Motivation and Problems

The major characteristics of a Geographical Information System (GIS) from other information systems is the capabilities to handle and analyze spatial data supporting decision making based on geo-referenced data. The Internet has grown so rapidly today and then it has led to an increasing concern over the Web GIS. In this paper, the authors present effective interface to deal with the spatial data analysis functions based on the Web and a query language, SQL/SDA, for those kind of query processing. Even though many Web GIS developer have been tried to access, transmit, retrieve and map to the spatial data and employ the access to spatial analysis functions such as map overlay, buffer, and feature fusion, it’s still not effective and easy for World Wide Web (WWW) client/server environment. And several prior works in terms of query language to support spatial data analysis such as Geo-SAL, which is different from SQL and QUEL, user-defined extensions to SQL by using a macro expander, procedural language as well as cartographic modeling, and Spatially Extended SQLs(SESQL) are more focused on the representation and management of spatial data rather than stepwise analysis of spatial data. So the authors develop SQL/SDA which is well-adapted to the general spatial analysis procedures using current GIS packages. GIS has been used as a tool for spatial analysis but it’s not still used world widely because of the complexity of comprehensive GIS software packages.
The purpose of this paper is to propose a spatial query language, called SQL/SDA (Spatial Data Analysis), to support the expression of complicated spatial queries dealing with various spatial analysis problems, for example, finding a land parcel suitable for growing coffee in Jamaica, within a framework for spatial query language design and development which provided in SQL3/MM and Open GIS SQL. Also, authors report a critical factor of the SQL/SDA design is compliance with the general spatial analysis procedure using current GIS packages and compatibility with the SQL design concepts. It's very important because that relational database language SQL is still a popular database language and its functionalities are valuable. In other words, efficient involving already existing technology to the new development make users to enable to feel familiar with new SQL and interface so that user would be more likely to use them. In summary, this paper provide effective and easy web-based interface to support spatial data analysis with SQL/SDA, incorporating relational database and SQL with Geometry type, given a general spatial analysis procedure of current GIS packages.

**Major contributions**

The authors' main contribution is an attempt to develop web-based interface for spatial data analysis. Especially, easy and intuitive visual interface combined with textual language are necessary for first user and this interface reduce some errors. For example, their control window provide command button to check commands' grammatical correctness by producing users and all the spatial functions in SQL/SDA are provided in the icons window with proper pictures. However, it's not clear how each windows work with the main text window. And since this interface is not very much different from SQL visual interface such as MS Access, it's not original idea without dealing with spatial data query.

That SQL/SDA incorporates the important derivation functions as well as the spatial relationship and metric functions, revising FROM clause is another contribution in the query language area. The idea of embedding an SQL statement in the FROM clause is easy to understand and use, that is, this language is in a natural language-like style comparing with Macro Language of GIS, which is a kind of procedural language have lots of unstandardized
concepts and special-purpose functions so that doesn’t employ optimization strategy. Also SQL/SDA present the complicated multistep spatial analysis problems comparing with Spatially Extended SQLs (SESQL) which need complex and redundant query presentation to retrieve same result as SQL/SDA query since only the attributes in the FROM clause can be complied in both the SELECT and WHERE clauses. Although the authors introduce new type of FROM clause, it looks like VIEW, not a new concept. The concept of virtual table which several values from spatial functions are appended to and created during intermediate relation is similar to VIEW. The difference between them is the time when they are produced. It means the authors should have shown how different VIEW and their intermediate table from FROM clause are.

Finally, the web-based interface provided in this paper is very clear and can be an origin of visual interface for spatial data analysis. Hybrid model the authors choose also shows us a definite architecture between client and server. According the architecture, client can control over the query execution and server processes query request communicating with both client and SDE. This interface is also using in a real world so that it can be employed in wide applications in the future.

**Key Concepts**

1. **Spatial data type & analysis functions in SQL/SDA**

A basic geometric unit for spatial representation and processing in SQL/SDA is a feature-based model. This model includes both spatial attributes (e.g., coordinates and topological relationships) and non-spatial attributes (e.g., name, type, and size) and represents each feature in the table, which is similar to a table in a Relational Database Management System (RDBMS). In this paper, the authors choose SQL with geometry types to implement spatial feature table. It refers to a SQL environment that has been extended with a set of geometry types.
The authors define a set of geometry types:

According to geometry types, four groups of spatial functions are defined in SQL/SDA. First group is for property functions: CENTROID, AREA, LENGTH. Functions to test spatial relationships are next and metric functions which calculate distance and direction are third group. Creating a new set of spatial features is the last function, called derivation functions such as VORONOI, BUFFER, CONVEXHULL, INTERSECTION, DIFFERENCE, UNION, and FUSION. The authors try to put on the incorporation of above functions to SQL/SDA in a proper way.

2. Presentation of SQL/SDA

Basically, this paper uses a general spatial analysis procedure for the design of SQL/SDA. The procedure, using current GIS packages, start choosing source tables from given query and do spatial analysis operations as mentioned above. All the results are then combined into an intermediate relation and the selection operation with certain conditions can be conducted. The procedure finally projects desired attributes.
The structure of SQL/SDA is based on standard query language structure, i.e., “SELECT… FROM … WHERE,”. The main difference from an SQL statement is the FROM clause. By embedding an SQL statement in the FROM clause, the authors apply the spatial functions in the nested SELECT clause to produce an intermediate relation. In other words, the subquery in the FROM clause creates an intermediate relation, in the relation, the results from spatial operations can be acted on by further queries. This new structure makes SQL/SDA suited for presenting complicated multisteps spatial analysis problems and, since the subquery in SQL/SDA is in a template-like format, it can be easily composed with the help of a visual interface.

3. Example & its result

Two examples which are concerned with typical spatial analysis problems such as site selection and land suitability evaluation are demonstrated in this paper. First one is to select a lab site with following criteria: preferred landuse is brushland, soiltpe should be “A”, ite must be within 300 meters of existing sewer lines, beyond 20 meters of existing streams and site must contain an area at least 2,000 square meters. Query statement using SQL/SDA is
\textbf{SELECT} \textit{lab Location}

\textbf{FROM}

\begin{align*}
&(\text{SELECT} \ast, \text{INTERSECTION}(\text{lu.Location, sl.Location})\text{ AS ILocation,} \\
&\quad \text{BUFFER}(\text{sw.Location, 300}) \text{ AS buf1Location,} \\
&\quad \text{INTERSECTION}(\text{IsLocation, bufLocation}) \text{ AS IsbLocation,} \\
&\quad \text{BUFFER}(\text{sm.Location, 20}) \text{ AS buf2Location,} \\
&\quad \text{DIFFERENCE}(\text{IsbLocation, buf2Location}) \text{ AS labLocation,} \\
&\quad \text{AREA}(\text{labLocation}) \text{ AS areaval}
\end{align*}

\textbf{FROM} \textit{landuse AS lu, soil AS sl, sewer AS sw, stream AS sm}

\textbf{WHERE} \textit{lu.type='brushland' and sl.type='A' and areaval > 2000}

Only the difference from SQL is the Cartesian product of FROM clause. In the intermediate
relation of the FROM clause, intermediate values from spatial operations are appended as new
attributes. Everything before and after performing FROM clause is same as SQL.

Second example use the classification value in the FROM clause and shows a comprehensive use
of spatial functions, which integrates the approaches in the previous spatially extended SQLs.

\textbf{SELECT} \textit{FUSION(ILocation)}

\textbf{FROM}

\begin{align*}
&(\text{SELECT} \ast, \text{INTERSECTION}(\text{lu.Location, sl.Location}) \text{ AS ILocation,} \\
&\quad \text{AREA}(\text{ILocation}) \text{ AS areaval}
\end{align*}

\textit{classifyval} = (\text{CASE} \text{ lu.type || sl.type}
\begin{align*}
&\text{WHEN ‘BrushlandA’ THEN ‘|||’} \\
&\text{WHEN ‘BrushlandB’ THEN ‘||’} \\
&\text{WHEN ‘WaterA’ THEN ‘|’} \\
&\text{WHEN ‘WaterB’ THEN ‘||’} \\
&\text{WHEN ‘ForestA’ THEN ‘||’} \\
&\text{WHEN ‘ForestB’ THEN ‘||’}
\end{align*}
4. Implementation on the web (user interface & server)

The interface of SQL/SDA implemented based on hybrid models provides visual constructs for SQL/SDA query. The Java-based client communicates with the SQL/SDA server, which is coupled with the Web server. Client compose query statement given 5 types window: text, control, comboboxes, icon and settings window. Such visual query languages provide intuitive and ease interface to users. Server receiving client’s query commands communicates with spatial database via SDE then gives back the results to client and they are displayed.

Validation Methodology

It is tough to validate the work presented in this paper without being given any information about any user studies or acceptance tests being performed. Even though all the concepts such as new FROM clause and visual interface, introduced in this paper are indeed very attractive, no evidence has been provided that the ideas were implemented well or the resulting interface was accepted and appreciated while users were performing actual work. In terms of interface, user
studies are crucial since interface is for users as itself. This paper focuses on the side of being too technical while completing omitting the people aspect of interface design acceptance. Although these weakness, two examples provided in this paper are shown well how SQL/SDA work to analyze spatial data. The query process is very clear following a general spatial analysis procedure by using current GIS packages.

In the case study for comparing SQL/SDA with macro language and spatially extended SQLs, the authors report how effective and easy to understand and use SQL/SDA in terms of new query language based on web interface for spatial data analysis. They should have compared what kind of benefit their new language have comparing with these language because main topic in this paper is the query language using web-based interface as well as supporting spatial data analysis.

**Assumptions**

One of assumptions is spatial indexing is already provided by spatial data Engine (SDE). Spatial indexing is very important to query optimization but the authors don’t concern about that since spatial functions (API) are executed by the use of SDE. Indexing is connected to the efficient of database such as speed. If different type of SDE is selected for the SQL/SDA’ web-based prototype, working between SQL/SDA server side and SDE would be different. Another one is that users should be familiar with spatial data as well as SQL and relational database. Since almost all concepts are related with spatial data, users need to know the spatial data types and operations for spatial data analysis even though the interface is visual and easy to use.