ASPRS Annual Conference, Denver, CO, May 23-28, 2004

SQL SUPPORTED SPATIAL ANALYSIS FOR WEB-GIS

Jun Wang **Jie Shan**

Geomatics Engineering
School of Civil Engineering
Purdue University
550 Stadium Mall Drive, West Lafayette, IN 47907

ABSTRACT

Spatial analysis is a fundamental function for geographic information systems. Although it is popular in many commercial desktop GIS packages, this capability is very limited in a Web-GIS environment, especially when relational database is used for geospatial data management. This paper proposes and prototypes a methodology to enhance spatial analysis capabilities of Web-GIS by extending and enhancing the standard query language (SQL) in relational database. The paper first discusses the suitability of available SQL standards for complex spatial analysis. Unlike many recent studies, we have selected OGIS SQL as our spatial query standard to be enhanced with advanced spatial analysis capabilities. OGIS SQL standard contains a set of spatial data types and functions that are crucial for spatial data querying. In our work, OGIS SQL has been implemented in a Web-GIS based on open sources. Supported by spatial-query enhanced SQL, typical spatial analysis functions in desktop GIS are realized at the client side. These functions include distance, buffer, intersection, map overlay and feature fusion. User specific data are also supported. In this way, the spatial analysis can be implemented for spatial data from both server side and client side. The client side is programmed in Java. Two types of query interfaces are provided in client side, one is command line mode for experience users to input SQL directly; the other is the visual mode for common users. In the view mode, SQL can be generated by drag-and-dropping icons. The server side is backed on open source databases: mySQL, PostgreSQL/PostGIS and java serverlet technology. Geographic Markup Language (GML) is tested to return query results to client side and export spatial data.

INTRODUCTION

The rapid development of database and Internet technology has changed ways to access, manage and analyze geographic data. Two distinct evolutional changes are happening and are brought into attention in both academia and industry. First, during the past years, commercial relational database management system (RDBMS) has been adopted to store both attribute information and geometric shapes (Shekhar, 1999). As a result of this migration, standard query tools in RDBMS such as the structured query language (SQL) becomes a common interface for most commercial GIS packages. SQL, which is often customized in a user-friendly graphic interface, is used to conduct queries based on the aspatial attributes of the geographic data. The second evolutional change is the integration of GIS and the Internet technologies. The Internet is being used as a channel and platform to transfer, browse and analyze geographic data. This brings much potential to make GIS information and tools finally open to the public (Peng and Tsou, 2003). Current popular GIS software packages have great spatial analyses captivity, but they are usually designed for GIS specialists, and require a relatively long learning curve to use these tools. Moreover, such software are not designed to be used in Internet environment. Although several web GIS products from major GIS vendors are available to the public, their spatial analysis functions are rather limited compared with desktop GIS packages.

Several technical challenges are emerging as an echo to these evolutions in technology. First, a synergy is needed in handling spatial and apsatial data. Standard SQL in RDBMS needs to be adapted and expanded to query spatial data in a similar way as to the query of aspatial information. For this purpose, spatial analysis functions need to be integrated into the standard SQL formulation and context. The spatial-enhanced SQL should be as intuitive as the standard SQL and as powerful as current commercial GIS packages in terms of spatial analysis and query. The second challenge is to enable public users to benefit the power of GIS through the Internet. Much success has been achieved in allowing the society to browse the published geographic information and conduct very useful database search, such as driving direction finder provided by many internet service providers. However, much more effort is to be made to introduce advanced spatial analysis functions, especially at the level of county, city and community.

Our work is motivated by the above observations. In this paper, a web GIS prototype is designed to investigate the capacity of using spatial SQL to meet the needs of spatial analyses functions in the web GIS environment. Several aspects of this prototype system, such as spatial SQL, users group support, and spatial data import/export are discussed in the paper. A visual query interface based on the Open GIS Consortium (OGC) standards is provided to help users to express their queries with spatial SQL. The prototype is implemented with open-source software.

SPATIAL SQL

Spatial database can be considered as an extended RDBMS that can handle spatial data, support spatial operations and conduct spatial analyses. For this objective, the standard SQL in RDBMS needs to be extended and enhanced to incorporate spatial data. Theoretical formulation and justification on this need are discussed by (Egenhofer, 1999). Lin and Huang (2001) report a prototype implementation that intents to define additional spatial operations under the context of standard SQL grammars. Given the complexity of the problem and the flexibility in defining grammar and semantic expression, a set of rules or standards need to be followed. In the past years, different spatial enhanced SQL have been created. The Open GIS Consortium (OGC, 1999) has published "OpenGIS Simple Features Specification for SQL", a standard that specifies an object model for extending an RDBMS to support spatial data in SQL92 with Geometry Types environment. Our work reported in this paper is based on the OGC standard.

OGC Standard

A geographic feature can be represented by a certain type of geometry object. OGC standard has proposed to expand the standard SQL with geometry type environment to support spatial data. Figure 1 shows the geometry model used in OGC standard. Each object is associated with a Spatial Reference System. Geometry, Curve, Surface, MultiCurve, and MultiSurface are defined as non-instantiable classes. They define a common set of methods for their subclasses. Point, LineString, Polygon, GeometryCollection, MultiPoint, MultiLineString, and MultiPolygon are instantiable classes. Spatial objects can be created from these classes. In a spatial database, spatial type columns can be referenced as other common database column types such as Number and String.

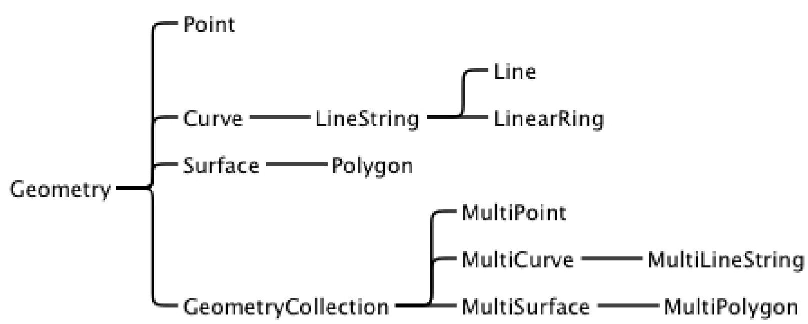

Figure 1. Geometry model of OGC standard

Operations for calculation, query and retrieval of spatial properties of such defined spatial objects need to be thereafter defined to support spatial analysis. These functions/methods associated with Geometry object can be grouped into three categories:

- Basic functions that return spatial properties of a geometry: Dimension(), GeometryType(), SRID(), IsEmpty(), IsSimple(), Envelope(), Boundary(), etc.
- Set operators that test spatial relationships between two geometry objects: Equals, Disjoint, Intersects, Touches, Crosses, Within, Contains, Overlaps, Relate, etc.
- Spatial analysis functions that create new geometries from existing ones: Distance, Buffer, Convexhull, Intersection, Union, Difference, SymmDifference, etc.

For database users, there is another type of functions that convert geometries between various formats that will be used frequently: GeomFromText(), GeomFromWKB(), AsText(), AsBinary().

Spatial Query Example

The basic form of spatial query is similar to the standard SQL: "SELECT FROM WHERE" and nested queries are allowed. Below is an example of finding the gas stations within one-mile range of the street Northwestern Ave.

Query: Find gas stations in 1 mile range from Northwestern Ave.
SELECT Gas.name
FROM Street AS St, Gasstations AS Gas
WHERE **Overlap**(Gas.Shape, **Buffer**(Street.Shape, 1)) = 1 AND St.name= 'Northwestern'

The result returned from the sample query shows in Figure 2: yellow line represents Nothwestern road; points are gas stations. It should be noted that there are spatial analysis or operation imbedded in the simple spatial SQL query statement. First, a one-mile buffer needs to be created around Northwestern Ave. An overlay operation will then allow for a spatial check on the gas stations that fall within the one-mile buffer. Traditional SQL query there is used to find the Northwestern Ave. street, gas stations, and report the query results. Most common queries like "find who/what where" can be intuitively expressed in SQL, and typical analysis function in desktop GIS, such as distance, buffer, intersection, overlay and feature fusion, can be realized with spatial SQL in the database level which is not dependent on the external GIS analysis package (Shekhar and Chawla 2003).

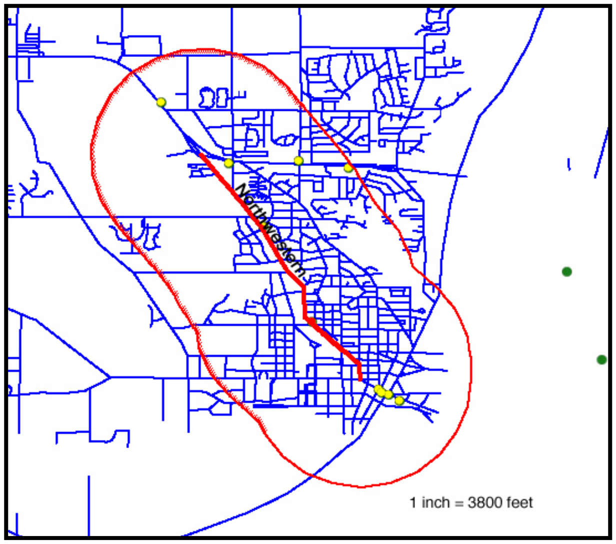

Figure 2. Result of the sample spatial query

WEB GIS PROTOTYPE

This study develops a prototype web GIS with spatial SQL functions. A three-tier structure is adapted as is shown in Figure 3. The middle tier is the web service middleware, developed with Java servlet technology, which enables different user groups to access and manage the entire system through a simple web browser. The map service supports DBMS connection to multi databases, including both relational database and spatial database. Besides establishing connection between users and databases, the map service also provides spatial data import-export function, and can exchange the data with other web servers under certain customization.

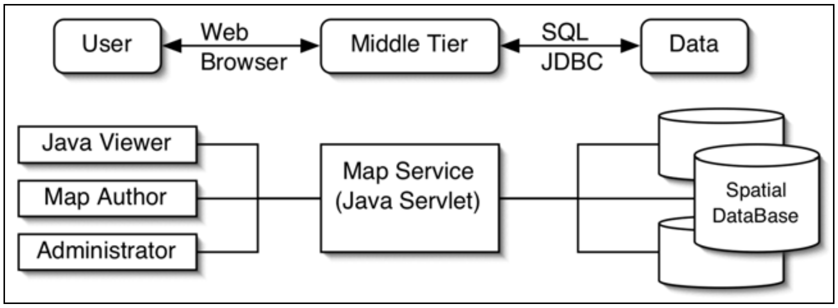

Figure 3. Architecture of the prototype web GIS

User Groups Support

The prototype web GIS groups users to three categories based on their roles and permissions they are granted. Their properties are summarized as follows:

- Browser: information viewer of web GIS. This is the most common capability for web GIS users. The customized Java applet displays spatial data in web browser and supplies the interface to access spatial analysis functions. The information viewer is not allowed to customize any graphic user interface.
- Map Author: designer of web GIS application, who will pump GIS data into spatial database, maintain existing databases and design and publish web-GIS applications.
- Administrator: IT technician of web-GIS, who will set up master database, add/remove database servers, manage user accounts and privileges, deliver security policies, etc.

Web service middleware supplies different interfaces and functionality to different user groups, however, all of them access the service through the web browser.

Spatial Data Import/export

The prototype web GIS supports ESRI’s shapefile and MapInfo MIF files, which are the two most popular vector data formats. When users upload these files to spatial database, map service middleware will create the table in user-desired database server and populate attribute data. The spatial data are imported into appropriate type of spatial columns in the same table. If the database server doesn’t have spatial data extension, the spatial data will appear as binary blob in the table. A sample data sheet is shown in Table 1. The spatial column “GEOM” stores the line features in street layer into multilinesstring data type in the spatial database.

Table 1. Sample data of street layer in spatial database

Id	Label	Speedlimit	Geom
0	'HARRISON'	25	'MULTILINESTRING((-86.9217399999998 40.4202299999997,-86.9167300000008 40.4202800000003))'
1	'RUSSELL'	25	'MULTILINESTRING((-86.9189100000003 40.4240900000004,-86.9194000000007 40.4233999999997,-86.9193899999991 40.4220399999995))'
2	'STATE'	25	'MULTILINESTRING((-86.9189100000003 40.4240900000004,-86.9195799999998 40.4240900000004))'

There are several different ways to export spatial data in this prototype system. Spatial data can be exported to shapefile format, or as the results of SQL query, be returned to Well-Known Text (WKT) format defined by the OpenGIS specifications. As an alternative, GML format can also be used as a standard for data import/export. GML (Geographic Markup Language), developed by the Open GIS Consortium, is “an XML encoding for the modeling, transport and storage of geographic information including both the spatial and non-spatial properties of geographic features.” GML 3.0 is the current version (OGC, 2003). GML provides a mechanism to allow GIS data exchange on web more easily without the trouble to transform file format among different systems. We tested to use GML to return query results to client side. The only problem with GML is its size; it is several times larger than shapefile.

Query Interface Design

Spatial analysis usually is a very complicated process, which may lead to a long, complex and nested SQL statement. Several visual tools have been designed to help a user to define the spatial query without having a professional knowledge on SQL (Blaser and Egenhofer, 2000; Kaushik and Rundensteiner, 2001). However, it is still difficult to use these tools, because they cannot incorporate the attribute data or SQL statements produced by the tools are not compatible with the OGC standard. Our prototype study provides both command line and graphic interfaces for SQL query.

Command line mode is designed for experienced users. They can input SQL commands directly and the query result is returned in table format, with "show on map" function that displays the spatial column in the result on the map. Figure 4 is an illustration of command line based spatial query interface where lakes larger than 1000 acres are being queried and selected.

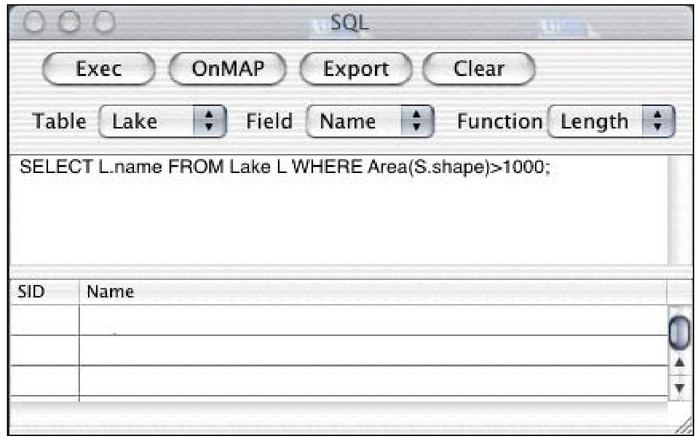

Figure 4. SQL Command line mode

For most GIS users, a graphic and more user-friendly interface needs to be designed to help them build the query without much knowledge of SQL. The interface needs to be simple and easy to implement in web GIS environment. To assist our development, several commercial desktop GIS packages are investigated. One of them is ESRI's ArcGIS 8.3, which provides three separated query interfaces, "Select by Attributes", "Select by Location", and "Select by Graphics" to perform the common spatial analysis function. It is very easy to use on desktop GIS, but on the web GIS environment, we have to combine these three interfaces together. For example, for the query "find gas stations in 1 mile range from Northwestern", a typical process in ArcGIS would be as follows. We first use "Select by Attributes" with the condition: Name="Northwestern" in street layer, and use then "Select by Location", choose "select features from gas station layer that are within one mile distance of the selected feature in the street layer". In this process, the "use selected features" option has to be checked, and "apply buffer to the feature" is checked and set value to 1 mile. For the more complex query, users have to go through the three interfaces several times, with each based on the previous processing. In desktop system, there is no problem to store the intermediate results. User can stop at any step and resume it later. However, SQL doesn't provide such kind of mechanism. The query has to be finished in one statement. This is a critical requirement in the web environment. An efficient user interface needs to be designed to performance these three functions together.

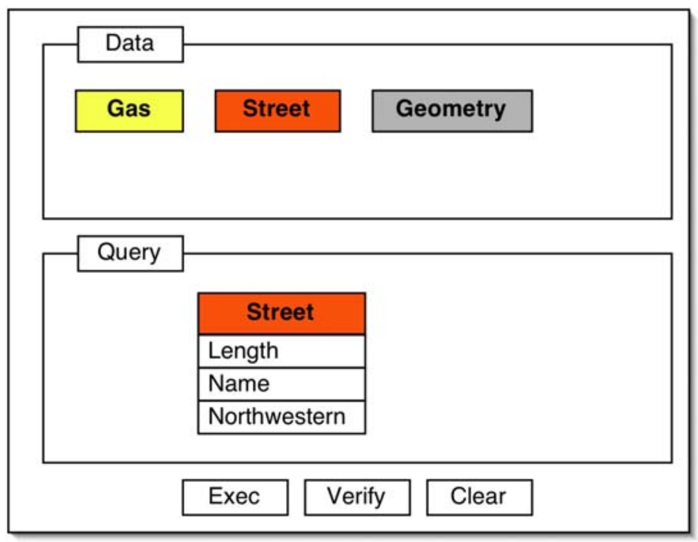

Figure 5. Layout of visual query interface

Figure 5 shows the layout of the visual query interface. The data panel shows all the available tables in the project. The geometry object is explicitly included to allow user to define object type such as lines and polygons from coordinates directly. User can drag-and-drop these icons into the window to form the query. The basic element of query builder is shown in Figure 6. And the relationship between different elements can be defined with the arrow. The sample query "Find gas stations in 1 mile range from Northwestern?" is shown in Figure 7.

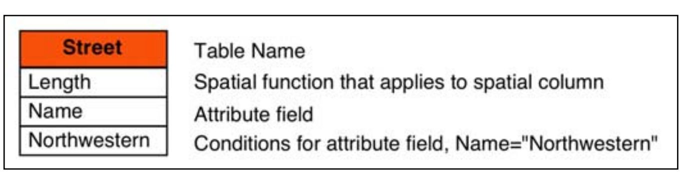

Figure 6. Element of visual query interface

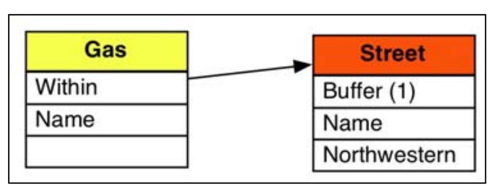

Figure 7. Examples of visual query interface

IMPLEMENTATION

The entire prototype study is implemented with open-source software. MySQL 4.1 (MySQL, 2004) and PostgreSQL/PostGIS (PostGIS, 2004) are selected as the backend spatial databases. Both of them support spatial extensions based on the OGC SQL specification. Its basic spatial data handling capabilities include storing and retrieving spatial data, spatial indexing. MySQL provides some OpenGIS-standard type functions that can test relations between minimal bounding rectangles (MBRs) of two geometries. They include MBRContains, MBRWithins, MBRDisjoint, MBREquals, MBRIntersects, MBROverlaps, MBRTouches. In some case, these functions are very useful to speed up the complex spatial processes, since comparing the spatial relationship between two MBRS is much faster than that between two complex geometries. Because these functions are not OGC standard compatible, we only tested them but did not use them in the prototype. PostGIS add support for geographic objects to the PostgreSQL object-relational database. PostGIS includes full OGC standard support, it is our main spatial database, and most spatial analysis is performed through this engine.

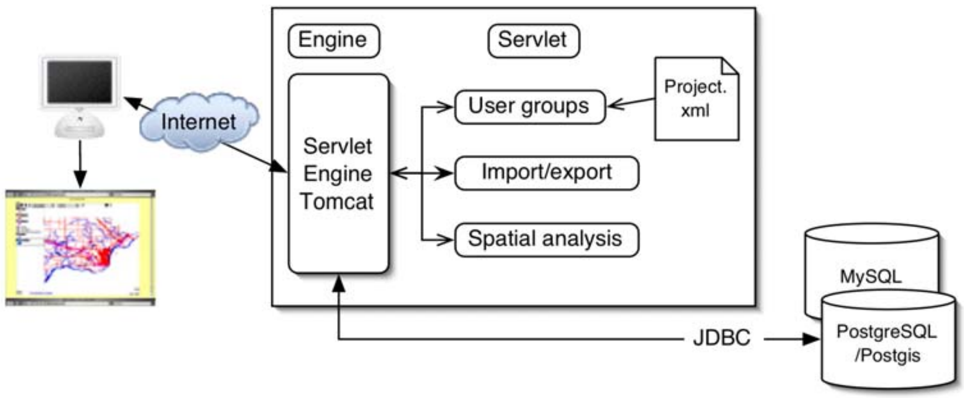

Figure 8. Components of the map service tier

The map service middleware is developed with Java servlet technology. Tomcat 4.2 (ApacheGroup, 2003) is used as servlet container. It is a free, open-source implementation of Java Servlet and JavaServer Pages technologies developed under the Jakarta project at the Apache Software Foundation. Java servlet provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and for accessing existing business systems (Coward, 2001). Its component-based, platform-independent method for building Web-based applications makes it ideal to develop web service middleware. The JDBC technology is used to provide cross-DBMS connectivity to any SQL databases including spatial database. This enables web service middleware to establish connections with databases, send SQL statements and process the results. There are three modules in map service middleware; first is the user support module, which generate different interfaces for different users, including the visual query interface. The interface element can be described by XML configure file. Map datasets and user interface information are described in project.xml file for each web GIS application. It is possible to give customized interfaces to every user, even they are in the same user group. Second module is in charge of spatial data import/export function, it can exchange data between spatial database with several popular file formats: ESRI shapefiles, MapInfo MIF file and GML. The third module is for spatial analyses, it connects the spatial database and users, transmits users' queries with spatial SQL statement to spatial database and processing and returning the results back to the user side. Figure 8 shows components of the map service middleware in the developed prototype web GIS. At the time of this study, MySQL 4.1 is still a technical preview version without supporting advanced spatial operations, such as Intersection, Union, Difference, Symdifference, Buffer and Convexhull, which will be available in the future release according to mySQL document (MySQL, 2004).

For the client side, customized map applet is used. ALOV Map (Alov, 2004) is a free, portable Java application for publishing vector and raster maps to Internet and interactive viewing on web browsers. It supports the complex rendering architecture and unlimited navigation, allows for working with multiple layers and thematic mapping. It also supports hyperlinked features and attribute data. In this study, a module is developed to enable ALOV to process spatial data in WKT and GML format. Figure 9 shows the result of the user defined SQL query. With "On Map" function, the result can be shown in the map. Another example shown in Figure 10 is the thematic mapping function, where streets are separated in several different groups according to their speed limits.

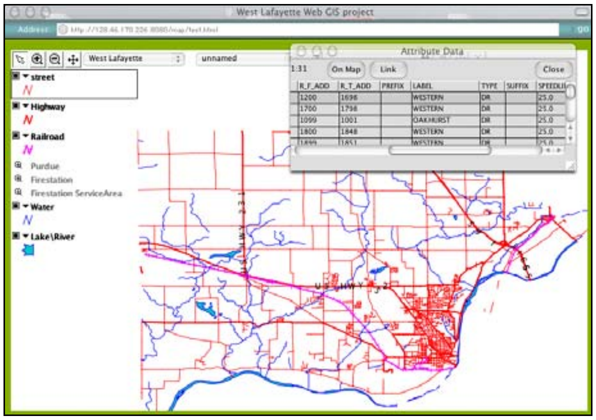

Figure 9. Example of client Java viewer

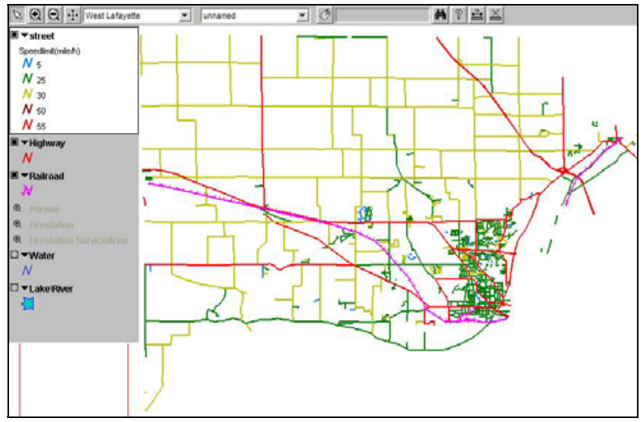

Figure 10. Example of thematic mapping function

CONCLUSIONS

GIS as an information technology needs to be enhanced with the capabilities of standard database and the Internet platform. Such enhancement and expansion should allow using spatial SQL to conduct spatial analysis in the web environment. Our study shows this can be accomplished, rather efficiently and effectively, by using and integrating open-source software. A prototype web-GIS has been developed to validate the concepts and experience the implementation. It is demonstrated that a proper user schema is able to provide different users with different access and manipulation privileges. Selection of spatial database can be a trade-off because of the considerations in its capabilities in spatial data handling and compatibility with industrial standards. Our study also shows that project-orientated development and customization of spatial enabled SQL is still needed to provide friendly and intuitive user interface. With such development, users can have access advanced spatial analysis tools that are only available in a desktop GIS. In our study, we successfully integrate all necessary functions in data browsing, manipulation, analysis, management and maintenance into one common interface of a web browser. This is shown to be very effective for both development and application.

REFERENCES

[1]. Alov (2004). Alov Map - free GIS tool. **http://www.alov.org**.

[2]. ApacheGroup (2003). Tomcat 4.x. **http://jakarta.apache.org/Tomcat/**.

[3]. Blaser, A. D. and M. J. Egenhofer (2000). A visual tool for querying geographic database. Advanced Visual Interfaces-AVI 2000, Palermo, Italy.

[4]. Coward, D. (2001). Java servlet specification version 2.3. **http://java.sun.com/products/servlet/**.

[5]. Egenhofer, M. J., J. Glasgow, et al. (1999). "Progress in Computational Methods for Representing Geographic Concepts." International Journal of Geographical Information Science 13(8): 775-796.

[6]. Kaushik, S. R. and E. A. Rundensteiner (2001). "SEE: A Spatial Exploration Environment Based on a Direct-Manipulation Paradigm." IEEE Transactions on Knowledge and Data Engineering 13(4): 654-670.

[7]. Lin, H. and B. Huang (2001). "SQL/SDA: A Query Language for Supporting Spatial Data Analysis and Its Web-Based Implementation." IEEE Transactions on Knowledge and Data Engineering 13(4): 671-682.

[8]. MySQL (2004). MySQL 4.1. **http://www.mysql.com**.

[9]. OGC (1999). OpenGIS Simple Features Specification for SQL: Revision 1.1, Open GIS Consortium (OGC), Inc.

[10]. OGC (2003). OpenGIS Geography Markup Language (GML) Implementation Specification: Version 3.0, Open GIS Consortium (OGC), Inc.

[11]. Peng, Z. and M. Tsou (2003). Internet GIS: distributed geographic information services for the internet and wireless networks. Hoboken, New Jersey, John Wiley & Sons Inc.: 1-35.

[12]. PostGIS (2004). PostGIS 0.80. **http://www.postgis.org**.

[13]. Shekhar, S. and S. Chawla (2003). Chapter 3: Spatial Query Language. Spatial Databases: A Tour. Upper Saddle River, New Jersey, Pearson Eduction Inc.: 52-82.

[14]. Shekhar, S., S. Chawla, et al. (1999). "Spatial databases: accomplishments and research needs." IEEE Transactions on Knowledge and Data Engineering 11(1): 45-55.