

Mapping the Future: Development of Transformation Rules for GML-Based Database Management

S Venaktesh

Koneru Lakshmaiah Educational Foundation, KLEF, Vaddeswaram, Guntur- 522302,
Andhra Pradesh, India

Abstract

Now-a-days the extensible Markup Language (XML) is also become essential in geographic information systems (GIS). To assist this demand, the OpenGis Consortium proposed the GML (Geography Markup Language), a markup language for specify geographical information, including space properties. One may understand GML as a collection of various geometric properties, objects to describe coordinate reference systems, a topology, specific time, measurement units and generic values. This scientific language agrees to describe the data by using of basic models which are published as per the OpenGis specifications. It applies the XML pattern and possesses the methods and principles to describe geographical information. This paper reveals a novel approach to store GML documents in object-relational database management systems. It also makes an analysis of the renovations of a repository and GML metadata for object-relational database management system.

Keywords: Geographic Information Systems, GML metadata, OpenGis, object-relational database management system

1. Introduction

With the advance of XML standard (extensible Markup Language) in exchanging information via the Web, mechanisms for the transportation and storage of XML documents (extensible Markup Language) became essential, especially in specific areas such as geographic information systems (GIS). To assist this demand, the OpenGis Consortium proposed the GML (Geography Markup Language), a markup language for specify

geographical information, including space properties. GML defines a group of geometric properties, besides providing types of object to describe coordinate reference systems,

Research paper

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

topology, time, units of measure and generic values. The GML language allows one to describe the data using basic models published according to the OpenGis specification.

The GML version 3.0.1 [15] is based on common concepts of the Geographic Information Systems (GIS), including the basic models for the description of geographical data. Application schemas can be built by the automatic translation of XML Schema and defined from conceptual models in another conceptual language, like UML [15]. In both cases, meeting the requirements of the GML specification demands that the application uses the basic constructions in a direct or indirect way, for each specialization. GML is therefore a standardized and unified way to make the exchange of geographic information easy by using the standard XML.

In an exchange of data between systems, the schemas should be shared, so that an application could recognize the type of data to be treated. A group of metadata is important in this context [15]. GML supplies a meta schema or classes and it is based on the OpenGIS data model called OGC Feature Model [15]. The types of data derived from basic GML types favors and guarantees the conformity with this model.

The OpenGIS specification may guarantee that a schema and its data will be useful for software based on GML. These schemas, translated to databases, provide the advantages of storing delegated systems and geographical data of other institutions (GML format) in a single system.

Once stored in a DBMS, users can manipulate the data using a more natural language and closer to the existent applications, the language of the traditional databases systems (SQL). The use of the database infrastructure is also important in scenarios demanding integration of conventional with geographical data. Another advantage is the possibility of using the functionality typical of a traditional database system, such as scalability, query language, and others.

This work describes a set of rules to map GML schemas for an object-relational data model (OR). To automate the mapping process, a tool using script language to define the database schema and the storage of GML instances into DBMS (Oracle 9i) was implemented. Therefore, a characteristic mediated by the tool is the creation of the object-relational schema starting from GML schemas.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

This article is organized as follows. Section 2 describes the main GML characteristics. Section 3 presents a review of the techniques of storing XML documents into databases. Section 4 presents a strategy for the translation of GML metadata for database schemas that takes in consideration to the main constructions used in practice. Section 5 presents the prototype developed for the generation of the definition script for the data types and the results of GML repository schema transformation. Finally, Section 6 presents some conclusions and future works.

2. GML Standard

GML is a XML Schema representation for modeling, transport, storing and sharing geographic information, including space and no-space properties of geographic features [15].

The GML specification defines the syntax, mechanisms and conventions that:

- supply an open framework for the definition of application schemas;
- allow profiles definition that support GML subsets;
- support the design and transport of application schemas for specific domains;
- Increase capacity of organizations to share geographic applications schemas.

Real world phenomenon corresponds, in GML, to the concept of geographical feature abstraction of a geographic phenomenon. A digital representation of the real world can be thought as a group of features. The state of a feature is defined by a group of properties, where each property can be thought as a triple one <name, type, value>. The number of properties that a feature can have, together with their names and types, is determined by the definition of its type. GML also defines several entities such as temporal reference systems, topologies and geometries through a hierarchy of GML objects.

3. GML Storage

Research paper

Bourret classifies the several administration systems [1, 21]. The most common storage system usually involves the XML mapping for regular data types in DBMS, because it is advisable that bases of XML data could benefit from the functionalities available in database systems [9, 12, 13, 20].

There are a number of approaches for storing XML documents [5, 6, 7, 17, 18] in DBMS, in order to store them in a relational or hybrid database. Some DBMS provide the support for

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

storing XML documents, but the specifications are fixed, limited, and built-in [8, 14, 16].

More recent versions supply the support for the native or hybrid storage.

The storage of GML documents is obtaining a growing attention of the research community [3, 10] however, the problem was not sufficiently studied. Corcoles and González present studies in the storage of GML documents in DBMS [3] and an approach presented in [10] proposes the mappings of GML documents for spatial databases (using a basic collection of spatial data types). In our case, the problem was studied through the generation of the script for the data definition in object-relational DBMS (collections, data types, etc.) and further on the storage of the GML documents.

The GML storage in DBMS corresponds to the problem of the database schema definition. This problem can be studied using a fixed schema or for the understanding of the structure of the document. It is also observed in the GML specification that there is a set of declarations of types of elements and a large set of structures of data types. Therefore, the mapping involves translating these structures into an object-relational data model, dealing with dependence relationships and the hierarchical definition of data types and collections.

4. Rules to Transform GML into Object-Relational Models

For each type of element of the repository of GML schemas [15, 11] a transformation rule is applied. A prototype that allows the automatic generation of the script for data types definition and the results obtained by the application of this rules are presented in the section 5.

Rule T1 - Elements and Global Complex Types: The elements and complex types defined in the root of the document correspond to types in the OR model. When this element type is declared in the root of the document, it becomes a candidate to be reused by other types of elements from the schema. This means that the objects can be referenced by other objects. In this way, the complex type (Figure 1) is mapped for the composed type Departed Feature Type, allowing it to be referenced by other objects.

Rule T2 – Simple Types: A simple type (simpleType) corresponds to existent and derived type. An attribute or sub-element (of the simpleType) it corresponds to the type defined for the simple type assuming the definitions of this type.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

```
<complexType name="DepartedFeatureType">
  <complexContent>
    <extension base="gml:AbstractFeatureType"/> </complexContent>
  </complexType>

Script: CREATE TYPE DEPARTEDFEATURETYPE AS OBJECT() UNDER
ABSTRACTFEATURETYPE;
```

Figure. 1: Departed Feature Type declaration and generated script.

Rule T3 – Anonymous Types: A complex type (complexType) declared as child of an element declared in the root of the document and anonym (without the declaration of the type of the element and without the name of the type of the complex or anonymous). The element corresponds to the composed type and the sub-elements sequence, attributes and references of the complex type it corresponds to attributes of this type.

Rule T4 – Elements of Complex Types and Extensions: An element declared as being of the complex type that was defined in the root of the document. As the complex type was translated for a composed type (Rule T1), the element is declared as being of the complex type, then the element should be translated for a subtype of the composed type (Figure 2). A complex type that extends another type through extension clause corresponds to a composed type and it also inherits the attributes of the composed type.

Rule T5 – Element Group and Attribute Group: The element group (group) and attribute group (attribute Group) declared in the document root correspond to composed types. The attribute group

(Attribute Group) is used by sub-elements of the schema because they define a collection of attribute declarations in only one hierarchical position (global) and they are used as reference by other sub-elements allowing reuse and increasing the schema modularity. In that way, the declaration of the group of attributes can be mapped for a class and the declarations of the attributes for properties of that class. They are commonly used in the definition of the attributes of a type.

Rule T6 – Hierarchy of Elements Types: A sub-element is declared as being of the complex type previously defined in the document root. The sub-element is derived from the complex type and it corresponds to a composed type subtype obtained by the complex type

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

transformation. The derived sub-element of the complex type inherits the composed type properties.

Rule T7 – Element Collection: A complex type and a sub-element including the maxOccurs attribute indicating that the sub-element can occur n times. The complex type corresponds to a composed type and the relationship between the complex type and the sub-element corresponds to an attribute of the type sub-element collection.

Rule T8 – Data Types Collection: A sub-element is defined by a complex type and with the maxOccurs attribute indicating that the sub-element can occur n times (Figure 3). The sub-element corresponds to a type collection defined by the complex type.

Rule T9 – Reference: The REF declaration is usually applied to allow an element previously defined to be reused. In general, REF refers to a value of a global declaration. In other words, this declaration refers to a global element and it allows the referred element to appear in the hierarchical level in which it is declared. The object-relational model (corresponding SQL-99) possesses a type of denominated REF. This constructor defines a type that is a reference for an object. The REF type can also be used to define attributes of a composed type. Some authors allow references to the types to be implemented as well by using one or more

attributes of a composed type [4]. Therefore, the REF declaration can be translated into an attribute of the composed type (ex.: Figure 6).

Research paper

```

<complexType name="CartographicSymbolType"> <complexContent>
  <extension base="osgb:AbstractFeatureType">
    .....
  </extension>
</complexContent>
</complexType>
<element name="CartographicSymbol" type="osgb:CartographicSymbolType"
substitutionGroup="osgb:_CartographicFeature"/>
Script: CREATE REPLACE TYPE CARTOGRAPHICSYMBOL AS OBJECT() UNDER
CARTOGRAPHICSYMBOLTYPE;

```

Figure. 2: CartographicSymbolType declaration and generated script.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

```

<complexType name="AbstractRoadInformationType">
  <complexContent>
  <extension base="osgb:AbstractFeatureType">
  <sequence>
  <element name="changeHistory" type="osgb:changeHistoryType" maxOccurs="unbounded"/>...
  </complexContent>
  Script: CREATE TYPE CHANGEHISTORYLIST AS TABLE of CHANGEHISTORYTYPE;

```

Figure. 3: AbstractRoadInformationType declaration and generated script for changeHistoryType sub-element.

Rule T10 – Reference Collection: The REF declaration possesses the max Occurs attribute meaning that the relationship between the complex type and/or element and the REF declaration is "one-to-many". The max Occurs attribute and the REF declaration may correspond to an attribute of the type referenced element collection (Figure 4).

```

<complex Type name="AbstractGMLType" abstract="true"><sequence>
<element ref="gml: metadata Property" minOccurs="0" maxOccurs="unbounded"/>...
</complex Type>
Script: CREATE OR REPLACE TYPE metaDataTable as table of metaDataTable;
CREATE OR REPLACE TYPE AbstractGMLType AS OBJECT (metaDataTable metaDataTable
null,);

```

Figure. 4: AbstractGMLType declaration and generated script for metadata Property sub-element.

The following section describes some results obtained in the transformation of a GML document group, in order to validate the set of rules presented.

5. Implementation, Tests and Results

The geographical applications make use of a subset of types described in GML specification. Basically, GML defines a set of types to codify GML objects with identity, properties and attributes. A schema of a GML application is, therefore, written according with rules of a vocabulary to define geographical objects for a private domain. So, the transformation from GML to object-relational model (SQL-99) requires the source must be written correctly, in other words, it have to satisfy the GML specification.

The next subsection describes the architecture of the implemented prototype and a critical analysis of results obtained by the application of the rules presented in the section 4.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

5.1 Prototype Architecture

A prototype was implemented to test the rules. It reads GML schemas and generates scripts for the DBMS Oracle 9i. The architecture of the prototype (Figure 5) contains three modules: Identification, Cl_Java and DDL_OR.

The Identification module uses an external API called Xerces [22] for syntactic recognition process, validation and identification of the elements of the schema using the DOM. The Cl_Java module controls the transformation of GML to Java classes. It was adapted from Castor framework APIs [2] which transforms GML schemas to Java objects and implements JAXB specification [19]; however, it has some important restrictions. To control the Java

classes generation it was used some templates to identify the types of elements and to create a GML schema.

Research paper

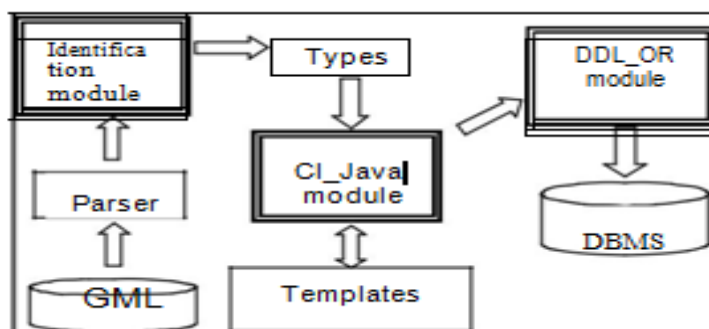


Figure. 5: Prototype Architecture.

Finally, the DDL_OR module produces a script to specify the data types that will be defined in the DBMS (in this case, DBMS Oracle 9i). This module produces a script instead of creating the data schema in the DBMS to allow the designer adapting the schema definitions as well as to add other properties that were not treated by the prototype. This module will be extended to add a component to storage in the database the data extracted from the GML documents.

To illustrate the transforming process of GML to object-relational model the Figure 6 shows a GML schema and Figure 7 shows the corresponding script generated by the prototype.

In Figure 6, the complex type SchoolType corresponds to a feature that has the geometric properties: location and address. The SchoolType also has an ID, name and description inherited from AbstractFeatureType.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

Using the script generated by the prototype (Figure 7), the designer can create the data types and define the table school in the database with the following command: CREATE TABLE "SCHOOL" OF "SchoolType".

```

<complexType name="SchoolType"> <complexContent>
  <extension base="gml:AbstractFeatureType">
    <sequence>
      <element name="address" type="string"/>
    </sequence>
  </extension>
</complexType>
  
```

```

<element ref="gml:location"/>
</sequence>
</extension>    </complexContent>
</complexType>

```

Figure. 6: School Type declaration.

```

Script: CREATE OR REPLACE TYPE "POINT" AS OBJECT ("srsName" varchar2(255)
null,"coordinates" varchar2(255) null);
CREATE OR REPLACE TYPE "Location" AS OBJECT ("Point" "Point" null);
CREATE TYPE "SchoolType" AS OBJECT ("name" varchar2(255) null, "address" varchar2(255)
null,"Location" "Location" null) UNDER ABSTRACTFEATURETYPE;

```

Figure.7: SchoolType script.

The prototype can be developed with the generation of the script definition of the tables from the generated data types. After defining the schema in the database, it can import the GML document to the database (Figure 8).

Since geographical application schemas are elaborated based on GML metadata, so before processing an geographical application schema it is necessary to transform the schemas used by this application. For instance, in the Figure 6, the element <AbstractFeatureType> must be transformed before the complex element SchollType.

To define the domain of values of an element, some procedures must be executed. For instance, the element <element name = Object " abstract = true " /> doesn't have a defined type and so, it will be defined further and his type is associated to a tag "anyType".

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

Research paper

```

C:\WINNT\System32\cmd.exe - sqlplus "scott/tiger@trabalho"
SQL*Plus: Release 9.2.0.1.0 - Production on Tue Feb 18 19:13:3
Copyright (c) 1982, 2002, Oracle Corporation. All rights rese
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production
SQL> select * from scott.school;
name                                address
-----
Location(Point('srsName', coordinates))
-----
Beta                                1673 Balsan St.
Location(Point('http://www.opengis.net', '40.0,5.0'))
SQL>

```

Figure. 8: GML data stored in the database.

5.2. Results and Tests

Most of the applications should use a subset of the schemas described in the GML specification.

This way, the metadata GML definition [15] was used too for the tests.

A group of tests accomplished for the validation and verification of the completeness of the proposed rules also included the use of the repository of GML documents of Ordnance Survey that describes topographical information [11, 15].

A digital map of Great-Britain created by the Ordnance Survey Committee includes topographical information of features for use in GIS and databases systems [11]. This repository includes the schemas (11 files) that describe the structure and the data of topographical information, address, images and highways of Great-Britain. For each repository schema was generated a database schema using the prototype. In Table 1 is presented the result of Ordnance Survey repository transformation.

It is observed that the number of times that each rule was applied and the percentual of each rule and its transformation. For instance, the GML definitions were used before because a GML schema uses its constructions. So the necessary metadata were translated to object types without restrictions (this elements were declared in the GML schema used).

Table 1: Number of times in that each rule (T1 T10) was applied with base in the amount of element types found.

Num. Occurrences in the gml schema	Rule - Transfor mation (Tn)	Ocurr. Struct (%)
103	T1	26,96
38	T2	9,95
7	T3	1,83
74	T4	19,37
11	T5	2,88
47	T6	12,30
10	T7	2,62
41	T8	10,73
46	T9	12,04
5	T10	1,31

It was observed that all the necessary metadata and the GML schema were translated without restrictions to object-relational types. After defining the schemas in the database, GML document were imported to the database. Others GML data may be imported to the DBMS.

6. Conclusions

1
2
7
1

This work described a set of transformation rules that can be used to store GML documents in object-relational database systems. It presented the architecture of a prototype developed to verify the applicability of the rules and it is showed the results of the transformations using

Research paper

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

an arbitrary repository of GML documents downloaded from the Web. In this case, the results show that all the GML element type were transformed in an object-relational structure and so, all the GML documents were transformed and storage in the database. Thus, these tests suggest that the set of rules is sufficient to transform any GML document, involves regular object-relational data types and the data could benefit from the functionalities available in database systems (like SQL-99).

As future work it can be mentioned: identification of GML elements frequently used in objects definition; extension of the prototype allowing inserting a group of instances of GML documents in the database and manage temporal GML elements into ORDBMS.

References

- [1] Bourret, R.. XML-DBMS. Available in: (<http://www.rpbouret.com/xmldbms>). Last visit on: Mar. 2003.
- [2] Castor, Castor Open source data binding framework. Available in: (<http://castor.exolab.org>). Last visit on: Oct 03.
- [3] Córcoles, J. E., González, P. Analysis of different approaches for storing GML documents. IN: ACM International Symposium on Advances in Geographic Information Systems - ACM-GIS 2002, McLean, USA, 2002. Proceedings...McLean: ACM Press, 2002.
- [4] Elmasri, R. and Navathe, S. B. Fundamentals of Database Systems. 3. ed. Reading: Addison-Wesley, 2000.
- [5] Florescu, D. et al. Answering XML Queries over Heterogeneous Data Sources. In: CONFERENCE ON VERY LARGE DATABASES – VLDB, 27, 2001, Rome, Italy. Proceedings... Rome: Morgan Kaufmann, 2001.

[6] Florescu, D., Kossman, D. A performance evaluation of alternative of alternative mapping schemes of storing XML data in a relational database. France: INRIA, 1999. *Research paper* (Technical Report 3680).

[7] Han, W.S. An XML Storage System for Object-Oriented / Object – Relational DBMSs. Journal of Object Technology, v.2, n.1, 2003.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Issue 2, 2022

[8] IBM SGBD DB2, IBM CORPORATION. Available in: (<http://www-306.ibm.com/software/data/db2/>) Last visiton: Sept 2003.

[9] Khan, L. et al. A Comparative Study of Storing XML Data in Relational and Object-Relational Database Management Systems. In: INTERNATIONAL CONFERENCE

ON INTERNET COMPUTING - IC, 2002, Las Vegas, USA. Proceedings... Las Vegas: CSREA Press, 2002.

[10] Li, Y. et al. GML Storage: A Spatial Database Approach. In: CONCEPTUAL MODELING FOR ADVANCED APPLICATION DOMAINS. ER2004 WORKSHOP ON CONCEPTUAL MODELING FOR GEOGRAPHIC INFORMATION SYSTEMS (CoMoGIS), 1, 2004, Shanghai, China. Proceedings... Berlin: Springer LNCS 3289, 2004.

[11] MasterMap, OSMasterMap schema repository, Available in: (<http://www.ordnancesurvey.co.uk/oswebsite/xml/schema>). Last visit on: Jul 2004.

[12] Matoso, M. et al. Xvertex: Armazenamento e Consulta de Dados XML em SGBD's. In: SIMPÓSIO BRASILEIRO DE BANCO DE DADOS – SBBDD, 17, Gramado, Brasil. Anais... Gramado: SBC, 2002.

[13] Mchugh, J. Data Management and Query Processing for Semistructured Data. Stanford University, USA, 2000. Tese de DS.C.

[14] MICROSOFT SQL SERVER 2000, MICROSOFT CORPORATION, Available in: (<http://www.microsoft.com/sql/>). Last visit on: Oct 03.

[15] Open Gis Consortium. Geography Markup Language Implementation Specification. Document OGC 02-023r4, 2003, Available in: (<http://www.opengis.org/specs>). Last visit on: Feb 04.

[16] ORACLE SGBD 9i, ORACLE CORPORATION, Available in: <http://www.oracle.com/ip/dep/otn/database/oracle9i>. Last visit on: Mar 03.

[17] Shanmugasundaram, J. et al. Relational Databases for Querying XML Documents: Limitations and Opportunities, In: INT. CONF. ON VERY LARGE DATA BASES – VLDB, 25, 1999, Edinburgh, Scotland. Proceedings... Edinburgh: Morgan Kaufmann, 1999.

ISSN PRINT 2319 1775 Online 2320 7876

© 2012 IJFANS. All Rights Reserved. UGC CARE Listed (Group-I) Journal Volume 11, Issue 2, 2022

[18] Srivastava, D. et al. A Mapping Schema and Interface for XML. IN: INTERNATIONAL WORKSHOP ON WEB INFORMATION AND DATA MANAGEMENT – WIDM, 4, 2002, LcLean, USA. Proceedings... LcLean: ACM, 2002.

[19] Sun Microsystems. Java Architecture for XML Binding (JAXB) Bindings Schema for JAXB. Available in: (<http://java.sun.com/xml/ns/jaxb>). Last visit on: Oct 03.

[20] Tian , F. et al. The Design and Performance Evaluation of Alternative XML Storage Strategies. SIGMOD Record, New York, v.31, n.1, pp. 5-10, 2002.

[21] Tompa, F.W., Salminen, A. Requirements for XML Document Database Systems. In: ACM SYMPOSIUM ON DOCUMENT ENGINEERING, 2001, Atlanta, USA. Proceedings... Atlanta: ACM, 2001.

[22] Xerces, Apache Xml Project. Xerces Java parser, Available in: (<http://xml.apache.org/xerces-j>). Last visit on: Sept 03.

Research paper

1
2
7
5