



Journal of
**Software
Engineering**

ISSN 1819-4311



Academic
Journals Inc.

www.academicjournals.com

Distributed Geographic Structure Query Language-DGSQL

^{1,2}Liang Wu, ¹Zhanlong Chen, ^{1,2}Xincai Wu and ^{1,2}Zhong Xie

¹College of Information Engineering, China University of Geosciences, Wuhan, 430074, China

²National Engineering Research Center for GIS, Wuhan, 430074, China

Corresponding Author: Zhong Xie, College of Information Engineering, China University of Geosciences, Wuhan, 430074, China Tel/Fax: +86-133-8750-5800

ABSTRACT

Combined with Spatial data model of MAPGIS, this study brings up distribute structure query language (DGSQL) and has designed the interpreter of DGSQL. This study has given the method of the analysis of the distribute structure query language to the GSQL of the node of the spatial data and built the mapping model of the global spatial query sentence to the local query sentence. Hence, DGSQL can sustain the query of distribute spatial data. This study has realized the spatial relation operation (Intersection, Union, Difference) and synthesized operation (Length, Buffer, intersecting region) and so on.

Key words: DGSQL, distribute, structure query language, spatial data

INTRODUCTION

The Spatial database technology has become the key technology of ensuring the effective usage of spatial data in the distributed circumstance. In the process of the design and realization of the spatial database, spatial data query language's studying which is the main method of the interactive operation with spatial database has aroused the high attention of the academia (Mohan and Kashyap, 1993). The past years have witnessed a growing stream of research on spatial data query language (Wang *et al.*, 1997, 1999; Eisenberg *et al.*, 1999). For example, Fang *et al.* (1999) has been studied the geographic structural query language of the vector data. Li *et al.* (2002) has studied the geographic structural query language of the raster data. Wan *et al.* (2006) has brought up the GSQL combined the data model of the OGIS. The former studies has proved that the better approach of realizing the spatial query language is extending the structure Structured Query Language (SQL) to provide the ability of accessing spatial data in the spatial database in a uniform mode (Egenhofer *et al.*, 1994; Bauer and Lehner, 1997; Bittner *et al.*, 2000).

The realization of the share spatial data is more difficult in the distribute computation circumstance than in the normal client server state. The distribution of spatial data possesses the characteristic of unification in logical and decentralization in fact. In the distribute computation circumstance, the independence of spatial data, the controlling mechanism of centralization combined with self-management, data redundancy and distribute transaction management needs to considering. The distributing spatial database is made up of a set of spatial data which is distributed in the different node in the distribute network circumstance. The characteristic of distribute database as follows:

- In distributing characteristic, the spatial data stored in the distributing spatial database is not stored in one field (exactly speaking, the spatial data is not stored in one storage of computer). This is the difference to the centralized database
- In logical uniform characteristic, these spatial data has relationship with each other in logical. The spatial data distribution stored in the different nodes is one entity in logical (the same as centralized database). This is the difference of the distributing database and centralized database

MATERIALS AND METHODS

The parsing of spatial data analysis and query is made up of global parsing and local parsing. The distributed database is not the sporadic realization of centralized database. The global parsing of spatial data analysis and query is corresponding with the logical uniform and independence of the distributed database.

The global parsing model of DGSQL is shown in Fig. 1. The user's application sends spatial data query request to spatial database through the client of the global parsing engine using GSQL. According to the global spatial data catalog and data dictionary, the global parsing engine response the request use's request and analyze the DGSQL in the global interpreter using spatial resource searching algorithm. After the analysis of the query sentence, the analyzed sentence has to be optimized by the global optimizer. In this step, the global parsing engine should resolve the data redundancy and uniform of duplications.

The global query sentences need to be analyzed when the distribute query sentence is submitted to the global interpreter. After separating the global query sentence, every separated sub-query sentence correspond one local node spatial database in the distribute network circumstance. However, the sub-query sentence is not the local query sentence. It should be translated to the local query sentence if it is not coincident with the certain node local spatial query language. In the distribute network circumstance, after the global query sentences are send to the global interpreter, the spatial data should be transmitted from the node server to the center server. In most circumstances, the only part spatial data can meet the need of the query. If all the spatial data of the nodes is transmitted to the computer in which execute the query, the efficiency of the query will be reduced. The best method to resolve this problem is the query parsing which is the mechanism of determining the local tables corresponding to the global table by querying the global spatial resource catalog and data dictionary. The flow chart of executing DGSQL is present in Fig. 2.

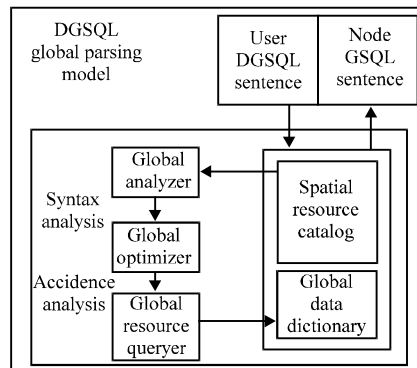


Fig. 1: Global parsing model of DGSQL

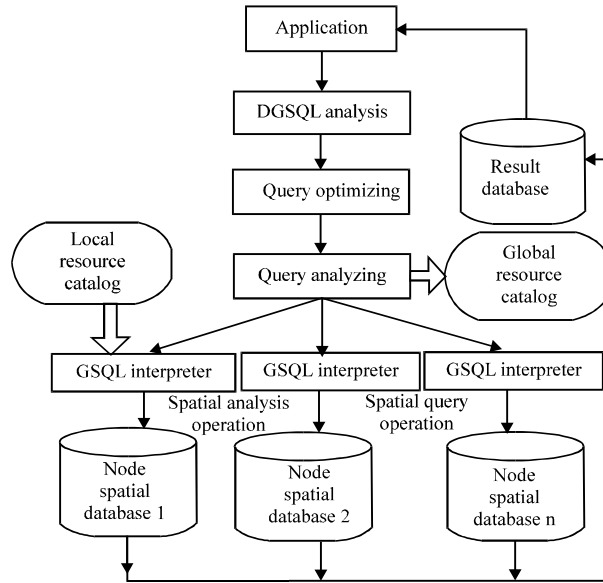


Fig. 2: Flow chart of DGSQL construction

Table 1: Global spatial resource catalog

Field name	Data type	Meemo
DBUID	Long-type	Node server guid
Servename	STR-type	Local server name
DBname	STR-type	Local database name
User	STR-type	User name of the database
PWSD	STR-type	User's password
Memo	STR-type	Memo information

```

CObjectCls *ptCObjectCls = NULL;//object class, ptCObjectCls = new CObjectCls;//new a object class, CATT-STRU stru;//attribute
structure of object class, CFIELD-HEAD fldhead, //field information, TYPE-XCLS-ID dmn-ID = 1;//dataset ID, lstrepy (fldhead.fieldname,-
T("DBUID")); //field name (DBUID), fldhead.fieldtype = LONG-TYPE; //field type, if(stru.AppendFld(1,&fldhead)>0) //append field,
if(ptCObjectCls->att-SetStru(stru)>0) //set attribute structure of object class, ptCObjectCls->Open(pGDB) ptCObjectCls->Close(), delete
ptCObjectCls; ptCObjectCls = NULL

```

Design of global spatial resource catalog: The module of global spatial resource catalog maintains the cache in which stores the spatial resource catalog and provides the fast searching algorithm of the distributed spatial data information. DGSQL is a storeroom ranking distributed spatial structure language. The aim of the spatial resource catalog is acquiring the local node servers in the distributed circumstance when interpret the GSQL sentence. After gaining the local node servers, the command sentence can be sent to these servers. The global spatial resource catalog is stored in the system's database because the change of global resource catalog is related to the addition and deletion of the node server in the network.

Global spatial resource catalog are described by the object class of MAPGIS7 as follows in Table 1 (take field DBUID for example).

Besides the global spatial resource catalog, the data type mapping table, the global data dictionary, the global keyword table and global translation dictionary should be designed.

Analyzer of DGSQL: The global query sentences need to be analyzed when the distribute query sentence is submitted to the global interpreter. After separating the global query sentence, every separated sub-query sentence correspond one local node spatial database in the distribute network circumstance. However, the sub-query sentence is not the local query sentence. The process of analyzing query sentence is the process of separation and combination of string. After the analyzing step, the query instruction is made up of single data source query instruction and multiple data source query instruction. There are no differences between the single data source query instruction and multiple data source query instruction in the construction of thread. The single data source query instruction need to start one thread using thread technology. After analyzing step, there are no differences in the form of single data source query instruction and multiple data source query instruction. The analyzer will inform the spatial data collector to store the result of single data resource query instruction in the temporary spatial database but not the result spatial database. According to the differences of SELECT sub-sentence, INSERT sub-sentence, UPDATE DELETE sub-sentence and spatial data analysis computation, the analyzer find the global table name in the instruction and searching the global data dictionary. And then the analyzer gains the corresponding serial number and the local data source and take place the global object to the local object and add the serial number at begin of the construction.

In the process of dealing with multiple spatial data source query instruction, the analyzer gains the information of feature class and object class stored in local database by analyzing the SELECT sub-sentence, FROM sub-sentence and WHERE sub-sentence of the global query sentence. The analyzer can get SELECT sub-sentence, FROM sub-sentence and WHERE sub-sentence of local query at last. However, the analysis of the WHERE sub-sentence should be especially analyzed, because it involves the complex joint of the distributed spatial database. The analyzing of WHERE sub-sentence needs to add the lists included by the query condition to SELECT sub-sentence of the local database query sentence. The name of feature class and object class queried in the global query should be added to the FROM sub-sentence of the local database query sentence. At the same time, it is necessary to add the list involved single database to the SELECT sub-sentence and the query condition to the FROM sub-sentence of the local query sentence. The flow of the realization of the global analyzer is shown in Fig. 3.

Interpreter of local GSQL: After being analyzed by the global analyzer, the local node spatial database server GSQL constitutes the DGSQL. The local GSQL sentences are send to the node spatial database to execute the local query. The local node spatial database server call the creating GSQL environment function before the local interpreter parsing the local GSQL. The local interpreter accepts the local GSQL sentences through the preparing interface. If the GSQL includes binding filed. For example, there is “disistance (river,:1)>100” in WHERE sub-sentence. “River” is the spatial information of a rive. “:1” denotes that this parameter should be bind. In this circumstance, it is necessary to appoint the serial no, type and address pointer of the parameter. The third step is to appoint the type of result set and bind the result set. The fourth step is to execute the query sentence. The local interpreter will take the result to the result set. The executing environment should be released by the corresponding interface. The inner realizing flow and the corresponding interface is shown in Fig. 4.

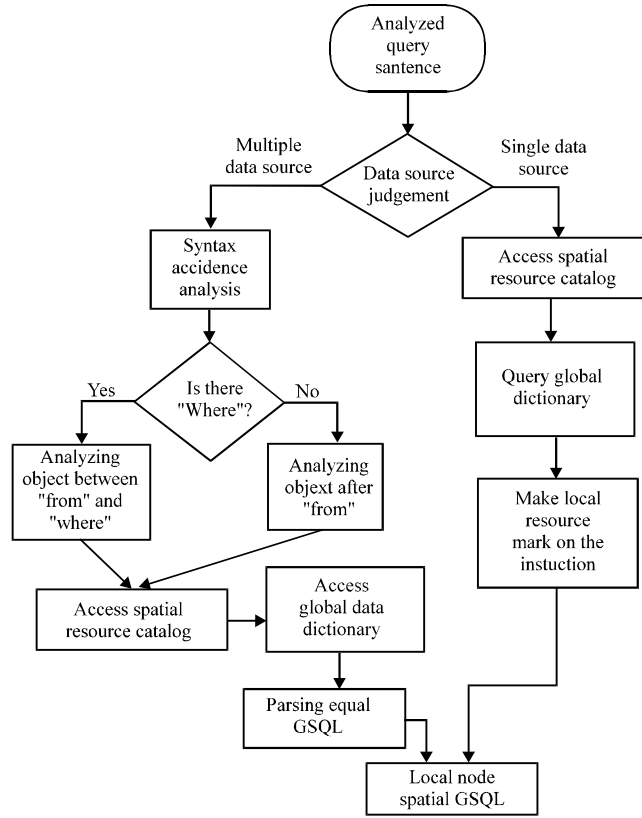


Fig. 3: Algorithm flow chart of querying analyzer

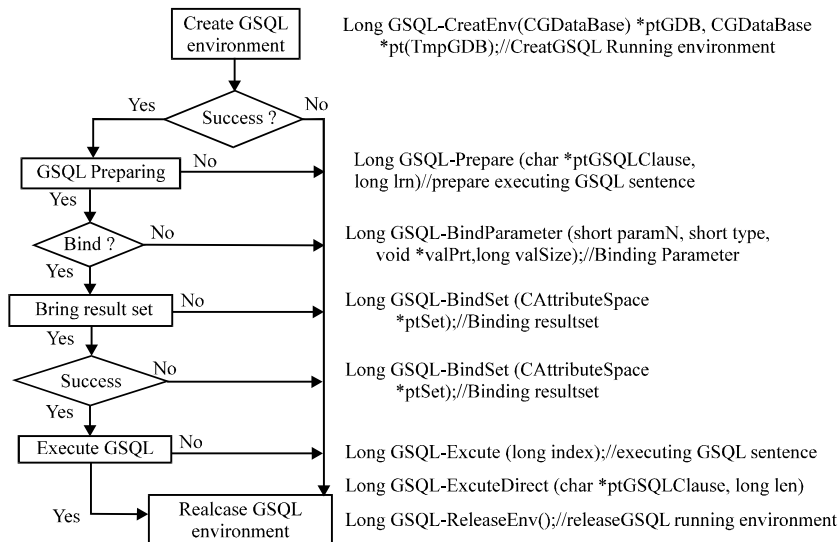


Fig. 4: Inner realizing flow and the corresponding interface

Spatial data collector: In the realization of the DGSQL, in order to get the result of the query, the spatial data collector must do two tasks. The first task is to get the result of each node

database querying. These results are the sub-set of the global querying result set. The second task is to combine the results of each node server in the network. Based on the combined results, the collector get the final result at last.

In order to get the final result, an operation should be taken on the sub-result of every local node in the network, so, it is important to store these sub-results. To get effective memory efficiency, the technology of temporary database existed in the life period of the query is adopted in the design of spatial data collector. The users do not create the temporary database and it is auto created by the data collector. The collector can auto create a temporary database if the users call the functions of operating feature class and object class using NULL parameter. The interface of temporary database can be gained by the following functions:

```
BAS70-EXPORT-CLASS CGDataBase *GetTempGDataBase();  
CLN-BAS70-EXPORT-CLASS long FreeTempGDataBase(CGDataBase *ptGDB)
```

After gaining the pointer of the temporary database, the querying result of local node are stored in the temporary database. The spatial data collector will operate the spatial data in the temporary database. For example, the spatial data collector can use A spatial data form node1 to intersect B data from node2 and at last get the result C. The spatial data collector will send the result to the final result database and make the temporary database empty. The user can get the final result spatial data from the result database.

RESULTS

To illustrate how the mechanism of DGSQL can be used in practical applications, a concrete example involving spatial queries in distributed circumstance is discussed here. The model base with the mechanism talked above is populated here.

There are resident spatial data (SC-RESPT) and road spatial data (SC-ROALK) in the different node in the network. A case it to query the residents at two sides of a certain road in the distance of 5 km.

It is necessary to construct query sentences bases the question. The steps as to answer this question as follows:

- Distill the certain road in SC-ROALK

```
SET at CLS-SC-NHW = SELECT ROALK FROM FCLS.SC-ROALK  
AS ROALK WHERE RN LIKE 'G%'
```

- Make road buffer within 5 kilometers

```
SET at CLS-BUFF = BUFFER(at CLS-SC-NHW, 5000)
```

- Within analysis, distill residents within the buffer

```
SET at CLS-RESPT-IN-BUFF = WITHIN(FCLS.SC-RESPT, at CLS-BUFF)
```

The spatial data collector stored the final result to the final result database when the steps above are completed. The result is presented in Fig. 5.

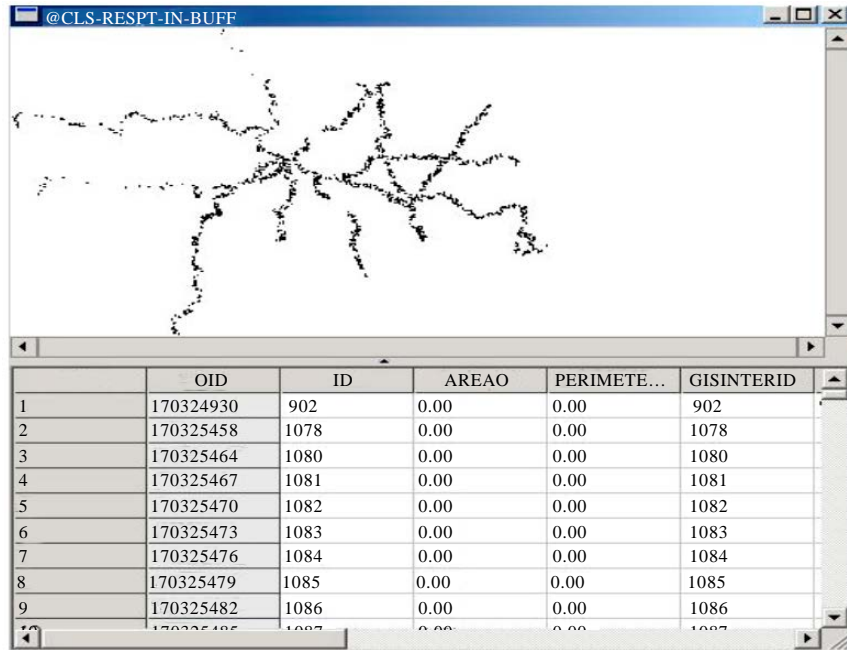


Fig. 5: Result using DGSQL

The novelty of this study is to bring the new concept of distributed uniform query. In the network, the user uses uniform query language to access spatial data in the distributed network and does not need to know where the spatial data is. The spatial data can be distributed in the different node in network.

DISCUSSION

At present, the realization method, spatial predication, spatial function extending and formalization definition of spatial query language have been studied maturely (Xie *et al.*, 2006; Benedikt and Libkin, 2000a, b). Most of the current GIS software platform has realized the interactive operation language based on their spatial data model.

However, with the development of the network technology, the magnanimous spatial data has to be stored in the different servers in the distribute network circumstance. The distribute deployment and computation becoming more and more important. Hence, the study of the spatial query language in the distribute network circumstance becomes the inevitable trend. In this study domain, Huang *et al.* (2006) has studied the GSQL global parsing under distributed computation circumstance which provides the reference for this study. The emphases of this study is the query and operation of the magnanimous spatial data based on the spatial data model (object class, feature class) in the distribute computation circumstance.

According to what talked about above, the core of the spatial data analysis and query is designing the corresponding distribute geographic query language and building the map mechanism from distribute geographic structured language to the local geographic structured query language in the distribute network system. The algorithm of the global parsing should reduce the transmission quantity. The local parsing has to accomplish the spatial data query and analysis in the node using GSQL. The key technology of local parsing is the syntax and accidence analysis based on the computer compiling technology.

Global parsing of DGSQL: Under the support of spatial resource catalog and global data dictionary, the global parsing is the method of making the global query disassembled equal local query. The global parsing should resolve the data redundancy and uniform of duplications. In the process of the global resource query, the analyzing algorithm has added restricted condition to the query sentence by retrieving the global spatial resource catalog and data dictionary to enhance the query efficiency. At the same time, the availability of the distributed spatial data resource should be checked. If the assigned spatial resource is invalid, the global resource interpreter should make response to this situation and chose other resource and build the GSQL of different node in the distribute network circumstance.

Local parsing of GSQL: The basic content of the local parsing includes: (1) Before the analyzing the query sentence, the query sentence should be separated. After being separated, the query sentence is made up of SELECT sub-sentence, FROM sub-sentence and WHERE sub-sentence and so on. (2) The construction tree of the local spatial database query sentence. In order to build a better syntax and effective GSQL sentence, it is necessary to make the syntax and accident analysis of the separated GSQL which back up the optimizing of the sentence in the next step. (3) Making the equal transformation to the optimized sentence. In this process, the transformation and computation of the data type and the count of scanning spatial data should be reduced. The realization of the local spatial data query is the process of the computation of syntax tree which it the result of the syntax analysis of the query sentence.

The computation of the syntax tree adopts the recursive down method. Every computation will make two operands and one operating symbol and the result of the computation is stored in the node of operating symbol. The result of the root of the syntax tree is a Boolean value which can be combined with the query item in the SELECT sub-sentence to produce the result of the query.

CONCLUSION

This study introduces a new concept of distributed geographic structured language and proposes a solution to queries associated with GSQL in the distributed network. We aim to expand the capabilities of current spatial information systems in accessing spatial data in the distributed network. This study has studied the key technology from global query to local query and realized the analyzer of DGSQL, the interpreter of Local GSQL and spatial data collector.

The emphasis of this study has been on using DGSQL to handle spatial information queries in the distributed network. The exact modalities of use of DGSQL for this purpose remain to be fully identified at this time. Because the complexity and hugeness of the spatial data, the efficiency of spatial data should be considered.

ACKNOWLEDGMENT

National Science and Technology Support Program for "The research of network service platform for geographic spatial information tool set" (2011BAH06B04) and Fundamental Research Funds for the Central Universities (CUGL090251).

REFERENCES

Bauer, A. and W. Lehner, 1997. The Cube-Query-Languages (CQL) for multidimensional statistical and scientific database systems. Proceedings of the 5th International Conference on Database Systems for Advanced Applications, April 1-4, 1997, Melbourne, Australia, pp: 263-272.

- Benedikt, M. and L. Libkin, 2000a. Relational queries over interpreted structures. *J. ACM*, 47: 644-680.
- Benedikt, M. and L. Libkin, 2000b. Safe constraint queries. *SIAM J. Comput.*, 29: 1652-1682.
- Bittner, T. and J.G. Stell, 2000. Approximate qualitative spatial reasoning. *Spatial Cogn. Comput.*, 2: 435-466.
- Egenhofer, M.J., 1994. Spatial SQL: A query and presentation language. *IEEE Trans. Knowledge Data Eng.*, 6: 86-95.
- Eisenberg, A. and J. Melton, 1999. SQL: 1999, formerly known as SQL3. *ACM SIGMOD Rec.*, 28: 131-138.
- Fang, Y., F. Chu and B. Chen, 1999. Spatial structural query language-G/SQL. *J. Image Graph.*, 4: 901-910.
- Huang, Z., X. Peng, K. Zhang, B. Chen, X.J. Ma and Y. Fang, 2006. An approach to geographic SQL global parsing under distributed computation circumstance. *Geogr. Geo-Inform. Sci.*, 21: 18-21.
- Li, Y., Z. Xie and Z. Huang, 2002. Design and implementation of a WAP GIS framework. *Proceedings of the International Conference on Web Information Systems Engineering Workshops*, December 11, 2002, Singapore, pp: 81-87.
- Mohan, L. and R. Kashyap, 1993. A visual query language for graphical interaction with schema-intensive databases. *IEEE Trans. Knowledge Data Eng.*, 5: 843-858.
- Wan, B., X. Pan, L. Yang and L. Li, 2006. Design and implementation of spatial query language GSQL. *Comput. Eng. Applic.*, 7: 140-142.
- Wang, W., J. Yang and R. Muntz, 1997. STING: A statistical information grid approach to spatial data mining. *Proceedings of the 23rd International Conference on Very Large Data Bases*, August 25-29, 1997, Athens, Greece, pp: 186-195.
- Wang, W., J. Yang and R. Muntz, 1999. STING+: An approach to active spatial data mining. *Proceedings of the 15th International Conference on Data Engineering*, March 23-26, 1999, Sydney, Australia, pp: 116-125.
- Xie, Z., X. Xu, K. Ban, M. Feng, R. Li and C. Yan, 2006. Component embedded GIS for network applications. *Earth Sci.: J. China Univ. Geosci.*, 31: 631-634.