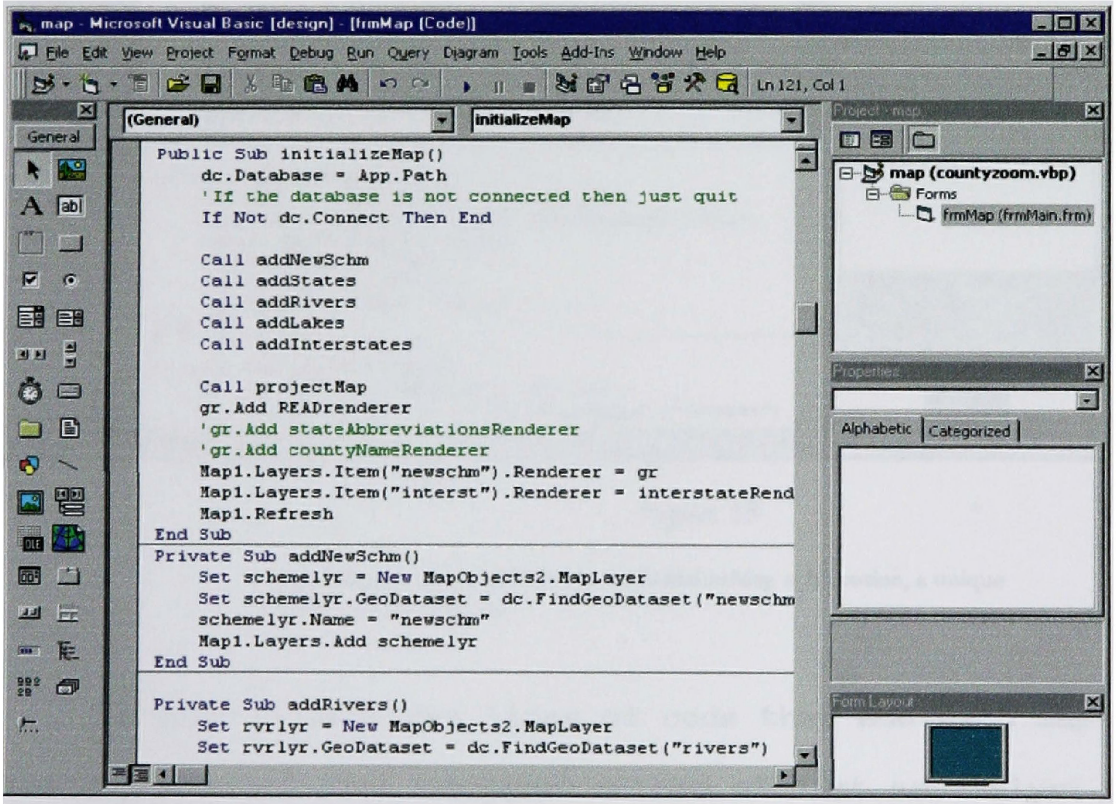
Figure 13

Image files can be dynamically created and inserted into HTML output.

at the same time as it is written to the image tag used for the requesting arguments. The weblink also provides the methods for writing standard HTML within the ICSA, allowing image files to be created, and dynamically inserted into the HTML (Figure 13).

Cartographic Sub-Functions

The web-link sub-function is used by ICSA to assemble images depicting the products of multiple cartographic sub-functions. The creation of cartographic sub-functions provides the ICSA user with options typically only available when using GIS in a programming environment.



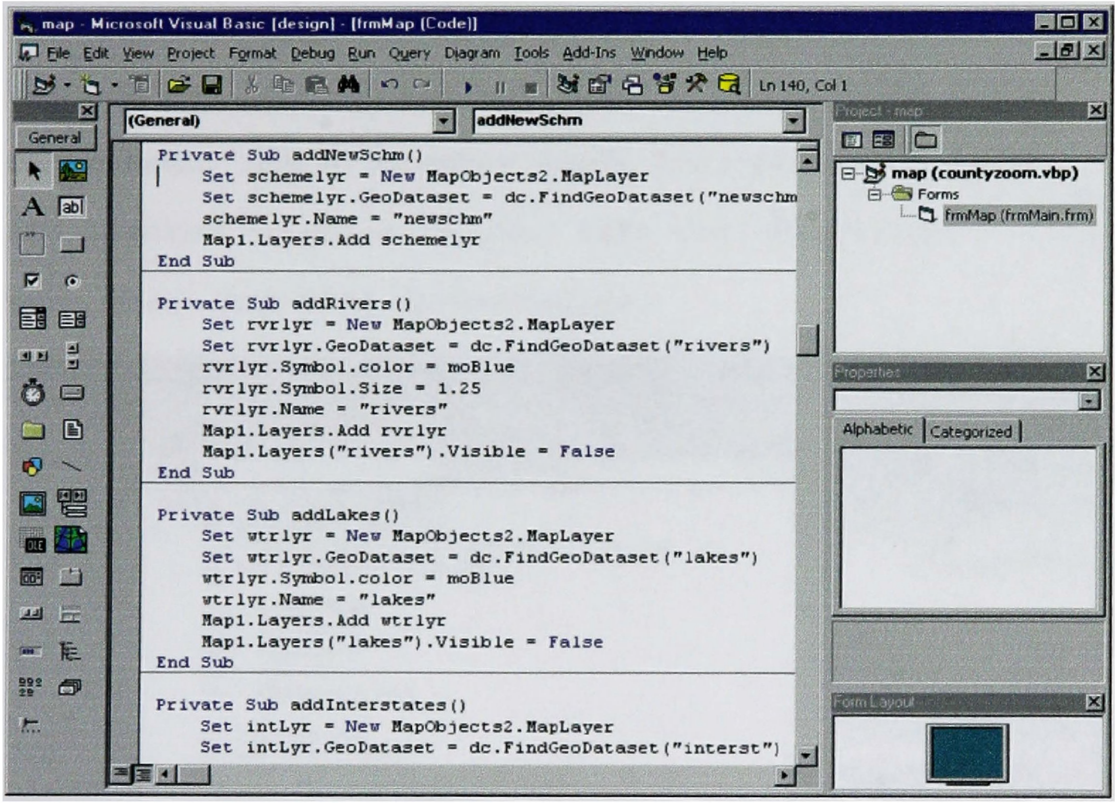
```
Public Sub initializeMap()
    dc.Database = App.Path
    'If the database is not connected then just quit
    If Not dc.Connect Then End

    Call addNewSchm
    Call addStates
    Call addRivers
    Call addLakes
    Call addInterstates

    Call projectMap
    gr.Add READrenderer
    'gr.Add stateAbbreviationsRenderer
    'gr.Add countyNameRenderer
    Map1.Layers.Item("newschm").Renderer = gr
    Map1.Layers.Item("interst").Renderer = interstateRend
    Map1.Refresh
End Sub
Private Sub addNewSchm()
    Set schemelyr = New MapObjects2.MapLayer
    Set schemelyr.GeoDataset = dc.FindGeoDataset("newschm
    schemelyr.Name = "newschm"
    Map1.Layers.Add schemelyr
End Sub

Private Sub addRivers()
    Set rvrlyr = New MapObjects2.MapLayer
    Set rvrlyr.GeoDataset = dc.FindGeoDataset("rivers")
```

Figure 14

The first instruction in initialize Map is where the geographic shapefile data is located on the hard drive.

By using the Initialize Map sub-function, data layers are specified and added to the Map Control (Figure 14).

Data layers are specified next to the Call command. Although the called layers are added to the map control only when ICSA is restarted, they are added to the map control from the initial map sub-function when it is called (Figure 15).



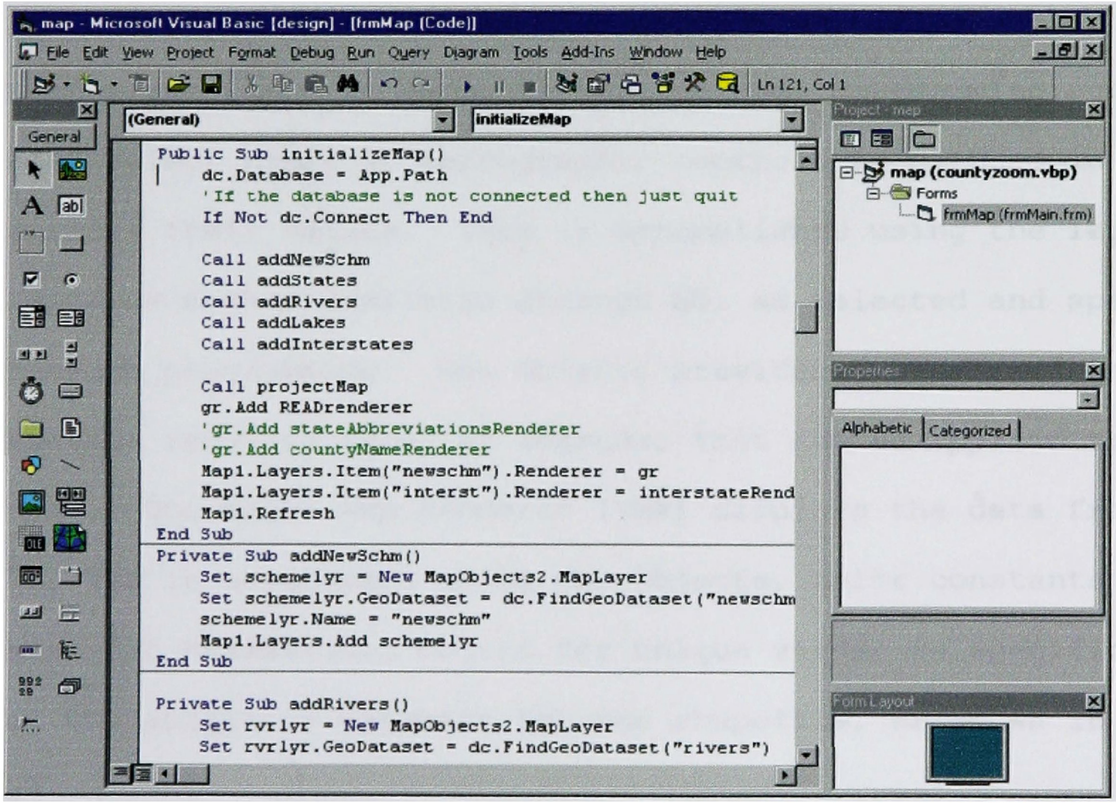
```
Private Sub addNewSchm()
    Set schemelyr = New MapObjects2.MapLayer
    Set schemelyr.GeoDataset = dc.FindGeoDataset("newschm
    schemelyr.Name = "newschm"
    Map1.Layers.Add schemelyr
End Sub

Private Sub addRivers()
    Set rvrlyr = New MapObjects2.MapLayer
    Set rvrlyr.GeoDataset = dc.FindGeoDataset("rivers")
    rvrlyr.Symbol.color = moBlue
    rvrlyr.Symbol.Size = 1.25
    rvrlyr.Name = "rivers"
    Map1.Layers.Add rvrlyr
    Map1.Layers("rivers").Visible = False
End Sub

Private Sub addLakes()
    Set wtrlyr = New MapObjects2.MapLayer
    Set wtrlyr.GeoDataset = dc.FindGeoDataset("lakes")
    wtrlyr.Symbol.color = moBlue
    wtrlyr.Name = "lakes"
    Map1.Layers.Add wtrlyr
    Map1.Layers("lakes").Visible = False
End Sub

Private Sub addInterstates()
    Set intLyr = New MapObjects2.MapLayer
    Set intLyr.GeoDataset = dc.FindGeoDataset("interst")
```

Figure 15

As each data layer is added in the initializeMap sub routine, a unique property must be set.

Figure 15 includes lines of code that add each map layer to the map window. The declared method of "Set schemelyr" within the sub-function determines what the name of the map layer will be in ICSA. The next line of code determines the type of layer defined as a geographic data set using the geodataset method. Within that same line of code, the sub-function is instructed to find a specific geographic data set.

A command was issued during the initialize map sub-function to establish the drive and directory that the geographic data is saved in. This command indicates the data is stored in the same drive and directory where the ICSA is saved in (Figure 16). The command is App.Path, which stands for application path. This command can prevent the application from erring during deployment, the programmer needs only place the files the application is going to use, like the shapefiles, in the same path where the ICSA is installed.



```
Public Sub initializeMap()
    dc.Database = App.Path
    'If the database is not connected then just quit
    If Not dc.Connect Then End

    Call addNewSchm
    Call addStates
    Call addRivers
    Call addLakes
    Call addInterstates

    Call projectMap
    gr.Add READrenderer
    'gr.Add stateAbbreviationsRenderer
    'gr.Add countyNameRenderer
    Map1.Layers.Item("newschm").Renderer = gr
    Map1.Layers.Item("interst").Renderer = interstateRend
    Map1.Refresh
End Sub
Private Sub addNewSchm()
    Set schemelyr = New MapObjects2.MapLayer
    Set schemelyr.GeoDataset = dc.FindGeoDataset("newschm
    schemelyr.Name = "newschm"
    Map1.Layers.Add schemelyr
End Sub

Private Sub addRivers()
    Set rvrlyr = New MapObjects2.MapLayer
    Set rvrlyr.GeoDataset = dc.FindGeoDataset("rivers")
```

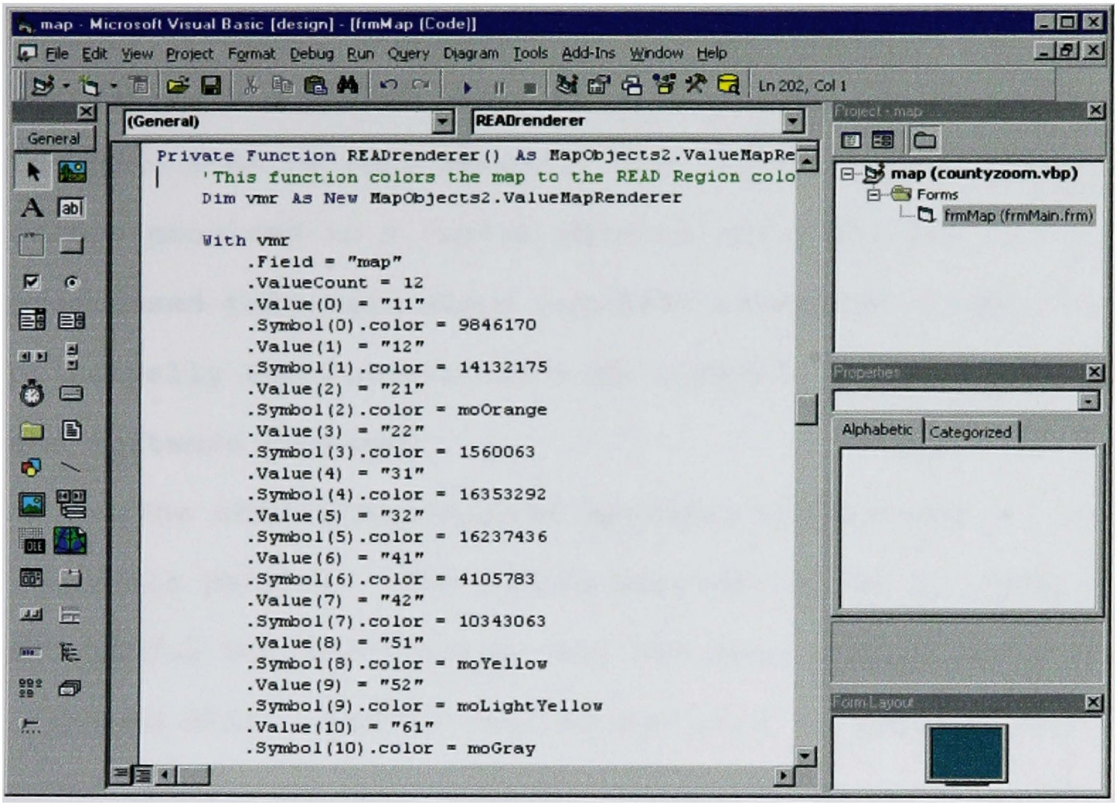
Figure 16

Using the App.Path command to indicate the drive and directory where the program is saved along with other files the program will need.

As shown in Figure 16, the command "dc.Database = App.Path" directs the application to find the data shapefile(s) in the same drive and directory where ICSA resides. The names of all the sub-functions are the same as the map-layers that will be used in the ICSA.

To make desired geographic data available to the ICSA, each map layer is added as an individual sub-function. This compartmentalizing of the map layers benefits the ICSA in two ways. First, the map layers are loaded when the program is started, always ready to be called. Second, the map layers are kept separate, which expedites zooming and labeling, accelaerating the application's ability to return map output. When a map layer is added in this method, programming actions must select types of cartographic controls to be used in order to specify their values. This is accomplished using the legend renderer method available through MO, as selected and specified through programming. Map Objects provides legends, or one of various rendered types of legends, that can be applied to the map data. The Value Map Renderer (VMR) displays the data from READ. The VMR is programmed using Map Objects, color constants and specific hexadecimal colors for unique values in specific fields of the attribute database for the shapefile, as shown in Figure 17.

The sub-function "READrenderer", a method for using the (VMR), is used to set the color scheme for READ data display. It is the legend renderer devised to allow linking particular colors to unique values in an array specified from the database files associated with the shapefiles. In Figure 17, the READ renderer declares a variable VMR as new "MapObjects.ValueMapRenderer".



```
Private Function READrenderer() As MapObjects2.ValueMapRe
    'This function colors the map to the READ Region colo
    Dim vmr As New MapObjects2.ValueMapRenderer

    With vmr
        .Field = "map"
        .ValueCount = 12
        .Value(0) = "11"
        .Symbol(0).color = 9846170
        .Value(1) = "12"
        .Symbol(1).color = 14132175
        .Value(2) = "21"
        .Symbol(2).color = moOrange
        .Value(3) = "22"
        .Symbol(3).color = 1560063
        .Value(4) = "31"
        .Symbol(4).color = 16353292
        .Value(5) = "32"
        .Symbol(5).color = 16237436
        .Value(6) = "41"
        .Symbol(6).color = 4105783
        .Value(7) = "42"
        .Symbol(7).color = 10343063
        .Value(8) = "51"
        .Symbol(8).color = moYellow
        .Value(9) = "52"
        .Symbol(9).color = moLightYellow
        .Value(10) = "61"
        .Symbol(10).color = moGray
```

Figure 17

Programming the Value Map Renderer to associeate unique color values with map themes.

The program then specifies, in the next line of code, the field from the shapefile that the renderer will use, in the character string ".ValueCount = 12" that specifies twelve unique values in the array or field called "map". The renderer has now been told in what field it will be mapping, and how many unique values it will use.

The next task, setting the color values for each data layer, has considerable influence over the final product and is particularly interesting. The READ ICSA is programmed to match the associations of colors with data layers used in the first generation of READ maps. Each unique value in the 12-element, zero-based array is used for one color category. The Map Objects program is pre-loaded with color constants for mapping, however, the READ ICSA is very specific and requires an improved set of colors provided in a custom palette built for MO. As programming progressed this task alone involved extensive research, principally experimentation with numerous programming languages and software packages.

The READ ICSA required specific experiments to reproduce the color palette. The colors provided by MO as color constants are useful but insufficient for the many gradations of color found in READ. Hexadecimal values used in ESRI software to designate colors were used to expand the MO color palette. Using hexadecimal numbers, enabled displaying the same colors via the ICSA that would be found on printed maps. The specific steps for establishing these colors was learned through trial and error. The trial and error method is compatible with GUI type object oriented programming because programs do not need to be compiled before running.

The method used for this study began with specifically designing the colors for READ in ArcView 3.2. Once the colors were defined in ArcView 3.2 the challenge was to find a method for converting the colors from the ArcView 3.2 format, which is in red green blue (RGB) to hexadecimal values. The Internet was searched for a program that would convert RGB numbers to hexadecimal numbers. The code for a small application was found, downloaded and compiled, and then used to change the numbers. Another issue confronted was that, although ArcView 3.2 legend label was that its colors RGB, they were actually in hue saturation brightness (HSB). To overcome this, values of HSB were entered into Adobe Illustrator and converted to RGB values then finally translated using an algorithm once again downloaded from the Internet. After trial and error, a custom palette for the READ scheme was effectively matched and programmed.

Programming and performing the cartographic enhancements to the basic applications provided by Visual Basic and Map Objects showed some challenges of electronic cartography that were overcome in createing the READ ICSA, the colors given are merely good starting points for the cartographer. This is an example of how online GIS software, manufactured for wide markets, generally requires users who have GIS and cartographic skills. The READ ICSA represents an improvement in providing maps via the Internet while taking advantage of cartographers' training and experience in making meaningful decisions to rapidly provide accurate useful maps to citizens.

Early in-house testing by the developer was, of course, crucial to providing a functional system. However, after confirming the basic operation, another approach to testing the ICSA was used. This real-time approach assesses software by subjecting it to use by a wide audience of personal computer operators. Any computer with Internet access is a testing facility. In the case of the READ ICSA, numerous individuals test this application every day. In documentation delivered to READ subscribers, Center for the Rocky Mountain West Director, economist Larry Swanson, states that this ICSA is used by economic development specialists, business analysts, researchers, journalists, university faculty, other educators, and other citizens needing demographic and economic information[11].

READ ICSA display standards and functionality were also tested by its users. These tests included extensive use of the two most widely available Internet browsers, Microsoft Internet Explorer and Netscape Navigator. These tests have demonstrated that this ICSA works with both Internet browsers to rapidly provide and display high quality up-to-date maps.

[11] Swanson, L. 2002. Documentation provided to READ subscribers.

Chapter 5

DISCUSSION

A limitation of the READ ICSA as a design tool is its dependence on multiple programming languages. It relies heavily on the middleware application known as Map Objects Internet Map Server (MOIMS). This is the weakest portion of the system's architecture because the cartographic application depends on three programs used with the Internet server, which are unpredictable and prone to failure. As of this writing, Internet programming and development tools and methods have become available that work more reliably and rapidly. When this project began, the World Wide Web Consortium (W3C) had not adopted modern web protocols such as Extensible Markup Language (XML). These technological advancements provide faster data browsing and more stable map applications.

Since the READ ICSA became available through the Center for the Rocky Mountain West, the approach developed for this project has been used to create other cartographic applications. For example, this researcher has just completed an ICSA that integrates more than ten years of surface water supply index (SWSI) data, and displays this information in maps available on the Natural Resources Information System web site.

The advancements learned from this project have lead to the creation of a fully functional COM component (DLL) installed on the server using Component Services or COM+. The mapping technology and objects are used in ASP allowing the cartographer much more rapid development of Internet images and web sites with geographic data display demands.

This study used object oriented programming approaches to maximize accessibility of information in GIS databases, enabling Internet users without programming experience to efficiently access cartographic data. The ICSA approach may be further refined as more powerful objects become available. Exciting advances may be realized with the creation of .dll objects that can be used within VB .net, and objects that can be used on open source platforms.

The Center for the Rocky Mountain West currently uses two other ICSA products, for selecting different regional areas. In development is an online map product that will provide clients with the ability to compare any years of data at the county level in two comparable time periods. The object oriented programming approach has exposed many opportunities to view READ as dynamic map products. The potential for web-services to deliver dynamic economic content in map form has been merely begun with the ICSA for READ. The future through XML promises faster data access thus allowing less processing time to be incorporated into READ.

GLOSSARY

Accessibility: An aggregate measure of how reachable locations are from a given location. The command computes values for accessibility as a function of the distance between locations and an empirically derived distance decay parameter.

Active Data Object (ADO): A set of object-based data access interfaces optimized for Internet-based, data centric applications.

Active Server Pages ASP: A specification for a dynamically created Web page with an *.ASP* extension that utilizes ActiveX scripting usually VB Script or JavaScript code. When a browser requests an ASP page, the Web server generates a page with HTML code and sends it back to the browser. So ASP pages are similar to CGI scripts, but they enable Visual Basic programmers to work with familiar tools.

Active X: A technology developed by Microsoft for sharing information among different applications, ActiveX supports new features that enable it to take advantage of the Internet. For example, an ActiveX control can be automatically downloaded and executed by a Web browser. ActiveX is not a programming language, but rather a set of rules for how applications should share information. It allows you to create cross-platform, language-independent controls.

Attribute: 1) A characteristic of a geographic feature described by numbers, characters, images and CAD drawings, typically stored in tabular format and linked to the feature by a user-assigned identifier (e.g., the attributes of a well might include depth and gallons per minute), 2) A column in a database table.

CERN: The European Organization for Nuclear Research, the world's largest particle physics center. Here physicists come to explore what matter is made of and what forces hold it together.

Class: All objects of the same type are said to be members of the same class. A generic description of a group of objects that will have a state and behavior.

Client/Server: A software system is said to have a client/server architecture when there is a central process (server) which accepts requests from multiple user processes (clients).

Component Object Model (COM): The object-oriented programming model that defines how objects interact within a single application or between applications. In COM, client software accesses an object through a pointer to an interface - a related set of functions called methods - on objects.

Command Line interface:A software product that allows the user to type in commands at a prompt. Contrast to forms interface.

Custom Map Set: Thematic maps providing information based on the map users needs and requests.

Data Access Security: Measures taken to control system users' ability to view or modify data. These measures can include logical views of data and explicit access rights by group or individual users.

Database: A logical collection of interrelated information, managed and stored as a unit, usually on some form of mass-storage system such as magnetic tape or disk. A GIS database includes data about the spatial location and shape of geographic features recorded as points, lines, areas, pixels, grid cells, or tins, as well as their attributes.

Database Design: The formal process of analyzing facts about the real world into a structured database model. Database design is characterized by the following phases: requirement, analysis, logical design and physical design.

DBF Format: The data file format originally used by the dBase product and now the most common PC DBMS format.

Data Portability: An interoperability objective in which the data exported by one storage system can be imported meaningfully by others.

Data Schema: Formal description of a data model.

Data Set: A named collection of logically related data items arranged in a prescribed manner.

Dynamic Shared Library Link: An image that exports functions and global variables to other images. A shared library is not included with the application code at link time but is linked in dynamically at runtime. Also known as a *shared library, dynamic library, dylib, dynamic link library,* and *DLL.*

Dynamic Typing and Binding: Binding a message to a particular method at run time is known as dynamic binding.

Embedded SQL: SQL statements that are embedded in a host language program.

Encapsulation: An object hides its instance variables and method implementations from other parts of the program. This encapsulation allows the programmer that uses an object to concentrate on what the object does rather than how it is implemented.

Event: A geographic feature occurring on or along a linear feature. There are three event types: linear, continuous, and point.

Event Source: This is a name assigned by the user to reference a DBMS table containing event data for use with the dynamic segmentation commands. This is similar to the relate name.

Feature Attribute Table: A table used to store attribute information for a specific coverage feature class. Feature attribute tables supported for coverages include:
<COVER>.PAT for polygons or points
<COVER>.AAT for arcs
<COVER>.NAT for nodes
<COVER>.RAT for routes
<COVER>.SEC for sections
<COVER>.PAT for regions
<COVER>.TAT for annotation (text)
where <COVER> is the coverage name

Feature Class: The type of geographic feature(s) that is/are being represented in a coverage. Feature classes include LINE, POINT, POLYGON, TIC or ANNOTATION. A single coverage can contain one or more of these feature classes.

Field: In a database, another term for column.

File: A set of related information that a computer can access by a unique name (e.g., a text file, a data file, a DLG file). Files are the logical units managed on disk by the computer's operating system. Files may be stored on tapes or disks.

Forms Interface: A graphic user interface characterized by user-controlled movement of a cursor from one data field to another. Contrast to command line interface.

GBF/DIME: For the 1980 census, the U.S. Census Bureau produced Geographic Base Files (GBF) and Dual Independent Map Encoding (DIME) files, containing census geographic statistical codes and coordinates of line segments for most metropolitan areas.

Geographic Data: The locations and descriptions of geographic features. The composite of spatial data and descriptive data.

Geographic Database: A collection of spatial data and related descriptive data organized for efficient storage and retrieval by many users.

Geographic Information System (GIS): An organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information.

Graphical User Interface (GUI): A graphical method of controlling how a user interacts with a computer to perform various tasks. Instead of issuing commands at a prompt, the user performs desired tasks by using a mouse to choose from 'a dashboard' of options presented on the display screen. These are in the form of pictorial buttons (icons) and lists. Some GUI tools are dynamic and the user must manipulate a graphical object on the screen to invoke a function; for example, moving a slider bar to set a parameter value (e.g., setting the scale of a map).

Hardware: The physical components of a computer system-the computer, plotters, printers, terminals, digitizers, and so on.

Hypertext Markup Language (HTML): Most common protocol for displaying text and graphics on Internet web pages.

Hypertext Transfer Protocol (HTTP): The set of rules for exchanging files (text, graphic images, sound, video and other multimedia files) on the World Wide Web. Relative to the TCP/IP suite of protocols (which are the basis for information exchange on the Internet), HTTP is an application protocol.

Inheritance: The process by which one class borrows the state and behavior descriptions of another class and adds them to its own state and behavior description. Inheritance is one of three ways in which classes may be linked, and is sometimes described as an "is a" relationship.

Instantiate: To create an object based on an existing class definition.

Interface: For data communication, a hardware and software link that connects two computer systems, or a computer and its peripherals.

Internet: An international consortium of wide area networks that operate using a standard set of addresses allowing machine-to-machine connectivity on a global scale.

Java: A programming language expressly designed for use in the distributed environment of the Internet. It was designed to have the "look and feel" of the C++ language, but it is simpler to use than C++ and enforces an object-oriented programming model.

Layer: A thematic set of spatial data described and stored in a database or a map library. Layers organize a database or map library by subject matter (e.g., soils, roads and wells).

Legend: The reference area on a map that lists and explains the colors, symbols, line patterns, shadings, and annotation used on the map. The legend often includes the scale, origin, orientation and other map information.

Map Control: The Active X component central to the Map Objects programming. Property of ESRI incorporated, Redlands CA.

Map Objects 2.1 (MO): Environmental Systems Research Institute Map Objects is an Active X component used in various programming languages to perform geographic queries and build maps displayed as images.

Map Objects Internet Map Server (MOIMS): Middleware software providing Internet functionality in conjunction with Map Objects enabled programs.

Map Projection: A mathematical model that transforms the locations of features on the Earth's surface to locations on a two-dimensional surface. Because the Earth is three-dimensional, some method must be used to depict a map in two dimensions. Some projections preserve shape; others preserve accuracy of area, distance or direction.

Map Query: The process of selecting information from a GIS by asking spatial or logical questions of the geographic data. Spatial query is the process of selecting features based on location or spatial relationship (e.g., select all features within 300 feet of another; point at a set of features to select them). Logical query is the process of selecting features whose attributes meet specific logical criteria (e.g., select all polygons whose value for AREA is greater than 10,000 or select all streets whose name is Main St.). Once selected, additional operations can be performed, such as drawing them, listing their attributes or summarizing attribute values.

Multiple Document Interface (MDI): Developed by Microsoft, with menus, buttons, tools, and windows called documents. ArcView is based on the MDI standard, including multiple document types: Project View, Table, Layout, Chart, and Scripts. Documents can be organized and manipulated in a variety of standard ways: tiled, cascaded, iconified, resized or closed.

Middleware: This software manages the communication between a client program and a database. For example, a Web server connected to a database can be considered middleware and the Web server sits between the client program (a Web browser) and a database. The middleware allows the database to be changed without necessarily affecting the client, and visa versa.

Navigation Handles: The data browsing paradigm established such basic Internet browsing tools as the back button, home, and the uniform resource locator address bar.

Object: As stated previously, an object is one of the basic building blocks in OO programming. An object can receive messages and then act on these messages to alter the state of itself (the size and position of a rectangle object for example). In software an object consists of instance variables (data) that represent the state of the object, and methods (like C functions) that act on these variables in response to messages. Rather than 'calling' one of its methods, an object is said to 'perform' one of its methods in response to a message. (A method is known as a 'member function' in C++.)

Open Database Communication (ODBC): A standard API (application program interface) used to communicate with database management systems, developed by Microsoft.

Object Linking and Embedding (OLE): Allows objects from one application to be embedded within another (e.g., taking an Excel spreadsheet and putting it into a Word document).

Object Oriented Programming (OOP): A modern compartmentalized programming method widely adopted throughout all programming languages. Programs produced as object oriented share three priciple tenets polymorphism, encapsulation, and inheritance.

Online Access: Direct access to data that does not involve file transfer.

Packet/Packet Switching: A packet is a chunk of information sent over a network. Packet-switching is the process by which a carrier breaks up data into these chunks or packets.

Polymorphism: A concept in type theory, according to which a name (such as a variable declaration) may denote objects of many different classes that are related by some common superclass; thus, any object denoted by this name is able to respond to some common set of operations in different ways.

Regional Economies Assessment Database (READ): The regional economies assessment database provides comprehensive economic trends and conditions for the West compiled and normalized by Larry Swanson covering the past 25 years updated periodically.

Relational database management system (RDBMS): A database management system with the ability to access data organized in tabular files that can be related to each other by a common field (item). An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage.

Relate: An operation that establishes a temporary connection between corresponding records in two tables using an item common to both (i.e., relate key). Each record in one table is connected to those records in the other table that share the same value for the common item.

Relational Database: A method of structuring data as collections of tables that are logically associated to each other by shared attributes. Any data element can be found in a relation by knowing the name of the table, the attribute (column) name, and the value of the primary key.

Soft Copy: Computer programming instructions for computer hardware referred to as soft copy software.

Structured Query Language (SQL): A syntax for defining and manipulating data from a relational database. Developed by IBM in the 1970s, it has become an industry standard for query languages in most relational database management systems.

Theme: A user-defined perspective on a coverage, grid, tin or image geographic data set specified, if applicable, by a coverage name and feature class or data set name, attributes of interest, a data classification scheme, and theme-specific symbology for drawing.

TIC File: The coverage file used to store TIC coordinates and TIC IDs for a coverage.

Value Map Renderer: A programming method of using the geographic attribute table and field to thematically color code and symbolize the geographic features providing color class breaks based on unique values.

View: A logical table whose data are not physically stored. You define a view to access a subset of the columns stored in a row, access a set of columns stored in different rows, or avoid creating a redundant copy of data that is already stored.

Visual Basic 6.0: Microsoft Visual Basic is a graphical "visual" programming language, based on natural language for programming commands.

W3C: World Wide Web Consortium.

World Wide Web: A collection of protocols, based on IP and infrastructure forming the current Internet.

eXtensible Markup Language (XML): A W3C designed cut-down version of SGML an ISO standard.

Zoom: To enlarge and display greater detail of a portion of a geographic data set.

BIBLIOGRAPHY

Anderson,R.,etal. 2000. ASP 3.0 Programmers Reference. Wrox Press Ltd., Birmingham, UK.

Arctur, D. 2002. Active Object Techniques For the Production of Multiple Map and Geodata Products from a Spatial Database. Laser-Scan, Inc. Cambridge, UK.

Berners-Lee, T. and R.C. 1992. World-Wide Web: An Information Infrastructure for High-Energy Physics. presented at the *Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics* at La Londe-les-Maures, France.

Birbeck, M.,et al. 2001. Professional XML, Second Edition. Wrox Press Ltd., Birmingham, UK.

Christos, J.P., et al. 1999. The History of the Internet: A Chronology 1843 to the Present. The Moschovitis Group, Inc., New York, NY.

Clark, K.C. 1995. Analytical and Computer Cartography, Second Edition. Prentice Hall, Englewood Cliffs, NJ.

Coppock, J.T., et al. 1993. The History of GIS, Volume one. John Wiley & Sons, Inc., New York, NY.

Environmental System Research Institute, Inc. 1999. MapObjects GIS and Mapping Components Programmer's Reference. ESRI, Redlands CA, USA.

Flangan, D. 1996. JavaScript: The Definitive Guide, Third Edition. O'Reilly & Ass., Inc. Sebastopol, CA USA.

Goodman, D. 1998. Dynamic HTML: The Definitive Reference. O'Reilly & Ass., Inc. Sebastopol, CA USA.

Harder, C. 1998. Serving Maps on the Internet: geographic information on the World Wide Web. ESRI, Redlands CA, USA.

Kleinrock, L. 1961. Information Flow in Large Communication Nets. RLE Quarterly Progress Report, Cambridge MA.

Landow, G.P. 2000. The Definition of Hypertext and its History as a Concept. Johns Hopkins University Press, Providence, RI USA.

Lomax, P. 1998. VB & VBA in a Nutshell: The Language. O'Reilly & Ass., Inc. Sebastopol, CA USA.

Perry, G. 1998. Teach Yourself Visual Basic 6 in 21 Days. MacMillan Computer Publishing, Indianapolis, IN.

Swanson,L. 2002. Documentation provided to READ subscibers.

Swanson, L. 1999. Regional Economies Assessment Database. Internet Publication http://crmw.org/read/. Missoula, MT.

Tobler, W. 1959. Automation and Cartography. University of Washington Geographical Review, Seattle WA.