Distributing Stand Inventory Data and Maps Over a Wide Area Network

Thomas E. Burk

Abstract.—High-speed networks connecting multiple levels of management are becoming commonplace among forest resources organizations. Such networks can be used to deliver timely spatial and aspatial data relevant to the management of stands to field personnel. A network infrastructure allows maintenance of cost-effective, centralized databases with the potential for updating by distributed personnel. Using networks as a delivery medium also opens up the possibility for inexpensive sharing of data with relevant persons outside the organization: for example, the general public for a state agency. The software tools necessary to implement such an approach are described, as is one particular implementation as applied to geographic information systems and stand inventories.

Making good decisions requires that accurate, timely data and data analysis tools be readily available to all involved in the decisionmaking process. Today, the maps, imagery, and associated attribute data that fill this need for tactical and operational forest management planning can be effectively encapsulated in a geographic information system (GIS). A GIS allows easy integration of data sources and provides straightforward analysis capability (Prisley 1996). Two difficulties stand in the way of widespread use of GIS capability. GIS's are complex and the expertise for applying them is not typically broadly deployed throughout a forest management organization. A GIS is only as good as the data that constitute it. Because nature and resource management practices are dynamic, base GIS data must be continually updated and quality control measures applied. Both these concerns suggest a centralized staff model that is at odds with an effective management decisionmaking model.

The widespread use of intranets and the Internet to link personnel within forest management organizations (Burk 1995) provides an opportunity to address the disparity between optimal data management and decisionmaking models. Software tools were developed that allow creation of interfaces providing access to and analysis of spatial and aspatial data stored on centralized servers. The Internet World Wide Web (WWW) client/server model was used as a basis so that the resulting applications operate in the commonly understood WWW browser environment. Such a solution allows centralized control or centralized-mediated control of data quality without sacrificing effective data access.

The ForNet project, out of which software and application developments grew, is first summarized. The software tools themselves are then described, as is a prototype application for forest stand inventory. Ongoing software development directions are outlined, as is the availability of the software tools. Additional information about the project, applications, and software can be found at the ForNet WWW site (http://www.gis.umn.edu/fornet).

Professor of Forest Resources, University of Minnesota, 1530 Cleveland Avenue North, St. Paul, MN, USA.

THE ForNet PROJECT

The ForNet project was a collaborative effort between the Department of Forest Resources, College of Natural Resources, University of Minnesota, and the Minnesota Department of Natural Resources, Division of Forestry (DNR-Forestry), with funding provided by National Aeronautics and Space Administration (NASA) through the Public Access to Remote Sensing Data program. The goal of ForNet was to research use of the Internet for delivering spatial data, particularly satellite imagery and associated ancillary data, as well as analytic tools to on-the-ground forest managers. DNR-Forestry has a long history of applying the latest technological solutions to management problems on their land, including the use of satellite remote sensing data (Bauer *et al.* 1994). This tradition sped ForNet efforts and led to the operational implementation of several successful ForNet prototype applications (cf. http://www.ra.dnr.state.mn.us/).

As part of an initial needs assessment effort, ForNet personnel identified three overlapping areas for which solutions and delivery mechanisms were needed in DNR-Forestry management planning:

- Image Delivery Systems
- GIS Data Services
- Information Resources

Image Delivery Systems work focused on raster data such as satellite imagery and digitized aerial photography. Prototype solutions were developed that allow map-based retrieval and viewing of images and photographs from large databases of materials over the Internet. Landsat Thematic Mapper (TM), Advanced Very High Resolution Radiometer, and scanned color infrared aerial photography data have all been dealt with. DNR-Forestry personnel can use Image Delivery Systems solutions to retrieve image data of specific interest to them. GIS Data Services addresses both the delivery of data in a GIS-ready-to-use format and the development of customized, on-line GIS capability. The former work allows selection and sharing of data layers of broad-scale interest within DNR-Forestry. Availability of customized GIS capability over the Internet allows field offices, many without high-end technology resources, access to data sources useful in their day-to-day work. Image Delivery Systems and GIS Data Services efforts stressed development of basic tools for delivering data and analysis solutions. Early in the ForNet project we ascertained the importance of enlisting self-selected DNR-Forestry personnel in applying information technology solutions to very specific problems they were interested in. This created a pool of motivated "early adopters" that were used to evangelize to their colleagues. Such efforts fell under the Information Resources component of ForNet.

The successes of ForNet have come about both because a need existed for which new technology could provide the solution and because researchers and practitioners worked collaboratively throughout the project. Collaboration took place because researchers wanted to develop products that would be used in practice and because practitioners were open to new approaches. This seemingly straightforward circumstance seems the exception rather than the rule when considering past technology developments in forest resource management. Without this "end-to-end" collaboration, the operational implementation of research prototypes to a full statewide basis would never have occurred, and future development possibilities would have been stymied. Given the success of the project, NASA has chosen to fund additional development for an additional 5 years.

SOFTWARE DEVELOPMENTS

Currently, ForNet software tools consist of three pieces. *MapServer* provides the capability to interact with vector data. *ImageView* provides similar capability for raster data. JavaScripts and Java applications provide the glue linking *MapServer* and *ImageView* as well as additional user interface control functionality.

Several requirements were specified at the development stage of the software. The software needed to work directly with the GIS and satellite imagery data formats used for general applications. Maintaining separate data systems for localized versus distributed applications would be impractical and fraught with error. The software would need to create views "on-the-fly"; relying on a database of canned views would greatly limit the usefulness of any resulting system. Since a WWW client/server model was to be employed, the software must be capable of converting results to WWW-compatible graphic formats such as GIF's or JPEG's. Any particular application would have specific requirements in terms of display and analysis functionality. The software would need to provide a mechanism for flexible creation of interfaces allowing simple extent and layer control. Since the attribute data behind the spatial data are often of equal import, access to these data would also need to be provided.

Based on these requirements, an interesting evolution of development philosophies took place. Initially, solutions were developed that were simply Common Gateway Interface (CGI) hooks into server-residing, canned GIS, and satellite image processing software packages. Such a solution proved to be limited both in terms of response time and scalability. This initial approach evolved into stand-alone CGI programs that accessed data in native formats, completed the necessary processing and conversions, and displayed the results in a static browser window. This approach was limited in terms of the integration of vector and raster data as well as the provision of effective user control mechanisms. When Java came on the scene, a new paradigm was adopted. Image servers were written and all processing and display capability moved to the client side. While this worked reasonably well for vector data, large raster data sets, as is commonly the case with extensive satellite imagery, were problematic. Further, the necessary Java applications were quite large, requiring significant computer resources of the client. In the end, a solution combining server and client computing capability proved most effective. Such a solution allows efficient use of computing resources, integration of multiple data types, and development of intuitive user-control capability.

MapServer

MapServer is a CGI program that provides the capability to work with vector GIS data. *MapServer* runs in conjunction with a WWW server. It has been implemented on various Unix versions, including Linux, as well as Windows NT. Currently, *MapServer* works directly with the shapefile format made available by ESRI (ESRI 1995) in conjunction with their Arc and ArcView software. Template files are used to specify data layers of interest, display characteristics, and WWW interface layout for any instantiation of *MapServer*. A typical implementation of *MapServer* consists of recursive calls with results built-up sequentially. A request is initiated by users through their (*MapServer*-created) browser interface. *MapServer* retrieves and combines the necessary shapefiles and creates a GIF image. The image and associated results can be returned embedded in hypertext markup language (HTML) code or as a uniform resource locator. Shapefiles can also contain attribute data; *MapServer* can thus be used to create map-based query interfaces to such data.

MapServer is highly extensible but provides several important baseline features including:

- scale-dependent drawing and labeling
- feature-based access to attribute data
- "GET" or "POST" CGI access
- automatic legend and scalebar creation
- configurable reference map
- support for tiled data
- "on-the-fly" projection conversion
- true type font support
- shapefile indexing

These and additional features can be seen in the prototype applications at the ForNet WWW site.

The shapefile format was chosen based on the popularity of ESRI products among the target user community (resource management organizations). Other formats could be implemented as well. The shapefile specification is in the public domain, so it is easy to move data to and from the format. Each shapefile corresponds to a map layer or coverage. No topology is attached to the shapefile reducing storage requirements and drawing overhead. A shapefile actually consists of a triad of files: one containing feature vertices, one containing feature attributes, and one containing an index.

ImageView

Like *MapServer*, *ImageView* is a gateway to spatial data. With *ImageView* the spatial data are raster images stored using the ERDAS Imagine file format (ERDAS 1994). ERDAS Imagine files were chosen for several reasons: (1) libraries necessary to read and write the files are publicly available, (2) the format is designed to handle multi-spectral image data and its spatial attributes, and (3) Imagine supports the creation of "pyramid" layers, reduced resolution images that allow for rapid creation of views at reduced resolutions. The *ImageView* image server is a CGI program built on the *MapServer* framework to handle user requests; an ancillary utility program creates WWW-compatible graphics as JPEG's or progressive JPEG's from Imagine files. Parameters such as band selection, data scaling, and areal extent can be set. Additional analysis capability is accessible through the ERDAS Imagine tool kit. Support for 8-bit image data in truecolor, grayscale, or pseudocolor modes is provided. Extensions of base *ImageView* capability allow simultaneous manipulation of multiple images, data extractions for downloading, and full integration within *MapServer* frontends.

Java Enhancements

The server-residing software, *MapServer* and *ImageView*, provides the horsepower for manipulating base spatial data and creating "on-the-fly" views in WWW-compatible formats. Given the disposition of such software, its capability for producing intuitive interface control mechanisms is limited. A need also exists to combine the outputs of the vector and raster data engines so that truly flexible "executive GIS's" can be produced.

Java fills these identified needs. JavaScripts can be directly embedded into *MapServer* template files so that separate vector and raster coverages can be requested, integrated, and manipulated in the output interface. Java applications, residing on the client, can be used to enhance control through selection, panning, and zooming of spatial data as well.

THE ForestView APPLICATION

Forest resource management organizations routinely conduct in-place inventories of their holdings. These data are a primary source of information for tactical and operational planning. Given the dominant role of location in such efforts, many organizations manage in-place inventory data using GIS's. The GIS allows manipulation and display of important spatial data layers and query of associated aspatial data: most commonly stand attributes including timber characteristics.

A prototype system for accessing stand examination data via the Internet was built for DNR-Forestry as part of ForNet: ForestView. A simple image map allows the user to choose a particular township of interest within Minnesota. Upon selecting a township of interest, a map of stands in the township is presented in an interface that allows simple GIS tasks to be applied to the data (fig. 1). At present that capability includes choosing between map types (cover types, tree size classes, topography), particular map layers (stand boundaries, inclusions, roads, section lines), as well as panning and zooming of the map. Figure 2 displays the results of zooming in 3x on a particular conifer stand of interest. A backdrop of TM data, produced through a simultaneous call to *ImageView*, provides the means for placing DNR-Forestry holdings within the context of all ownerships; alternative TM dates can be selected as well. Once a stand of interest is located, its inventory attributes can be retrieved with a simple mouse click (fig. 3). To aid in interpreting the attributes, inventory reports include URL's to detailed descriptions and definitions of attribute labels.

The entire ForestView system runs on top of *MapServer* software. Work is currently underway to link this system to a growth model, allowing projection of stand attributes through time. The system is not intended to replace the

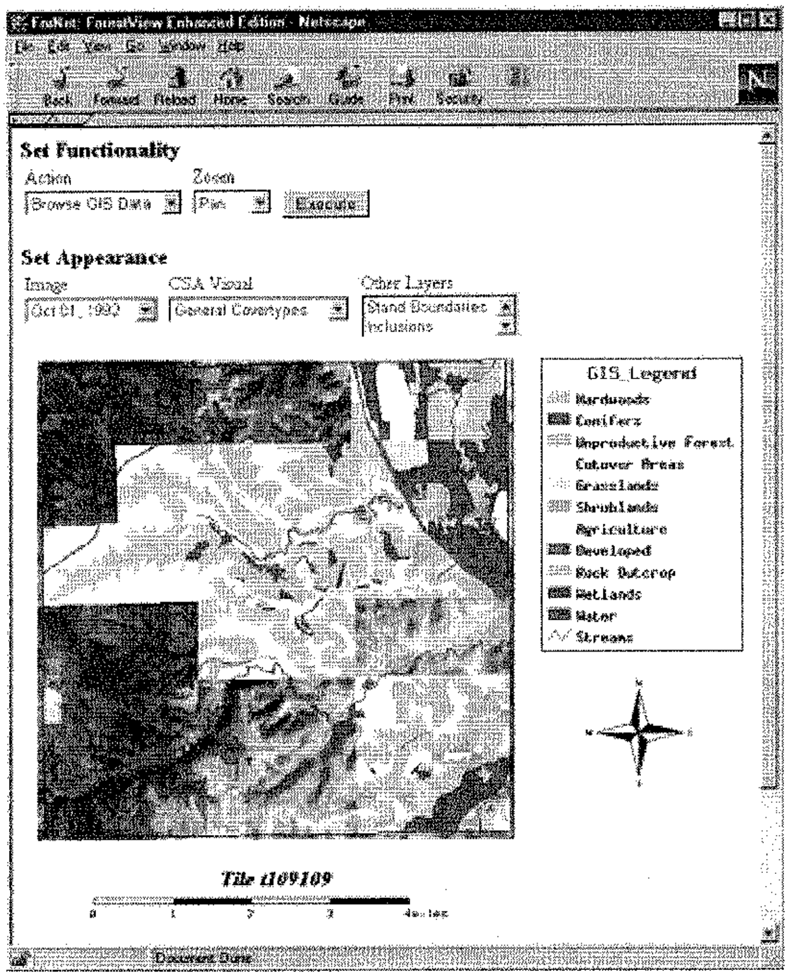

Figure 1.—*ForestView interface showing forest stands in a township with GIS data and a TM backdrop.*

use of GIS's but to instead deploy simple GIS capability to a broader audience of users. Since the application requires only commonly available WWW browser software, technology requirements are much reduced over those of a full-blown GIS. DNR-Forestry has made their entire statewide stand inventory database available to the public via the Internet. The ForestView implementation allows centralized maintenance of GIS data sources, an approach preferred by many organizations. A planned extension of the application will allow updating of spatial and aspatial data by distributed agency personnel through the WWW interface.

CONTINUING SOFTWARE DEVELOPMENT

A current direction for enhancement of the ForNet software tools is the addition of more general analysis capability. For example, *MapServer* can be used to create spatial displays and return attribute data with coordinate information attached. JavaScripts allow user control of the extent for which result data are returned. Server and/or client residing analysis modules can grab the result data and return HTML-formatted output. For simple processing, such functionality can be provided by JavaScripts within *MapServer* template files. More sophisticated requests, such as geostatistical calculations, require invocation of separate applications. Java-based statistical tool kits such as provided by the new XGobi environment (cf. Cook *et al.* 1997) could be easily integrated into the framework described. Basic GIS processing can also be added through such an approach including proximity analysis, overly analysis, and buffering.

Figure 4 displays a simple implementation of these ideas. A *MapServer*-generated interface is used to display data on waterfowl production areas and other relevant layers for western Minnesota (in figure 4 an area of interest has been zoomed in on). Java code allows the user to set a point buffer by clicking a point that is used in combination with a selected buffer distance; the result is the circle shown in figure 4. *MapServer* returns the attributes of the selected features within the buffer area, which are in turn processed by a Java application to summarize, in this

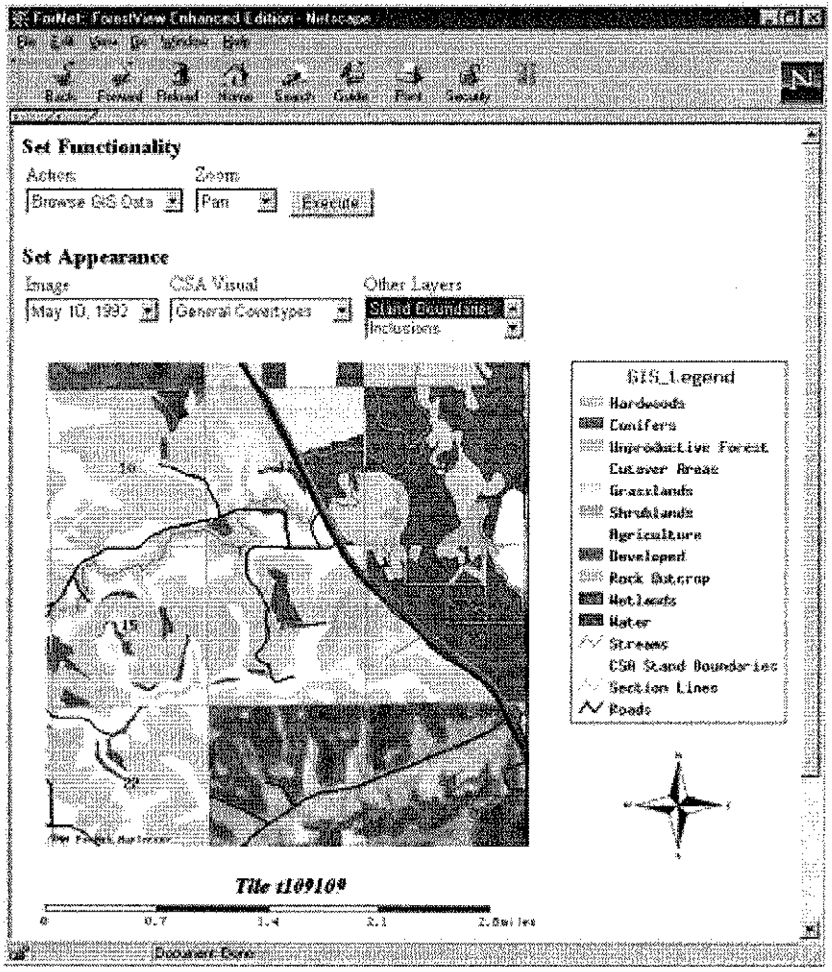

Figure 2.—*ForestView interface from figure 1 zoomed in by a factor with stand boundary, roads, and section layers turned on.*

case, area and perimeter of lakes (cf. scrollable dialog). Analysis capability can be targeted or made quite general within this *MapServer*-centered analysis framework.

SOFTWARE AVAILABILITY

MapServer and *ImageView* are distributed according to the open source concept. Each package requires use of other freeware or public domain code. Code, documentation, and test applications are available at the ForNet WWW site. We hope that open source distribution will encourage others to add functionality to the packages and share those results broadly. This has already happened. Open source also facilitates employing a distributed server environment, important for applications based on, for example, country-wide databases. Two basic Java applications can also be made available that provide integration and additional user interface control.

ACKNOWLEDGMENTS

Research supported by the National Aeronautics and Space Administration through grants NAG5-2698 and NCC-316 and by the Agricultural Experiment Station, University of Minnesota, through project MIN-42-044. Mr. Kirk Bloomquist prepared the figures in the article.

The manuscript was reviewed by Dr. Dietmar Rose and Mr. Hector Franco-Lopez, Department of Forest Resources, University of Minnesota, St. Paul, MN, USA.

LITERATURE CITED

Bauer, M.E.; Burk, T.E.; Ek, A.R.; Coppin, P.R.; Lime, S.D.; Walsh, T.A.; Walters, D.K.; Befort, W.; Heinzen, D.F. 1994. Satellite inventory of Minnesota forest resources. Photogrammetric Engineering and Remote Sensing. 60(3): 287-298.

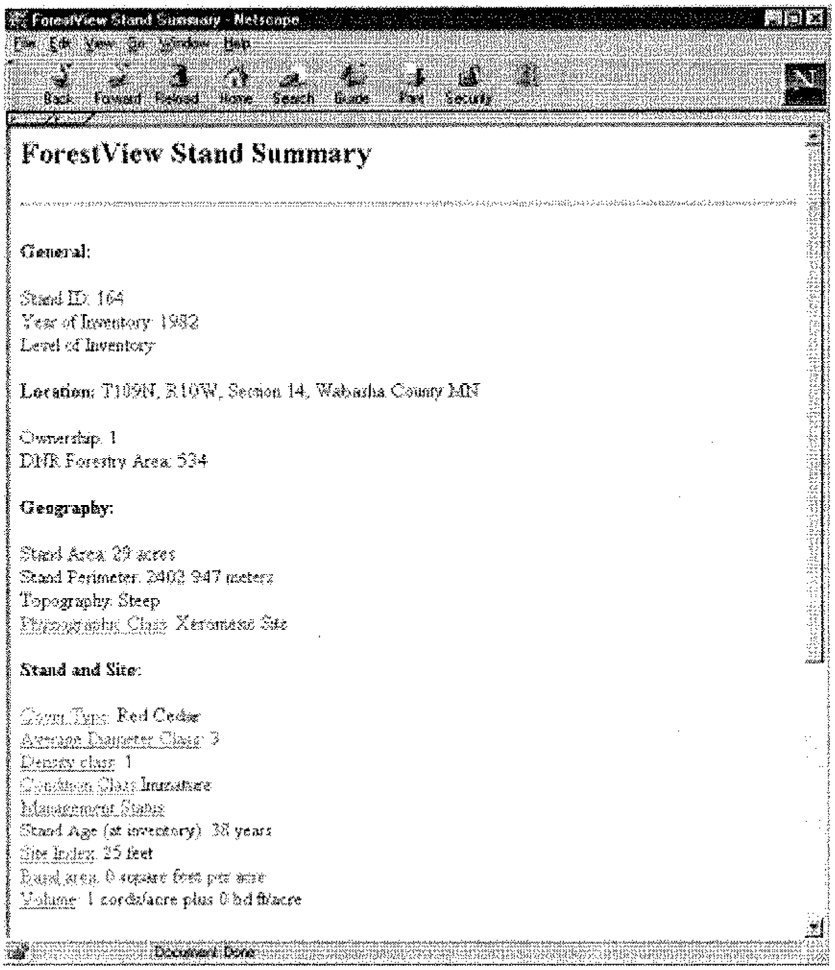

Figure 3.—*Result of clicking on the conifer stand in the center of figure 2: attribute data for the stand coded in HTML.*

Burk, T.E. 1995. Forestry on the Internet: an introduction. Journal of Forestry. 93(5): 6-7, 50-54.

Cook, D.; Symanzik, J.; Majure, J.J.; Cressie, N. 1997. Dynamic graphics in a GIS: more examples using linked software. Computers and Geosciences: special issue on Exploratory Cartographic Visualization. 23: 371-385.

ERDAS. 1994. ERDAS field guide. Atlanta, GA: ERDAS, Inc.

ESRI. 1995. ArcView shapefile technical description. ESRI White Paper Series. 14 p.

Prisley, S. 1996. Westvaco's forest resources information system. The Compiler. 4(3): 4-8.

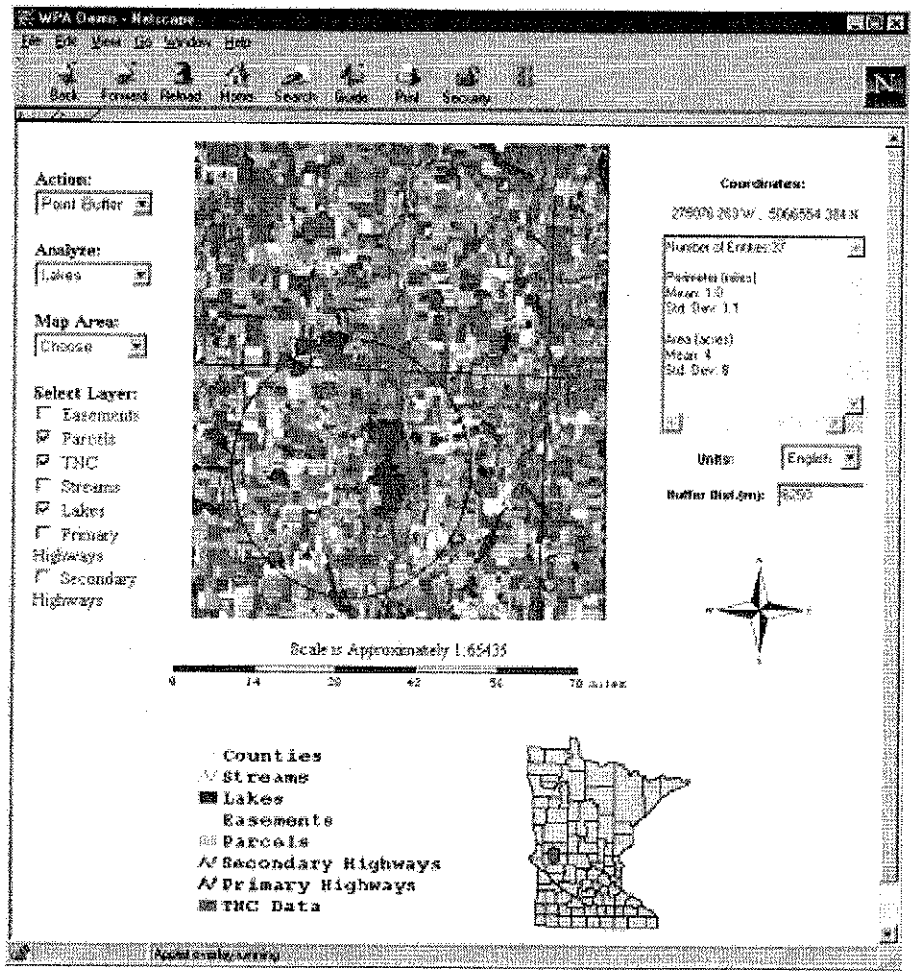

Figure 4.—*Example of analysis capability added to MapServer through Java applications.*