Neural, Parallel & Scientific Computations 15 (2007) 207-220

IMS: a Web-based Map Server for Spatial Decision Support

Salah Juba[1], Vasilis Samoladas[2], Nikos Boretos[1], Ioannis Manakos[1] and Christos G. Karydas[1]
[1]Mediterranean Agronomic Institute of Chania, Makedonias 1, 73100, Chania, Greece
[2]Department of Electronic and Computer Engineering, Technical University of Crete, University Campus – Kounoupidiana, 73100, Chania, Greece

Abstract
WebGIS is becoming prominent in spatial decision support applications, as it allows researchers and stakeholders to benefit from sharing, analyzing and visualizing large, up-to-date geospatial data sets with minimal effort and cost. This paper addresses the integration of open-source, open-standards software packages and state-of-the-art web technology to develop an interactive web mapping portal for spatial analysis. It demonstrates that, open-source software offers a level of flexibility, availability and lowered cost that is typically unavailable with commercial software, while an architecture and design based on open standards ensures system interoperability and data reusability.

The resulting system aims at enhancing collaboration and decision making among researchers and stakeholders in environmental decision-making, being highly accessible, and requiring minimal computing expertise. Where standard functionality is insufficient, the system can be extended via scripting to adapt to emerging needs.
Keywords - WebGIS, interactive web applications, spatial decision support systems, raster algebra, environmental applications

1. INTRODUCTION

Senior managers and decision makers heavily depend on geospatial data, as a vital resource for improving economic productivity, decision-making, and delivery of services. Yet, even today, spatial data and geographic information are under-utilized by *local* administrations and stakeholders. High cost and lack of expertise prevent the incorporation of geospatial data analysis and interpretation in everyday business.

The Internet has become the major carrier of information around the globe, offering increased accessibility and mobility. As such, it is the primary means for distributing geospatial information and data with minimum cost. Beyond data sharing, the integration of the World-Wide Web with Geographical Information System technologies (WebGIS) is becoming prominent, as companies and institutions around the globe realize the significance of WebGIS applications in Spatial Decision Support Systems. Initial applications appeared roughly a decade ago and involved centralized dissemination of maps, first static and later

Received November 8, 2006; final form May 24, 2007 1061-5369 $15.00 © Dynamic Publishers, Inc.

dynamic (allowing pan/zoom as well as primitive layer composition). As the foundations of web technology matured, WebGIS applications could provide more sophisticated cartography and spatial visualization features (Lu et al., 2003; MacEachren, 2001; Kraak & Brown, 2001; Mitchell, 1005; Marshal, 2002).

A challenge for WebGIS remains the interactivity limitations of the basic web platform (HTTP, HTML, Javascript, etc.), in contrast to applications that execute on local workstations. These limitations spurred a number of proprietary technologies that enhanced the interactivity of web applications (Java applets, ActiveX controls, Flash animations, etc.), resulting sometimes in a babel of incompatible, non-interoperable web content. WebGIS designs utilize these technologies to some extent (e.g., Tsou, 2004; Kotzinos & Prastacos, 2001), improving interactivity but also suffering from reduced interoperability. During the past few years, the web platform has been evolving steadily, with the introduction of *standardized* dynamic presentation and interactivity features and the maturation of web browsers.

Open-source software is distinguished from typical proprietary software products in that the source code is available to the user. Access to the source code of GIS tools is important for GIS application developers, since GIS algorithms can be very complex, and proper understanding can greatly influence the quality and performance of modeling and spatial analysis (Mitasova & Neteler, 2002). Open-source practices promote a synergistic effect among developers of different backgrounds and expertise, contributing to the generation of more robust, stable and cost effective software.

In the last decade a great collection of open-source GIS algorithms, digital encodings, image formats, and software packages appeared. However, no scheme for interoperability based on a common format has been successfully produced. As a result of this failure, OGC (Open Geospatial Consortium) was formed in 1994 to develop a new interoperability approach based not on formats, but on open, common software interfaces. OGC activities led to the development of geospatial standards and location-based services (e.g., Asghar et al., 2004).

A major benefit of WebGIS arises in the sharing of geospatial data over the internet. However, certain limitations and restrictions must be addressed (Bothelho et al., 2003, Zaslavsky & Baru, 2003):

- Metadata incompatibilities: identifying and characterizing datasets, as well as specifying subsets of large data sets that need to be shared/ retrieved.
- Data transfer costs: geospatial datasets can be huge, especially if comprised of raster data. Furthermore, despite the increase in network bandwidth over time, the increase in size of spatial datasets matches and often surpasses network speedup.
- Data format proliferation: Many formats have been developed over the years; each company, research agency, or university uses its own, often with insufficient specification. The well-known GDAL library (www.gdal.org) supports over forty popular formats, at least partially.

- Policy restrictions: They are placed on the distribution of geospatial data both because of security or intellectual property concerns, as well as for fiscal ones: frequently, the primary motive for assembling geospatial data is special-purpose, and added costs of making data multi-purpose are hard to justify.

In this work, the design and architecture of the ISOTEIA Map Server (IMS), a web-based spatial decision support system, is presented and discussed. IMS provides data sharing, visualization of geospatial data and spatial decision support services for environmental planning and management (Dragicevic & Balram, 2003, Anselin et al., 2004, Kingston et al., 2000, Sakamoto & Fukui, 2004). IMS uses Map Algebra as a tool for spatial analysis. Map Algebra refers to the use of images as variables in normal arithmetic operations (Eastman, 2003). Raster algebra tools are used to combine image layers; they are used in the creation of multi-criteria suitability maps.

IMS is developed in the framework of ISOTEIA project (INTERREG IIIB/ CADSES Community Initiative). ISOTEIA aims at environmental protection in CADSES area through the establishment of an Integrated System designed to promote best practices in Territorial and Environmental Impacts Assessment (TEIA). The project requirements include the generation of a number of GIS applications, based on alternative scenarios generated by Spatial Decision Support Systems (SDSS) in national and trans-national study-areas, in environmental topics, such as surface water management, ecosystem protection, forest management, industrial siting, irrigation management, water supply optimization, and sustainable agriculture.

The requirements set for ISOTEIA applications dictate that IMS must facilitate geospatial decision support web applications that are ubiquitously available via all popular web browsers, demanding minimal network bandwidth, and requiring no configuration or dependence on additional plug-ins. At the same time, IMS servers must be portable across major platforms and free of software licensing restrictions. The main contribution of this work is that, by using state-of-the-art web development technologies and open-source GIS tools, the IMS design can satisfy the above requirements without sacrificing application functionality or user-interface quality.

2. DESIGN AND IMPLEMENTATION

WebGIS can be classified as fat-client and thin-client. In fat-client systems, a significant proportion (often, the bulk) of data processing happens at the client, whereas the server is primarily responsible for data storage (e.g. Tsou, 2004). By contrast, thin-client systems strive to minimize processing on the client; except for presentation and user interaction, data processing occurs at the server. For IMS, the thin-client approach was adopted, based on the following criteria:

- The system must be accessible on the internet, where users may have insufficient network resources to download and process massive data locally. Only visualization data should be transmitted to the client. Additionally, user interaction must not create unnecessary network traffic (e.g., unneeded page refreshing).

- The system must be accessible from different platforms and environments, such as UNIX and Windows operating systems, as well as all popular web browsers, without dependence on additional software (specialized plug-ins, local applications, etc.) and with zero configuration.
- The IMS server software should be portable across a broad range of server platforms and technologies, with minimal (preferably zero) dependence on proprietary software, in order to allow stakeholders to deploy servers with maximum versatility and minimum cost.
- IMS servers must be scalable, both with respect to the number of concurrent users and in terms of storage capacity and computational performance. At the same time, server administration should be simple, and require only modest technical expertise.

It is not clear that the above criteria can be satisfied without serious compromises to application functionality and user-interface quality. The design that is outlined in this section demonstrates that, by using state-of-the-art web programming, combined with a server that integrates a broad range of open-source tools and applications, it is possible to achieve all of the above goals with currently available WebGIS technology.

2.1 Application Architecture

IMS is architected as a typical 3-tier system, depicted in Fig.1

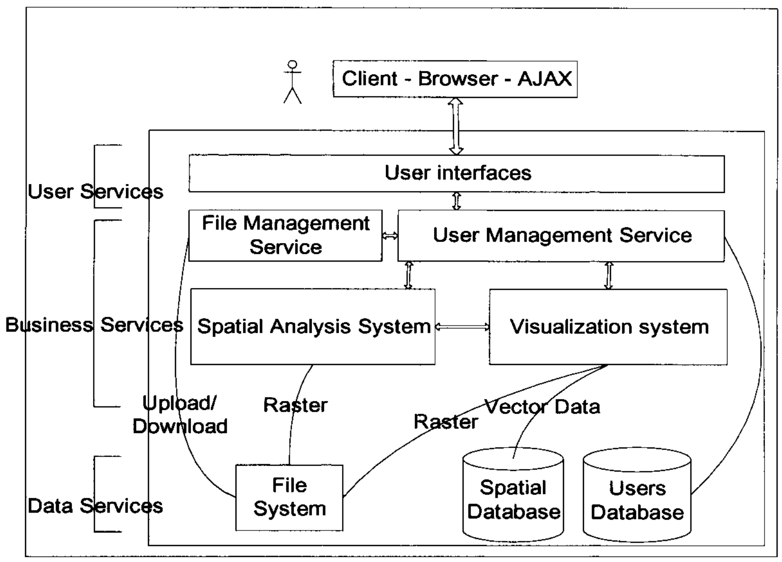

Fig.1: Open source web mapping portal with decision support system

Functionality is distributed in three service tiers, namely User Services, Business Services and Data Services. The User Services tier is created as a typical web application that provides the visual interfaces through which the user interacts with the application. The technology used is Dynamic HTML (DHTML) and AJAX (Asynchronous Javascript And XML). AJAX (which is discussed in more detail later) is a state-of-the-art approach to web application design, which provides superior user-interface functionality and interactivity compared to traditional (i.e. form-based) web applications, based solely on standard web browser features. By adopting AJAX, a portable, friendly, functional, highly-interactive user interface, without resorting to proprietary "applet" technologies (including Java applets) were provided.

The Business Services tier encapsulates the business logic for the entire application. It includes a number of subsystems. The user and session management systems, provide standard authentication, auditing and personalization functionality to applications. The geo-visualization system, which is based on the MapServer open-source software from the University of Minnesota, is responsible for composing image representations of geospatial data, embeddable in HTML pages that can be shipped to the clients over the network. Finally, the data analysis system implements the spatial decision support capabilities in the form of a number of spatial analysis services, notably map algebra processing and spatial query processing over geographic databases. These services are available to the user as tools, accessible through the user interface in a user friendly manner.

The Data Services tier provides services that store, retrieve and update information through a simple data model. The main storage mechanisms are (a) file systems and (b) spatial databases. File systems are used for storing spatial data files in a variety of formats, managing per-user directories and providing format transformation services for spatial data. Spatial databases have a dual role; they store metadata related to files stored in the file system, and they can be used to manage (store, update, search, process) vector spatial data for application-specific purposes.

2.2 System components

The current implementation of IMS is based on the following open-source software packages:

1. Tclhttpd web server - the main skeleton for the portal. It is based on the Tcl/tk script language.
2. AJAX – used to provide interactivity in the user interface.
3. University of Minnesota MapServer - used to provide the visualization service.
4. Numerical array processor (NAP) - used for map algebra computations.
5. Geospatial Data Abstraction Library GDAL – used for data format translation.
6. PostgreSQL and PostGIS – a robust relational database management system (RDBMS) used as a spatial engine for vector data, as metadata repository, and as a backend for user and session management.
7. TclSOAP – a middleware solution used for communication among the web server and compute servers running NAP (for map algebra processing).

8. Miscellaneous packages: pgtcl, nstcl, HDF, CBE JavaScript library, Dbox JavaScript library, freetype library.

These packages are integrated and coordinated by glue code written in Tcl/tk (Welch, 1999, Zerbst, 2002), a mature, highly portable script language. By adopting it, all of the third-party software listed above was integrated with minimum programming effort; Tcl/Tk bindings for all of these packages were readily available. In addition, any future extension of the IMS system will be relatively painless. We will be looking at adding new processing tools (e.g., spatial analysis tools, such as GRASS, or proprietary solutions) as application needs arise. Note also that the current choice of third-party software is not binding: although some of the packages used in IMS implementation are intimately bound to the current code (notably, MapServer), others are easily exchangeable. For example, it should be easy to employ a different database server in the place of PostgreSQL.

2.3 User Interface

The main view presented to the user at any time consists of a map pane and one or more application-specific panes. A view of the map pane is depicted in Fig.2. The map is implemented as an image, created by the visualization service by composing a number of layers, where the user can activate/deactivate layers. Surrounding the map are a set of standard tools (Rinner et al., 2005) for browsing, querying and measuring. The supported browsing functionalities include zoom in/out/previous/next/full, and panning. Querying allows the user to select data appearing on the map, either at a specific location, or within a bounding box drawn by the mouse. Finally, by using the measuring tools, the user can measure the area bounded by a box or the distance between two points. By using AJAX, these functionalities are accessible by using mouse drag and drop operations or just by using single clicks, and do not involve page refresh.

AJAX is a novel approach to coding highly interactive web applications, without relying on proprietary extensions. Instead, AJAX relies exclusively on standard technologies included in all modern web browsers; the Document Object Model (DOM) for dynamic interaction and display, XML and XSLT for data interchange with the web server, XMLHttpRequest for asynchronous retrieval of new content, CSS and XHTML for standard interfaces, and finally JavaScript to combine the other technologies.

In the classic, form-based web application model, each time the user interacts with an HTML form, an HTTP request for a full page is sent to the server, and the user must wait for server processing, page transfer and page rendering. Each of these steps may easily take a few seconds, significantly reducing the user-friendliness of the application.

An AJAX-based application eliminates this coarse level of interaction on the Web, by introducing an intermediary — an AJAX engine — between the user and the server as shown in Fig.3 (Garrett, 2005). The engine is implemented as a JavaScript library executing inside the web browser. It stores “page state”, i.e. information on the current visual content of the page at any given moment. As the user interacts with the page through the browser,

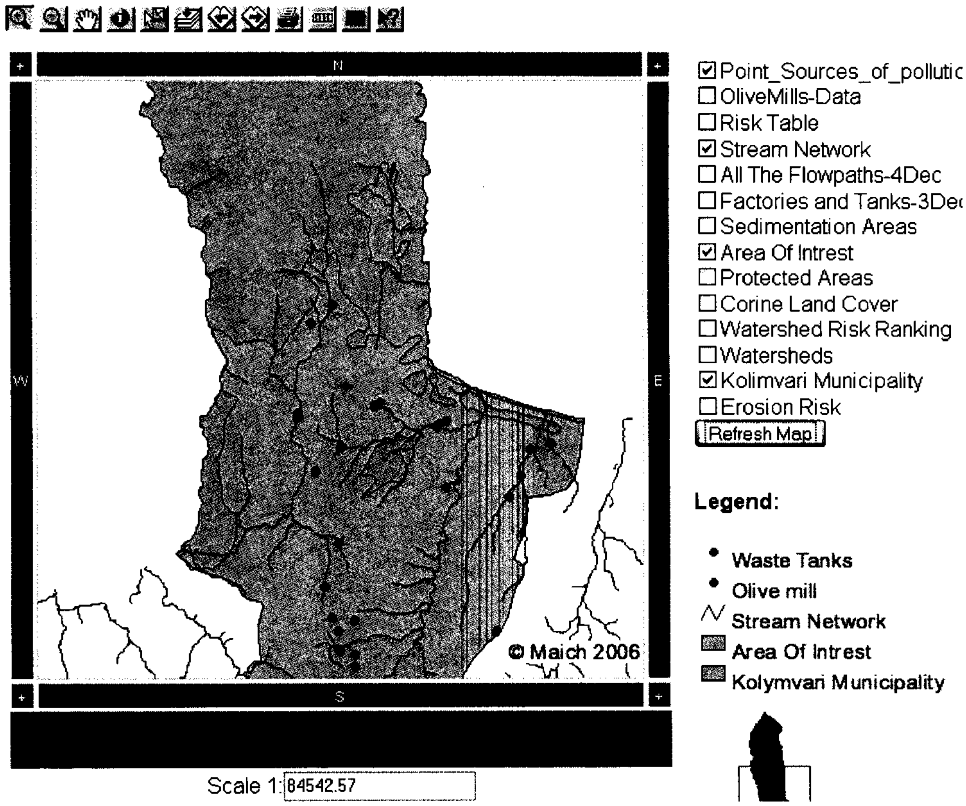

Fig.2: A view of the map pane. The map displayed shows layers from Isoteia case study 6.

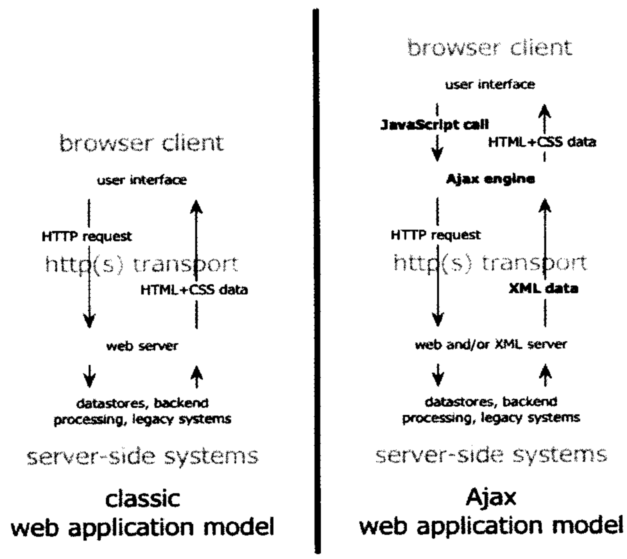

Fig.3: Classic Web Application Model versus AJAX Model

suitable pieces of logic (encoded in JavaScript) are executed, and the page state is altered appropriately. When new information is required from the server, the AJAX engine sends a suitable XML-encoded message and retrieves (instead of a whole new page) only the minimum necessary data needed to update page state. As page state changes, the shown page is modified locally (through the DOM) instead of a full rendering. Furthermore, XML transfers are done asynchronously, so the user can interact with the shown page while XML transfers are in progress.

AJAX is capable of providing functionally rich web interfaces with unprecedented responsiveness. Yet, AJAX-based applications are cross-platform; most major browsers are supported, including Microsoft IE, Safari, Opera and all Mozilla based browsers.

2.4 Spatial data visualization

In order to visualize spatial data, IMS employs MapServer from the University of Minnesota. MapServer is one of the leading packages for web mapping applications, providing feature-rich cartographic output, such as scale bars, legends, reference maps, and labelling. MapServer has the ability to generate thematic maps based on classes and regular expressions. Moreover, MapServer implements the open geospatial consortium implementation specifications for open web services (OWS), namely Web Mapping Service (WMS), Web Feature Services (WFS) and Web Coverage Service (WCS). It also supports GML and provides on-the-fly projection capability.

MapServer can operate in two different modes: MapScript APIs and via a CGI executable. IMS implementation is currently using CGI. In this mode, the rendering depends on two files: the configuration file "map file" and the HTML template. The map file is an ASCII file which defines the data objects used to produce the map. The template file is an HTML file with tagged replaceable parameters enclosed within square brackets. MapServer processes each replaceable parameter in the template file and returns the result.

Although the MapScript interface allows for more flexibility compared to CGI, CGI is more adequate for IMS purposes, also more advanced business logic are coded outside of MapServer (Greenwood, 2002).

2.5 Data services and data sharing

The effortless sharing of geospatial data is a major requirement of IMS. In order to support collaborative decision making among stakeholders, IMS data services were developed to overcome most technical impediments to the accessibility of shared data. In IMS applications, spatial data is expected to be massive, collected by a variety of sources and represented in a multitude of formats. In order to allow effective sharing, IMS provides users with a number of data services.

IMS data storage is either file-based, or utilizes spatial relational databases. For file-based data, there are the following basic services: the file management service, the data translation service and the data composition service.

The file management service allows IMS users to upload, download, delete and inspect spatial data files in a variety of formats. File transfer to and from the server uses standard web mechanisms. Inspection applies to raster images and presents the user with file related metadata, without accessing the bulk of the file.

The file translation service is essentially a front-end to the Geospatial Data Abstraction Library (GDAL) for raster data translation and the OGR Simple Features Library for vector data translation. Together, these two libraries support translation between over 40 different geospatial data formats.

A number of spatial database servers can also be supported for storing spatial data. Currently, only simple feature vector data can be stored in spatial databases. IMS implementation is based on the PostGIS RDBMS, although using different RDBMS products should be straightforward. There is no explicit data entry interface to populate the database; rather, this feature serves primarily to support application-specific data storage. To provide this functionality, the user must create a spatial database schema into which the data is to be stored. Data is stored either through specialized forms, or graphically, by selecting the location of new records on a map, using the mouse. Once populated, a spatial database can be queried, or composed with file-based spatial data, to construct new views.

The data composition service is used to combine together spatial data files or database-stored records. Data from the selected files or database tables is used to define a number of layers, as well as queryable data attributes, via a form-based interface. The final product of this process is a new map, which can then be loaded into the visualization service, to be explored and/or processed by the spatial analysis tools.

2.6 Spatial data analysis

Spatial processing in IMS is currently heavily geared towards processing raster data via map algebra. Map algebra operations can be implemented effectively by using array processing libraries (e.g., Matlab, NAP, octave). The data flow implementing map algebra is as follows: the arrays are extracted from raster files, manipulated by the array processing library and the results are converted back to a raster format. The raster result is then fed to the visualization service.

The map algebra service provides a plethora of unary and binary arithmetic, logical and fuzzy-logic operators. Also, it provides for several classification methods, where each raster point in a raster map is classified to an appropriate category according to its value, and the categories are shown properly colorized by a selected pallette.

All map algebra functionality on raster data is currently implemented by the Numerical Array Processor library (NAP). NAP is a powerful and efficient tool for processing n-dimensional arrays, similar in essence to other array-processing languages such as APL, J, IDL and Matlab. NAP has a number of innovative features including support for grid-oriented data based on continuous spatial coordinates (Davis, 2002).

Although NAP is an efficient array processor, map algebra operations often require considerable CPU power, in the order of several seconds even for simple operations on medium-size datasets. In a setting where map algebra operations are computed on the same machine as the web server, it would be difficult to support more than a few concurrent users without severe drop in system responsiveness (Hawick, 2003). To alleviate this situation, IMS data analysis services are architected on a client/server model: processing is performed on a number of dedicated compute servers, (possibly) separate from the web server machine. Jobs are dispatched to these machines using the Simple Object Access Protocol, SOAP (Laurent et al., 2001). This is an XML-based, standardized middleware protocol, used by a number of vendors, including Microsoft (in the .NET platform), IBM (in their Web Services product line), and Sun (through the Java Enterprise specifications). Also, SOAP is similar to widely adopted standards for Grid computing (Globus and the Open Grid Software Architecture standard).

Currently, IMS is somewhat naïve in dispatching jobs to compute servers. There is little attention paid to load balancing, optimization of data transfer to the processing nodes, caching etc. In the future, the integration of IMS to Grid platforms and other distributed processing frameworks will be investigated.

3. ENVIRONMENTAL DECISION SUPPORT WITH IMS

IMS's development was motivated by the requirements of Territorial and Environmental Impact Assessment (TEIA), within the ISOTEIA project. However, its functionality extends to all applications that deal with geographically oriented information in space. IMS's capability of integrating and merging data layers of different natural variables by a stepwise application of complicated functions enables multi-criteria, multi-temporal, and multi-scale data handling. Applications, such as irrigation management, precision agriculture, soil erosion risk assessment, environmental resource management, land use mapping, crop yield monitoring, regional and rural development planning, as well as modeling with physical models, can be supported. Furthermore, IMS enables the cooperation of different task groups sharing a common geospatial data base by the implementation of a multidisciplinary project. Results from one group are made instantly available to parallel working groups in an interactive way.

IMS is designed to support computation intensive analytical processes, such as the ones that require handling of huge geospatial datasets. For instance, Guerrero et al. (2006) report enormous computation requirements by a technique for land cover mapping in the Mediterranean based on object-oriented classification of high-resolution satellite images on a multi-level and iterative segmentation and classification basis. Such situations are common in environmental applications when extrapolating the results of pilot study areas to the whole area of interest. To achieve reasonable computation times, this type of computation should be performed in parallel on high end supercomputers or computing clusters. In this context, IMS is well suited to act as a web portal to wide-spread interfaces of high-performance computing platforms, such as computational grids. These interfaces are made available to IMS users as additional spatial analysis services, through scripted extensions. IMS's function as a portal is to assemble and preprocess the input dataset, dispatch and monitor the execution asynchronously, and visualize the results.

4. CONCLUSIONS AND OUTLOOK

This paper presented IMS, a web-based spatial decision support system which integrates a broad array of free, open-source, open-standards GIS/SDSS software tools. Careful integration of these packages and the use of state-of-the-art web programming techniques resulted in a powerful and robust WebGIS. This work affirms previous research, in that, WebGIS systems depending solely on standards-compliant web browsers, can be valuable tools in spatial decision support. They exhibit satisfactory interactivity, and are advantageous for spatial data dissemination, advanced spatial analysis, and modeling, compared to standalone systems.

IMS can substantially enhance collaboration and decision making among participants of environmental projects and involved stakeholders. It requires minimum computing expertise, is ubiquitously accessible, and offers rapid dissemination, effortless manipulation and high-quality visualization of spatial data. Where the standard functionality of IMS may be insufficient for the task at hand, the ease of integrating additional functionalities and libraries via scripting, yields a system that is adaptable to emerging future needs.

A number of technical issues are under investigation. Currently, IMS is somewhat naïve in dispatching jobs to compute servers. There is little attention paid to load balancing, optimization of data transfer to the processing nodes, caching etc. In the future, the integration of IMS to Grid platforms and other distributed processing frameworks will be investegated. Additionally, while Tcl and Tclhttpd offer a robust threading model, the thread safety of the numerous extensions used has not been investigated, currently opting to use the Tcl event loop instead of threads.

IMS is still evolving, and is currently being evaluated in practice and against its objectives; it is planned to contribute to the enhancement of interactive stakeholder training and participation on geoinformation analysis and interpretation on local, regional, national, and international level.

ACKNOWLEDGEMENTS

Project part-financed by the European Union (European Regional Development Fund) within the INTERREG III B CADSES Neighbourhood Programme. The second author was partially supported by a project co-funded by the European Social Fund & National Resources – EPEAEK II - PYTHAGORAS.

REFERENCES

1. Anselin, L., Kim, Y. W. & Syabri, I. (2004). Web-based analytical tools for the exploration of spatial data, *J. of Geographical Systems*, 6(2), pp. 197-218.
2. Asghar E., Mohammadi, A. Aien & Mohammadi, H. (2004). Developing an internet GIS application using GML technology, *Proc. XXth ISPRS Congress.*
3. Botelho, L. R., Strauch, J. C. M. & de Souza, J. M. (2003). WISE: An architecture for geo-referenced data integration, *Proc. Next-Generation Geospatial Information Systems (NG2I'03).*
4. Coddington, P. D., Hawick, K. A. & James, H. A. (1999). Web-based access to distributed high-performance geographic information systems for decision support, *Proc. Hawaii Int'l Conf. on System Sciences (HICSS'99).*
5. Davis, H. (2002). The NAP (N-Dimensional Array Processor) extension to Tcl, *Proc. 9th Annual Tcl/Tk Conference.*
6. Dragicevic S. & Balram, S. (2004). A Web GIS collaborative framework to structure nd manage distributed planning processes, *J. of Geographical Systems*, 6(2), pp. 133-153.
7. Eastman, J.R. (2003). *IDRISI Kilimanjaro Tutorial.*
8. ESRI (2003). Spatial Data Standards and Interoperability, *http://www.esri.com/library/whitepapers/pdfs/spatial-data-standards.pdf*
9. Garrett, J. J. (2005). Ajax: A New Approach to Web Applications, *http://adaptivepath.com/publications/essays/archives/000385.php.*
10. Greenwood, R. W. (2002). Using MapServer to Integrate Local Government Data, *Proc of the Open source GIS - GRASS users Conference.*
11. Guerrero, I., Tanase, M., Manakos, I. & Gitas, I. (2006). A semi-operational approach for land cover mapping in the Mediterranean, *Proc. 26th EARSeL Symposium on New Developments and Challenges in Remote Sensing* (to appear).
12. Hawick, K. A., Coddington, P. D. & James, H. A. (2003). Distributed frameworks and parallel algorithms for processing large-scale geographic data, *Parallel Computing*, 29(10), pp. 1297-1333.
13. Kingston, R., Carver, S., Evans, A. & Turton, I. (2000). Web-based public participation geographical information systems: An aid to local environmental decision making, *Computers, Environment and Urban Systems*, 24(2), pp. 109-125.
14. Kotzinos, D. & Prastacos, P. (2001). GAEA, a Java-based map applet for performing GIS on the web., *SAMS*, Vol.6, pp. 593-606.
15. Kraak, M. J. & Brown, A. (2001). Web Cartography development and prospects, *New York: Taylor and Francis Inc.*
16. Laurent, S. St., Johnston, J., Dumbill, E. (2001). Programming Web Services with XML-RPC., *ISBN: 0-596-00119-3 O'Reilly.*
17. Lu, C. T., Kou, Y., Wang, H., Shekhar, S., Zhang, P. & Liu, R. (2003). Two web-based spatial data visualization and mining systems: Mapcube & Mapview, *Proc. Next-Generation Geospatial Information Systems (NG2I'03).*
18. MacEachren, A. M. (2001). Cartography and GIS: Extending collaborative tools to support virtual teams, *Progress in Human Geography*, 25(3), pp. 431-444
19. Marshall, J. (2002). Developing Internet-Based GIS Applications, *GIS India*, 11(1), pp. 16-19.
20. Mitasova, H. & Neteler, M. (2002). Freedom in geoinformtion science and software development: a GRASS GIS contribution, *Proc. Open Source Free Software GIS-GRASS users Conference.*
21. Mitchell, T (2005). Web Mapping Illustrated, *ISBN: 0596008651 O'Reilly.*
22. Rinner, C., Raubal, M. & Spigel, B. (2005). User interface design for location-based decision services, *Proc. (CD-ROM) 13th Int'l Conference on Geoinformatics.*
23. Sakamoto, A. & Fukui, H. (2004). Development and application of a livable environment evaluation support system using Web GIS, *J. of Geographical Systems*, 6(2), pp. 175-195.

24. Stojanovic, D. & Djordjevic-Kajan, S. (2001). Internet GIS Application Framework for Location-Based Services Development, *Proc. 7th EC-GI.*
25. Takatsuka, M. & Gahegan, M. (2002). GEOVISTA Studio: a codeless visual programming environment for geoscientific data analysis and visualization, *Computers & Geosciences*, 28(10), pp. 1131-1144.
26. Tsou, M. H. (2004). Integrating web-based GIS and image processing tools for environmental monitoring and natural resource management, *J. of Geographical Systems*, 6(2), pp. 155-174.
27. Turban, E (1995). Decision support and expert systems: management support systems, *Englewood Cliffs, N.J., Prentice Hall.*
28. Welch., B. B. (1999). Practical programming in Tcl and Tk, *Third edition, ISBN 0-13-022028-0, Prentice-Hall Inc.*
29. Zaslavsky, I. & Baru, C. (2003). Grid-enabled mediation services for geospatial information, *Proc. Next-Generation Geospatial Information Systems (NG2I'03).*
30. Zerbst, C. (2002). Web Applications With the Tcl Web Server, *http://www.linux-magazine.com.*