Anning River Valley and to conduct a series of spectral and spectral-spatial classification for the mapping of snail habitat. Classification results will be validated through field work. To further improve the classification, we will assess the use of ancillary ground data and ecological knowledge in the classification. This will be followed by an exploration of higher resolution remotely sensed data for snail habitat identification. If the results obtained are satisfactory, we will investigate the possibilities to extend the techniques we developed for Anning River Valley to other parts of the endemic areas in China including the Three Gorges Area.

The next step in applying the GIS-based spatial network analysis and modeling is to replicate the approach in a prospective mode to other endemic areas. Possible sites are the endemic areas in Meishan county roughly 100 km west of Chengdu. The SIPD in conjunction with the local anti-endemic stations has been active in these areas in 1996 and 1997. Activities have involved infection surveillance in human and animals, snail surveys, mouse bioassays, human water contact surveys, and both praziquantel treatment of selected groups and some environmental control activities. With the models developed in the Xichang study area, the research can move directly into the site-specific calibration of the models using the data generated by the SIPD from 1996 forward, and also the intervention history at the sites in the calibration activities. One issue that needs further research is how to model complex interaction of temporal and spatial aspects of transmission in a unified way.

ACKNOWLEDGEMENTS

This project has been partially supported by an NASA-Ames Joint Research Interchange (NCC2-5102), pilot funding from the NIEHS Mutagenesis Center at Berkeley (5 P30 ES 01896-20 ZES1), a University of California Pacific Rim Research Grant, a grant from the Ministry of Health, China and a grant from the National Natural Science Foundation of China (49825511).

REFERENCES

[1] Beck, L., M. Rodriguez, et al. (1994). "Remote sensing as a landscape epidemiologic tool to identify villages at high risk for malaria transmission." *Am. J. Trop. Med. Hyg.* 51(3): 271-280.

[2] Davis, G., R. Wilke, et al. (1997). "Cytochrome Oxidase I-based phylogenetic relationships among the Hydrobiidae, Pomatiopsidae, Rissoidae, and truncatellidae (Gastrophoda: Prosobranhcia: Rissoacea)." *Malacology* 39(in press).

[3] Davis, G. M., Z. Yi, et al. (1995). "Population Genetics and Systematic Status Of Oncomelania Hupensis (Gastropoda, Pomatiopsidae) Throughout China." *Malacologia* 37(1): 133-156.

[4] Devlas, S., G. Vanoortmarssen, et al. (1996). "Schistosim - a microsimulation model for the epidemiology and control of schistosomiasis." *American Journal Of Tropical Medicine And Hygiene* 5: 170-175.

[5] Dister, S., L. Beck, et al. (1993). The use of GIS and remote sensing technologies in a landscape approach to the study of Lyme disease transmisison risk. *Proceedings of GIS '93 Symposium*, Vancouver, BC.

[6] Glass, G., B. Schwartz, et al. (1995). "Environmental risk factors for Lyme disease identified with geographic information systems." *Am J Public Health* 85: 944-8.

[7] Gu, X. G. (1990a). Effect of Schistosomiasis control programs in Xichang experimental fields, Sichuan Province, Sichuan Institute of Antiparasite Diseases.

[8] Gu, X. G. (1990a). Epidemiological investigation of schistosomiasis in Minhe and Hexing villages, Xichang, Sichuan Province, Sichuan Institute of Antiparasite Diseases.

[9] Levin, S., B. Grenfell, et al. (1997). "Mathematical and computational challenges in population biology and ecosystems science." *Science* 275(5298): 334-342.

[10] Li, Y. Q. (1990). Snail ecological observation in Xichang experimental fields, Sichuan Province, Sichuan Institute of Antiparasite Diseases.

[11] Li, Z., P. Yuan, et al. (1990). "Identification of Distribution Area of Oncomelania by Remote Sensing Technique." *Acta Scientiae Circumstantiae* 10(2).

[12] Linthicum, K., C. Bailey, et al. (1997). "Detection of rift valley fever viral activity in Kenya by satellite remote sensing imagery." *Science* 235: 1656-59.

Martens, W., T. Jetten, et al. (1997). "Sensitivity of malaria, schistosomiasis and dengue to global warming." *Climatic Change* 35: 145-156.

[14] Spolsky, C., G. Davis, et al. (1996). "Sequencing methodology and phylogenetic analysis: Cytochrome b gene sequence reveals significant diversity in Chinese populations of Oncomelania (Gastropoda: Pomatiopsidae)." *Malacologia* 38(1-2): 213-221.

[15] W.H.O. (1993). The Control of Schistosomiasis, Second Report of the WHO Expert Committee, World Health Organization.

[16] Washino, R. and B. Wood (1994). "Application of remote sensing to arthropod vector surveillance and control." *Am. J. Trop. Med. Hyg.* 50(6): 134-144.

[17] Wood, B., R. Washino, et al. (1991). "Distinguishing high and low anopheline-producing rice fields using remote sensing and GIS technologies." *Preventive Medicine* 11: 277-288.

[18] Zhou, Y. (1998). *GIS-Based Temporal and Spatial Modeling of Schistosomiasis Infection for Local Transmission Control*. PhD Dissertation, School of Public Health. University of California, Berkeley.

[19] Zhou, Y., D. Maszle, P. Gong, R.C. Spear, and X. Gu (1996). "GIS-based spatial network models of schistosomiasis infection." *Geographic Information Sciences* 2(1-2): 51-57.

A Model of GIS Virtual Machine

Li Zhang[*], Bin Li[†] and Hui Lin[§]

[*]MRF Geosystems Corporation, Calgary, AB, Canada, T2P 0T9
[†]Central Michigan University, Mount Pleasant, MI 48858
[§]The Chinese University of Hong Kong, Shatin, Hong Kong

Abstract
This paper presents a model of Geographic Information System Virtual Machine. The GIS VM is a software architectural model that facilitates the development of Web-based GIS applications. It is a three-tiered system. Java applets or ActiveX objects, with Web browsers as the container, form the presentation tier that interacts with the user. The core of the GIS VM is the servers in the middle and third tiers. The middle tier includes server components for basic GIS functions and data discovery. The third tire mainly contains the spatial data access server. In addition to these basic servers, the GIS VM includes a model object manager (MOM) that establishes linkages with external models and other application programs. The MOM can also extract metadata about models and carry out the execution process on a user's behalf.

I. INTRODUCTION

The Internet can serve not only data but also sophisticated computation. Using the Web to perform spatial analysis, modeling, and advanced visualization is no longer just a novel idea. The technological foundation is in place. This paper discusses how we can adapt the technological advances to developing Web-based geo-computational services.

The model we are going to present is a GIS Virtual Machine (GVM). It is a software framework allowing different functional components to interoperate through the Internet to provide geo-computational services. It is "virtual" because all the parts are software components that reside on different locations. And the connections among these parts are established through "virtual" object buses. The GVM is a machine in the sense that it is able to produce information product by processing spatial data.

Although this is the first time the term GIS Virtual Machine is used, researchers have begun exploring distributed geo-computational services recently. Internet-based user interfaces has been a popular study for improving spatial data access and usability (Li, *et al.*, 1996). Lin and Zhang (1996) demonstrated a web-based GIS catalog browser for distributed spatial data retrieving. Li and Zhang (1997) presented a model of component oriented GIS, which explores the re-organization of GIS in light of the maturity of distributed object technology. Li (1999) further suggested that the Internet and distributed object technology would elevate existing GIS onto what he calls Geographic Information Services. Li (1998) also demonstrated the idea through a prototype of Geographic Computational Services, which allows a user to perform advanced spatial autoregressive modeling through the World Wide Web. These researches have shown the unprecedented opportunity and challenges for GIS research and development.

The aim of GIS VM is to make GIS a ubiquitous technology through the Web. Realizing this goal requires both conceptual and technical solutions. The fundamental questions have to do with what constitute the core of the GVM, how the components relate to each other, how the GVM integrates with the Web and existing legacy software systems. We hope this paper would contribute to the understanding of these technical and conceptual challenges.

We will first describe the conceptual framework of the GIS Virtual Machine. The discussion focuses on the three-tier structure, the essential server components, and the mechanisms for interoperations. The key server components, i.e., the catalog server, data access server, and the model manager are described in further detailed. We conclude with a brief discussion on the benefits of the GIS VM and anticipated problems in research and development.

II. CONCEPTUAL FRAMEWORK OF THE GIS VIRTUAL MACHINE

The idea of GIS Virtual Machine (GIS VM) was inspired by SUN Microsystem's "Java Virtual Machine"

1082-4006/98/0401~2-23$3.00

(JVM), which specifies the architecture for Java components from different platform to work together via the Internet (Lindholm and Yellin, 1996). Similar to JVM, GIS VM is a "soft" computer for geographic information processing that can be implemented in a variety of ways. It is an abstractive machine designed to be implemented on top of the Java Platform and HTTP server to provide a uniform object interface to GIS-oriented applets and applications. The architecture of a GIS VM is partitioned as a three-tiered system that includes presentation components, application logic components, and data components (Figure 1). Presentation components manage user interactions and make requests for geo-processing application services by calling the middle-tiered components. Usually they are byte-coded Java applets embedded in HTML documents which can run on any computers with the Java runtime environment. The middle-tiered GIS application components perform geo-processing services and make requests to databases and other resources. Data components handle database connections and data transactions.

Geo-processing servers in the middle tier and the spatial data access server in the third tier form the core of a GIS VM. Geo-processing servers provide the basic functions for spatial data analysis and visualization. Analytical and cartographic functions typically found in current GIS programs are included in the middle-tier. In addition, a catalog server is needed to facilitate data discovery. The catalog server is able to allow the user to manage and extract metadata from spatial data catalogs that are distributed on the network. In the third tier, a spatial data access server connects spatial data stores on the network. It provides a transparent access to heterogeneous databases through a common interface. These servers in the middle and third tiers are distributed objects. They can be resided in different computers. They work together through object buses such as those from CORBA and DCOM.

Note that in figure 2, we added a model manager. This server component is responsible for managing advanced spatial modeling tasks. It bridges external modeling software and the GIS VM. In a typical scenario of spatial modeling, the user would request from the model manager the parameters, data formats, and other background information necessary for constructing and executing the model. Once the model is constructed, the user would send the instruction to the model manager who would carry out the steps for running the model. To allow such flexibility, all the models must be also treated as distributed objects. We therefore also call the model manager the Model Object Manager, or MOM. In implementation, legacy modeling systems can become distributed object through wrappers.

The GIS VM is designed to work with the World Wide Web. The integration requires a connection between two primary protocols, the HyperText Transfer Protocol (HTTP) that typically the client used to communicate with the server, and the Object Request Broker (ORB) protocols that govern the interoperations among distributed application objects. Such connection is realized through the Inter-Internet ORB Protocol (IIOP) (Object Management Group, 1998).

With the above structure in place, we can illustrate the operation of a GIS VM through a typical senario of a web-based GIS application.

1). A user starts a browser to connect to the GIS service provider or data provider (who owns and manages a GIS VM in their Intranet, and provides services via its HTTPD server).
2). Java applets are transferred from the Web server to the users' desktop computers.
3). The user formulates the requests and makes the selections.
4). The user submits the requests to the server.
5). An agent component passes the requests to specific servers in the middle-tiered unit (which may or may not reside on the same machine as the HTTPD server). Alternatively, the requests are directly sent to the corresponding server components.
6). The middle-tiered servers connect with the data servers which usually run on computers in the service provider's firewall.
7). The Data server queries the spatial data warehouses based on users's request and return the results to the middle-tiered servers.
8). The middle-tiered servers start processing the data. When all the tasks are completed, the results are sent back to the user's desktop computer.
9). The browser refresh its window and display the result.

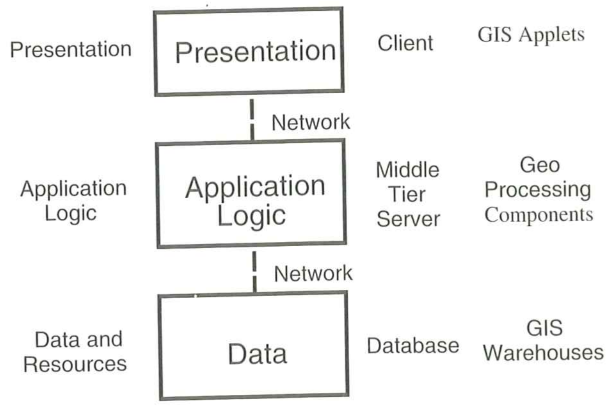


Figure 1. The architecture of a three-tiered GIS VM

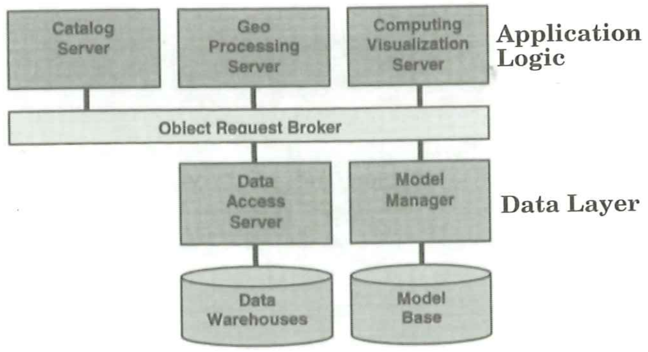


Figure 2. The core servers in a GIS VM

Figure 3 depicts the framework of a Web-based GIS VM. In the following sections, we will discuss some of the key components in the GIS VM. They are the catalog server, the data access server, and the components for application integration (application gateway and model object manager).

III. CATALOG SERVICE IN THE GIS VIRTUAL MACHINE

One of the basic services a GIS Virtual Machine provides is catalog service. It is the component that helps users locate geographic data on the Internet through spatial data catalogs. Spatial data catalog is defined by the Open GIS Consortium as a collection of entries (catalog entry), each of which describes and points to a feature collection (Buehler and McKee, 1996). A catalog supplies a subset of the Metadata that describes the content, quality, condition, origins, and other characteristics of the data (FGDC, 1996). At a minimum, a spatial data catalog includes the name for the feature collection and the location of the data stores. Locating and extracting relevant information from data catalogs on the Internet are fundamental operations in a GIS VM.

On the client side is a Java-enabled Web browser. The client interface is a Java applet embedded on the HTML page the user sees when connected to the Web site for catalog service. Once downloaded to the user's computer, the applet will be invoked and establish a connection with the catalog server. User requests will be then transferred to the server that will search the catalog database and return the results of the query.

A spatial data catalog can be implemented with a number of approaches ranging from simple flat files to high performance relational data base management systems. Figure 4 shows a relational scheme where catalog, catalog entry, catalog registry, and feature collections are mapped to relational tables and are linked through the primary and foreign keys. The catalog registry contains information about feature collections, e.g., their names, the URLs of the catalog, and the data source name (DSN) of the catalog. Given the name of a feature collection, a user submits an SQL query to the back-end database server, which retrieves the location of the catalog and sends the information back to the user. From the catalog table, the user is able to find the location of the feature collection and the corresponding metadata. If the data meet the needs, the user will then download them for processing and analysis.

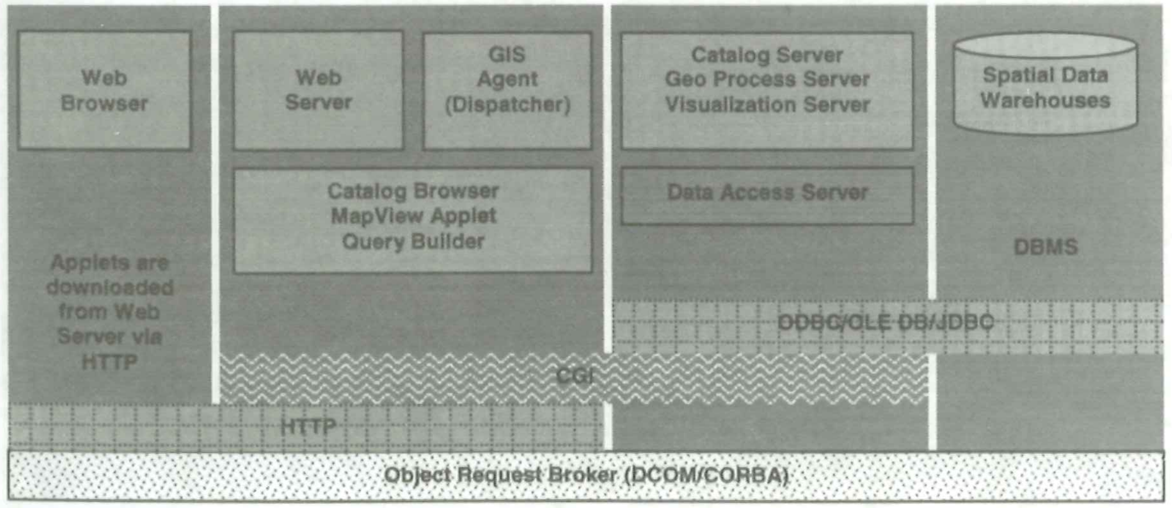
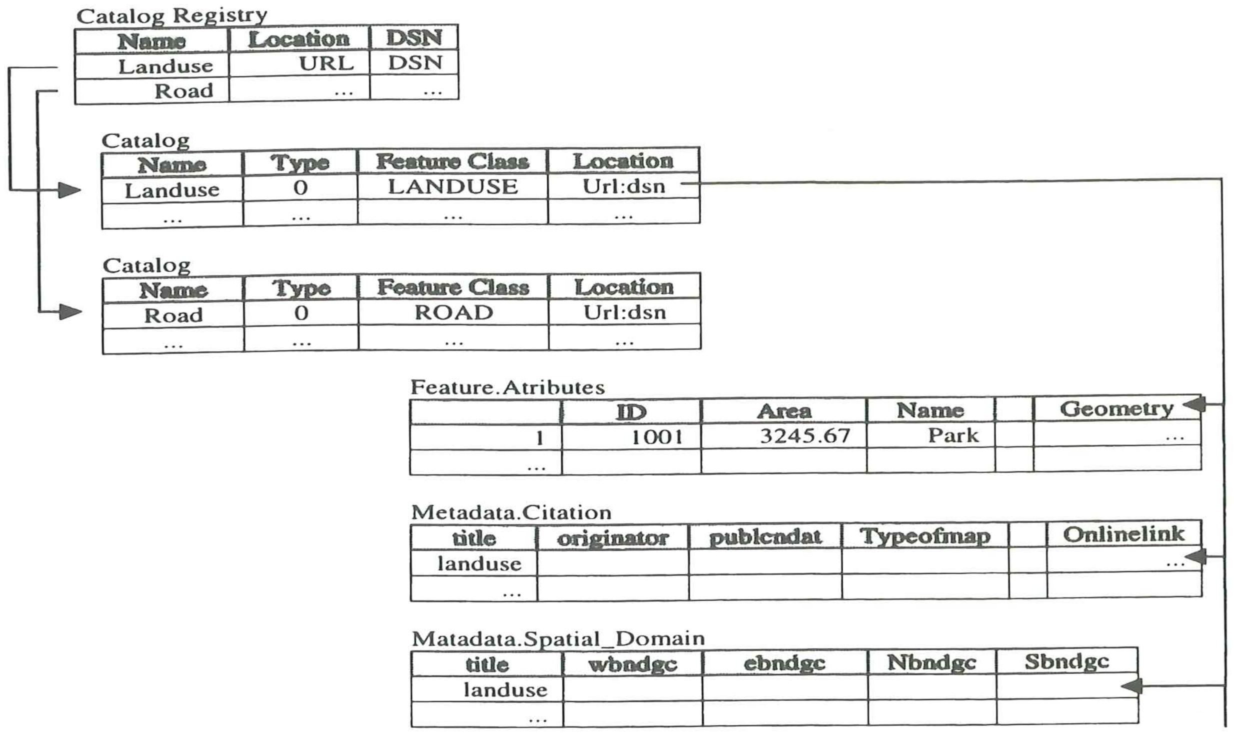


Figure 3. Integration of the Web and GIS



Catalog Registry

Name	Location	DSN
Landuse	URL	DSN
Road	...	...

Catalog

Name	Type	Feature Class	Location
Landuse	0	LANDUSE	Url:dsn
...	...	...	...

Catalog

Name	Type	Feature Class	Location
Road	0	ROAD	Url:dsn
...	...	...	...

Feature.Atributes

	ID	Area	Name		Geometry
1	1001	3245.67	Park		...
...					

Metadata.Citation

title	originator	publcndat	Typeofmap		Onlinelink
landuse					...
...					

Matadata.Spatial_Domain

title	wbndgc	ebndgc	Nbndgc	Sbndgc
landuse				
...				

Figure 4. A sample scheme definition of catalog management

IV. HETEROGENEOUS SPATIAL DATABASE ACCESS

Spatial data are collected, processed, and stored in different ways. Using them together for a single project has been a costly process that typically requires tedious data conversion. Facilitating transparent access to heterogeneous spatial databases on the Internet is another fundamental task the GIS Virtual Machine must carry.

Moffatt (1995) outlines three solutions to heterogeneous database access. These strategies seek to enable transparent access by developing a common *interface*, or a common *gateway*, or a common *protocol*. A *common interface architecture* provides a common Application Interface (API) at the client side to access multiple back-end databases. Microsoft Open Database Connectivity (ODBC) is an example of such architecture. SUN Microsystem's JDBC is anther example (Hamilton and Cattell, 1996). A common gateway architecture replies on a gateway to manage the communication with multiple databases. A common protocol approach uses a common data transfer standard to achieve heterogeneous access. Neither approach is as well received as the common interface architecture. In our GIS VM, we developed the database access service based on ODBC and JDBC.

ODBC is based on the Call Level Interface (CLI) specification developed by the SQL Access Group. It consists of three components. An *application* performs processing, calls ODBC functions to submit SQL statements, and to retrieve results. The driver manager is responsible for loading the drivers on behalf of the application. The third component, the driver, processes ODBC function calls, submits SQL requests to a specific database, and returns results to the application. ODBC can be used in different configurations, depending on the database to be accessed. Usually, a one-tiered architecture is used for flat files, e.g., xBase files. A two-tiered architecture is adopted when the SQL database and the application are not resided on the same computer on the network. A three-tiered configuration uses a gateway to communicate with remote databases, where the ODBC driver passes requests to the gateway which in turn sends them to the DBMS. In our research, we used Sybase SQL Anywhere as the back-end DBMS, which allows us to adopt either two-tiered or three-tiered configuration.

One of the major shortcomings with ODBC is the lack of direct connection through the Internet. JDBC overcomes such a problem thanks to the platform neutral and network-oriented language JAVA (Hamilton and Cattell, 1996). JDBC facilitates a two-tiered access to database servers through the Internet. Performance, however, can be improved with a three-tiered configuration, which we adopted in the prototype. As depicted in figure 3, the Data Access Server is located in the middle-tier of the GIS VM. It can be implemented on top of JDBC, ODBC, or OLE/DB, depending upon the data consumers in the middle-tier. For example, if the Catalog Server, which accesses the catalog registry database and spatial data warehouse via the Data Access Server, is implemented with Java, then JDBC should be used on the Data Access Server. Otherwise, if the implementation language is C/C++, then ODBC should be the choice. The prototype catalog browser we implemented is a Java applet that connects the Java-based catalog server in the middle-tier which uses the JDBC interface to access the catalog registry through the JDBC/ODBC bridge driver. The Visualization Server, a part of the GIS VM as shown in Figure 2, is implemented in C and therefore the connection between it and the Data Access Server is built on top of ODBC.

V. MODELING AND VISUALIZATION SERVICES

In addition to the basic services for catalog and data access, the GIS VM also provides the infrastructure to incorporate software programs for spatial modeling and visualization, which are typically implemented as independent entities without direct Web interfaces. The key components of such infrastructure are the Application Gateway (AG) and the Model Object Manager (MOM).

Real world problem solving often requires advanced spatial modeling and visualization. Software programs that carry out such tasks are often tailored to specific domains. Good examples include three-dimensional fluid visualization, soil erosion modeling, multivariate clustering, and spatial autoregressive modeling. While they are often used for specific applications—hence it is not appropriate to package them all in a single system, these modeling and visualization programs share common needs for such basic services as catalog, data access, and spatial query, which the GIS VM provides. It is logical then to make a linkage between these legacy systems with other components in the GIS VM so as to utilize the common services more efficiently. In addition, most of these specialized programs are not Web-enabled. The Application Gateway eliminates the need to develop customized Web interface for both existing legacy systems and newly developed modeling modules. With the Application Gateway, it is possible to construct Web-enabled modeling and visualization systems for a specific application project.

The modeling and visualization processes are facilitated by the Model Object Manager (MOM) in the GIS VM. When a model is registered with the GIS VM, its parameters, execution procedures, data formats, and other information about the model, or model metadata, are stored in the Catalog Server. The MOM will bring the metadata to the user, based on which the modeling process is constructed, typically in a form of scripts. Upon receiving requests from the user, the MOM will execute the scripts that may involve interactions with various components in the GIS VM and the specific models.

VI. DISCUSSION AND CONCLUSION

Despite its complexity, the GIS VM will have profound impact on the GIS community. The benifits are very clear for the end-users. Applications based on a GIS VM provide a common and hypertext-based geographic processing environment on the Internet/Intranet. Adaptation of GIS VM may influence how end users use and purchase GIS software and how vendors (middle users) provide information services. GIS VM will facilitate distributed decision making ranging from trip planning to site selection and to ecological system modeling. For application developers, GIS VM provides an environment to assemble quickly application software by using the reusable software components and the services provided by the GVM. More importantly, GIS VM will promote the formation of geographic service providers, who supply value-added information services to on-line customers. GIS VM allows services from different providers to work together, resulting in more specialized development and higher quality services.

Among the technical challenges, the key problems are in performance and the detail specification of the Geoprocessing server. Geo-computational tasks often involve highly complex calculations that require considerable amount of time. It may not be realistic to guarantee synchronous response. Asynchronous service requires a new set of design, e.g., scheduling and callback service. In some cases, parallel computation may need to be incorporated in the GIS VM to overcome bottlenecks.

What should be the basic functions to include in the geo-processing server remains a debatable issue. Despite years of research on the basic functions of GIS, there is no agreement on what should be the primitive functions for GIS. This is particularly so for vector-oriented GIS. The evolutionary nature of information technology adds to the difficulty in defining such a primitive set. The solution perhaps is to make the structure sufficiently flexible so new functional components can be easily added to the core. The object-oriented infrastructure presented in the GIS VM should be able to accommodate such a need.

ACKNOWLEDGEMENT

This paper is based on work partially supported by the Research Grants Council of Hong Kong Government under RGC Earmarked Research Grant No. CUHK 318/95H. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessary reflect the views of the Research Grants Council.

REFERENCES

[1] Buehler, K. And L.McKee, 1996. *The Open GIS Guide*. OGIS TC Document96-001

[2] FGDC, 1996. *Specification for Digital Geospatial Metadata*.

[3] Hamilton, G. And R. Cattell, 1996. *JDBC: A Java SQL API*. Sun Microsystem.

[4] Li, Bin, 1999, A component perspective on geographic information services, *Cartography and Geographic Information Science*, (in press).

[5] Li, Bin, 1998 "A Model of Geographic Computation Services," *Geoinformatics'98*, Berkeley, CA: CPGIS, pp. 153-166.

[6] Li, Bin and Zhang, Li, 1997. "A model of component-oriented GIS," *GIS/LIS'97,* Bethesda, Maryland: American Society for Photogrammetry and Remote Sensing, (CD-ROM).

[7] Li, C.S., David Bree, Adrian Moss, James Petch, 1996. Developing Internet-Based User Interfaces for Improving Spatial Data Access and Usability. *Third International Conference/Workshop on Integrating GIS and Environmental Modeling* (CD-ROM. http://www.ncgia.ucsb.edu/conf/SNTA FE CD ROM/program.html).

[8] Lin, Hui and Li Zhang, 1996, "a web-based GIS catalog browser for distributed spatial data retrieving", *Proceedings of Geoinformatics'96 Wuhan*, pp.272-279.

[9] Lindholm, T. and Yellin, F., 1996, *The Java Virtual Machine Specification,* http://java.sun.com/docs/books/vmspec/.

[10] Moffatt, C., 1995. Designing Client-Server Application for Enterprise Database Connectivity. *Third International Conference/Workshop on Integrating GIS and Environmental Modeling* (CD-ROM). http://microsoft.com/msdn/library/technote/hetdb.htm.

[11] Object Management Group, 1998, *CORBA/IIOP 2.2 Specification*, http://www.omg.org/corba/corbaiiop.html.

The Configuration and Implementation of a Hybrid 3-D GIS for Urban Data Management

Xinhua Wang and Armin Gruen

Institute of Geodesy and Photogrammetry
Swiss Federal Institute of Technology (ETH) Zürich
ETH Hönggerberg, CH-8093 Zürich, Switzerland

Abstract

With the development of modern cities, 3-D spatial information systems (SIS) are increasingly required for spatial planning, communication systems and other applications. The geometric information to be used in a spatial information system usually includes two types: vector data (such as buildings, traffic ways, waterways, trees, DTM, etc.) and raster data (such as orthophotos, original images from aerial or still video cameras, etc.). Considering the availability and advantages of relational database technology, it is an important task to develop a data structure which integrates vector and raster data.

In this paper, a self-developed 3-D data structure (V3D) is presented, in which the geometrical, topological, texture and thematic information is defined. Also, the configuration and implementation in a relational database will be investigated and we will report about a prototype system, the CyberCity Spatial Information System (CC-SIS).

I. INTRODUCTION

The generation and visualization of 3-D city models became an important issue in the recent past due to the increasing demand for a realistic presentation of the real world. 3-D models facilitate the processes of city planning, communication system design, control and decision making, tourism, especially in urban areas. Different applications may require different data types and manipulation functions. The geometrical information to be operated on in a 3-D city system usually includes two types of data: vector data (e.g. man-made objects) and raster images. An appropriate data model should not only represent the geometrical information, e.g. shape, length, area etc. but also implicitly or explicitly describe the topological relationship between geometrical objects, such as adjacency relations, link relations, in/out relations, positional relations. In the case of texture mapping, it also must have the ability to manipulate raster images. The complexity of spatial objects and the variety of data types, especially 3-D objects and images as two completely different data types, makes it a challenging task to develop a 3-D spatial model and data structure for the purpose at hand.

At the Institute of Geodesy and Photogrammetry, ETH Zürich we are involved in a project, "Integration of Raster Images and 3-D Objects into Geodatabases". Two important research topics treated by our group are: (a) the generation of the topology of 3-D objects by using photogrammetric tools, and (b) the investigation of the data model and the development of a system to manage the vector data and raster images based on the relational database technology. The former problem has been addressed with our CC-Modeler (CyberCity Modeler) system (Gruen, Wang, 1998). In this paper, we will present a technology for management of data, which supports the following applications:

- Photorealistic presentation with possibilities for navigation through the 3-D city model
- Abilities to create, store, design, analyse and query city objects

The goal of this paper is to present our self-developed data structure as well as the implementation based on relational database technology for integrating 3-D vector data and raster images. In section 2 the data set for the 3-D city model will be addressed. A 3-D city model data structure will be presented in section 3. In section 4 the issue of implementation based on relational database technology is addressed. Finally, an application prototype is presented.

II. DATA SET

The implementation of a 3-D city information system depends on the application purposes. There are usually three types of data sets involved:

- Digital Terrain Model (DTM)
- Objects, e.g. buildings, roads, waterways, etc.
- Original images or orthoimages

1082-4006/98/0401-29$3.00