Adopted native XML Database to Store and Index GML Spatial Data

Chuan HU
Department of Traffic engineering, Sichuan college of architectural technology, Deyang, China, 618000
E-mail: sujsin@yahoo.com.cn

Abstract: In this paper, the author first analyzed the model and structure of the native XML database (NXD) and studied the application schema and structure of the GML 3.x implementation. Based on these, the author proposed the schema cluster storage and the numerical index technology to store and index GML spatial data, and adopted JavaBean technology to extent the inner module of the eXist-db to prove and implement the theory that advised before.

Keywords: GML, GIS, storage, index, Native XML Database, interoperation

1 Introduction

As increasing of the geographic information and the software of GIS, information sharing and interoperability become the main problem for GIS application, analysis and development. In order to solve this problem, Open Geospatial Consortium (OGC) published the GML (Geography Mark-up Language) standard in 1999s. GML is a modeling language developed by Open Geospatial Consortium (OGC) as a medium of uniform geographic data storage and exchange among diverse applications. Now it is not only the global standard for the XML encoding of geographic information and is also the foundation for the Geo-Web. However, GML documents are usually large and complicated in structure. Existing techniques for XML document processing, either streaming-based or memory-based, may not deal with such GML documents efficiently. Besides, Increasing amount of geographical data is being presented in GML as its use widens, and raising the question of how to store GML data efficiently to facilitate its management and retrieval. There is an urgent demand to adapt existing XML techniques to support the processing of large XML/GML documents, as well as to express GML-native geospatial operations. The storage and index is the first step to use this data effectively.

To solve this question, many researchers study in the area, who are using many methods to storage and index GML data. Corcoles compared the storage and query performance in the GML spatial database based on three different relational databases and extended the SQL language for query [1–2]. However, it isn’t the best query language for GML because it conflicts with XQuery language. Vatsavai compared several XML query language and extended the XQuery language to support GML despite didn’t implement it [3]. Warnill Chung studied how to query moving objects [4]. Lakshmi N Sripada evaluated the merit use the spatial database to store GML data [5]. Yuzhen Lu analyzed the topology data scheme and used the different file to store the spatial and non-spatial GML data document into database [6].

According to the above discussion, which concluded that the main storage method was RDBMS, but that still have many problems such as lose much information when query GML data in it. For solving this problem, the author main study on how to store and index the Geography Markup Language (GML) spatial data used native XML database in this paper. GML was the extension application from XML coding in spatial. Therefore, GML adapt XML storage and index method was the best way. The paper supposed method could avoid the information loss when to insert or delete GML data, and save much time when to query in the native GML database.

2 THEORY of GML

2.1 GML INTRODUCTION

Geography Markup Language is an XML grammar written in XML Schema for description the application schemas, transport and storage of geographic information. It provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure and generalized value. GML is a large, rich, expressive language designed to have the ability to express any geographic concept in common usage. Projects start with an application schema, or profile of GML. As explained here, "Profiles live in the GML namespaces (http://www.opengis.org/gml) and define restricted subsets of GML. Application schemas are XML vocabularies defined using GML and which live in an application-defined target namespace. Application schemas can be built on specific GML profiles or use the full GML schema set." Basically, profiles and application schemas are smaller subsets of the GML schema designed by a specific information community and tailored to a small
number of uses. In this paper, it was to store this GML data which had the common application schema through using the same collective.

2.2 GML DATA MODEL

The key concepts used by GML to model the world are drawn from the OGC Abstract Specification. The basic concept is a Feature, i.e., an (object) abstraction of the real world phenomena, with spatial and non-spatial attributes. Figure 1 shows a city cut in four districts:

![Figure 1. City's districts](image)

Figure 2 shows the UML schema of the town example used in this paper, with respect to the OpenGIS abstract model. City and parcels inherits from Feature, and parcel has a Geometric property.

According to the UML and the GML application schema we can make the GML code. The section of the example tags as follows:

```xml
<city name='GZ' area='500' population='non'>
  <parcel id='1'><polygon>...</polygon></parcel>
  <parcel id='2'><polygon>...</polygon></parcel>
  <parcel id='3'><polygon>...</polygon></parcel>
  <parcel id='4'><polygon>...</polygon></parcel>
</town>
```

2.3 GML character

GML data have this character as following compared with traditional spatial data:

1. GML made by the OGC accorded the abstract geography model, many GIS software manufacturer and the third party software manufacturer supported the model. So that shouldn’t loss the some information when we translated this GML information.

2. GML used the document to represent the geo information. So it was very simple, frank, easy understand that we can use the generalize word process software and XML editor to read and edit this file.

3. Easy to control the correctness of the GML spatial datum. Because GML schema defined the content and structure for the GML document. Use this schema to validate GML data whether or not to agree with it when we edit or translate the data.

4. GML easy to connected with the non-spatial data.

5. We can translate this GML data into any one of the vector data format, for example, SVG or VML and show it on the any kinds of monitor needn’t install any graphics inserter.

6. GML based on XML, so any technology of the XML adapt to GML. For example, XML Schema, Link, Pointer, XSL and so on.

7. Geo information and attribute all enveloped in GML, this GML feature included a series of attribute, geometry information and topology information. There have some advanced method to describe this information. So use GML to construct this geo feature very easy.

8. Coordinate reference systems enveloped in GML, it was the base for data process in GIS.

9. GML to practice the interoperability very facility, we can operate any data if it on internet in spite of it store on different database or computer.

10. Not only use GML to represent vector data that base on feature model but also use it to represent grid data that base on field model.

2.4 GML SCHEMA

Feature is the base for the GML, any method and structure all accord it to implementation. GML model as we talked above. The general rule is GML uses an explicit syntax to instantiate a GML application schema conformant with the general feature model in an XML document. A feature is encoded as an XML element with the name of the feature type. Other identifiable objects are encoded as XML elements with the name of the object type. Each feature attribute and feature association role is a property of a feature. Feature properties are...
encoded in an XML element. The follow is parts of

\[
<\text{name}='\text{GZ}' \text{area}='500' \text{population}='\text{non}'>
\]

This is the section of a schema.

Base schema components just for some special goals, examples a root XML type from which XML types for all
GML objects should be derived, patterns for collections and arrays, and components for generic collections and
arrays, components for associating metadata with GML
objects, components for constructing definitions and
dictionaries. At same time defined some base object and
GML properties and so on.

3 METHOD OF GML STORAGE AND INDEX

3.1 GML Management

3.1.1 Relational database

At present, many researchers proposed to use existing
relational database or orient-object database to manage
the XML data [12–13]. This kind of database namely for
enables the XML database which usually said (XML
Enabled Database, XED), the mainstream relations
database or the orient-object database suppliers all are
select this method to provide to XML the support, their
database interior still used the original relational table to
carry on the memory, this kind of database product had
Oracle9i, SQL Server2000, DB2 and so on. Its main
thought is to break up the structure of XML tree and
reorganization this data transforms it into form of the
relation database to store. When we want to use this data
we just use the SQL language to draw-out it from the
form which stored in the database and translate this data
into XML data structure. Its main merit is to use an
existing database management system technology. Its
main shortcoming is as followings: firstly, because the
XML data is the half structures data, but the relational
database management system is the structure data,
translated the data between the two model must lose
some information. Moreover, in order to adapt the XML
documents expression way must increase massive
redundant data that destroyed the table structure, which
also has wasted the storage space. Secondly, because of
the XML documents have the structure changeable
characteristic, when the XML structure changed it must
arouse the relate structure of the relational database
changed, this should reduce the existing database system
performance that not agree with the characteristic of the
structure data manage method. In addition, when to add
the row or delete this data, it must arouse the structure
change then the data form should to change for consistent
with it. Like the file system, when use relationship
database system to processing the large number of spatial
data that will reduce the efficiency.

3.1.2 NATIVE GML DATABASE

There is much more problem to take this system to
manage the GML data, so we should find more effi-
ciency method to achieve it. Because GML based on the
XML, so we can use NXD to manage this data. Native
XML database to manage GML spatial data is the best
way in all the methods. It represented the orientation
of development for the GML database because it used the
natural method to store XML file to achieve the storage
and manage for the semi-structure data of the XML. The
database adopted the standard format of XML in interior
needn’t translated this data file so that can show the merit.
Similarly, we can establish index for this document in the
NXD, store the data and index in the same database to
support the query that through speed the search velocity
find out this special information document in database.
The system has Xquery, XML storage and kinds of
operation aim at the XML data to design, so we needn’t
translate this information that shouldn’t lose the
information and lowest the performance. The native xml
database becomes more and more important in the
information technology. At present, there is a series of
excellent product, for examples, eXist, Xindice, OrientX,
Tamino, X-Hive/DB and so on. They are not only support
the lasted standard of the XML to implement the Xquery
but also provided the traditional database function
includes transaction, lock and so on. There have three
type of native xml database according the store
granularity: based on element, based on sub-tree, based
on document. Many of the native databases not
considered the schema or DTD. EXist-db considered the
schema and supported the Xquery language. So in the
paper we adopt eXist-db for us research object.

3.1.3 Cluster store by the schema

Now, we know using native xml database system to
manage GML spatial data is the best way, however,
different NXD have the different model and data
structure so we must select the best database to my
research object. In the paper, we select eXist-db for the
object. That’s why we select it because that has some
perfect performances examples support Xquery and
provide the collective for storage.

EXist-db is an open source database management system
together built on XML technology. It stores XML data
according to the XML data model and features efficient,
index-based Xquery processing. EXist-db provide the
client for customer so that we can edit XML code direct
in the collective. The top collective is the object that we
abstract it from natural. They are defined two function
which is document() and collection() because the
database include much more document collective so it
query engine need those function to decide which file
document is our current operate object when we input
some appointed file into the specific collective. The
function document() may accept an independent
document name, a list of the document name or asterisk
wildcard as the input parameter. When use the asterisk wildcard (*) that told us all document must be selected. The function collection () confirmed which document will be the operate object when we use Xquery to query. For example as follow:

```
collection('/db/city')//SCENE[SCENE[name = 'gz']]/TITLE
```

The root collective must always is /db in database. So we can use it to collect GML schema into the same collective. The figure 3 represents the structure of the store model by the schema collection.

![Feature schema](Feature.png)
![Geometric primitives schema](Geometric.png)
![Topology schema](Topology.png)
![Coverage schema](Coverage.png)

**Figure 3. Schema collection**

The figure3 just describe parts of the structure, supposed the root collective was the city that has many parcel. Any element of the city has one schema for example all have area so each parcel hold itself Coverage schema so we use one collective to store all the parcel's coverage schema. That means all object have the schema collect this schema in the same collective. Thus, it can very convenient to manage these GML data through the schema collection. We named the method was cluster store by the schema which collected this kinds of file that have the same schema in the same collection. The other reason to use this way to manage GML data is that they are convenient to establish index.

### 3.2 GML Index

#### 3.2.1 Index summarize

GML index includes XML index and spatial index. The XML inquiry and the traditional database inquiry compares, besides looks up based on the value to inquire, what are more is inquires based on the XML structure. Therefore, to satisfy the XML inquiry, besides conventional value index, it is necessary to design XML index based on the element index and path index. Since around SVN revision 6000, spring 2007, i.e. after the 1.1 release, eXist provides a new mechanism to index XML data. This mechanism is modular and should ease index development as well as the development of related (possibly not so) custom functions. As a proof of concept, eXist currently ships with two index types. The first is NGram index, An NGram index will store the N-grams contained in the data's characters, i.e. if the index is configured to index 4-grams, `<data>abcde</data>` will generate these index entries: abcd; bcde; cde#; de##; e###.

A spatial index will store some of the geometric characteristics of Geography Markup Language geometries. Follow was one section of GML code.

```xml
<gml:Polygon xmlns:gml = 'http://www.opengis.net/gml'
srsName='HC'>
  <gml:outerBoundaryIs>
    <gml:LinearRing>
      <gml:coordinates>
        232515.400, 111060.450 232515.150, 111057.950 232516.350, 111057.150
        232546.700, 111054.000 232580.550, 111054.900 232609.500, 111048.100
        232609.750, 111051.250 232574.750, 111054.650 232544.950, 111057.450
        232515.400, 111060.450
      </gml:coordinates>
    </gml:LinearRing>
  </gml:outerBoundaryIs>
</gml:Polygon>
```

It will generate index entries among which most important are:
- the spatial referencing system
- the polygon itself, stored in a binary form
- the coordinates of its bounding box

The numeric index of spatial index will we discussed in details further below.

#### 3.2.2 Numerical Index

At present, there are several services for the XML document index. However, No one of these methods applied to spatial data index. In the section main talked about the numeric index according the cluster store by the schema. As discussed above, we stored this spatial data into the different collective according different schema. If the object has the same schema then we would store this schema in the same collective. So we accord the schema to build the node tree, different tree node represented different schema. So we can give a number to the node then store this number in together. When to query some spatial information we just through this number to query the schema collective for the object. That would improve the efficiency of GQuery. Figure 4 is the schema implement in NXD that just represented the parts of data. So we design the tree described by figure 5.

The schema cluster storage adopts layer to organize this data, and we can find that on the top level is the feature object. In the figure 6 the object is “city”. The next level is this schema. They are just told us which category store in the collective. So we decide according level to coding this node. In the figure empty frame told us there haven’t any instance but we must code it that can be a relatively simple algorithm to achieve the automated coding.

Now we can start using JAVA to program in the eXist-db database system to expand its functions realize the
function of spatial index. There we defined a series of in the program. The main function include IndexManager(), IndexController(), newInstance(), etc to manager, control and build that index. This completed the GML spatial data storage management and index based on that store method, such as the establishment of a series of work.

With the continuous increase of GML spatial data, GML data management will become an important task. This paper is based on importance of GML data storage management how to store and index GML documents use native database system. The paper suggestion a construction method that is schema cluster storage based on native XML database system as well as the numerical index technology base on that store model. However, there we are only a small part of the work; due to time constraints we have a lot of work has not been done. For example, attribute query, spatial query and query those at same time, etc. we should compare that method with others, for example NXD compared with RDBMS or orient-object database, as well as when there have a large number of GML data this way’s efficiency how to change. Therefore, future work is also very difficult.

4 CONCLUSION AND FUTURE WORK

With the continuous increase of GML spatial data, GML data management will become an important task. This paper is based on importance of GML data storage management how to store and index GML documents use native database system. The paper suggestion a construction method that is schema cluster storage based on native XML database system as well as the numerical index technology base on that store model. However, there we are only a small part of the work; due to time constraints we have a lot of work has not been done. For example, attribute query, spatial query and query those at same time, etc. we should compare that method with others, for example NXD compared with RDBMS or orient-object database, as well as when there have a large number of GML data this way’s efficiency how to change. Therefore, future work is also very difficult.

References

[10] Chia-Hsin Huang,etc; Efficient GML-native Processors for Web-based GIS: Techniques and Tools.
[14] www.exist-db.org

Figure 4. schema storage implement in eXist-db

Figure 5. numerical coding technique based on schema storage