Storage

Different Approaches for Storing GML*
A study on the effectiveness of storage models in terms of query processing. A performance study is conducted using three data sets and the experimental results are given.

GML Storage: A Spatial Database Approach
To store GML data more efficiently, an approach is proposed to store it in a spatial database.
First the schema tree is generated and then it's mapped to relational schema to store all spatial objects as values of the mapped tables' field. Spatial query can be submitted in XQuery-alike language with spatial functional extensions, and GML query is first translated into equivalent SQL query that is evaluated by spatial database management system.

Storage GML Document: A Model-Mapping Based Approach
An approach to store and query GML, focusing on non-schema documents. First GML is parsed, the document tree is made. Then the nodes are analyzed and schema mapping is generated to store the doc in object-relational DB. Spatial and non-spatial queries are supported.

STUDY ON NATIVE XML DATABASE BASED GML STORAGE MODEL
Considering both characteristics of XML DB and GML spatial data, a native XML database based GML storage is proposed. A prototype system is developed, on the basics of JAXP (Java API for XML Processing) program API and JTS, containing schema mapping constructor, document storage tool, and spatial analyzer.

Query GML

Querying GML Documents: An XQuery based Approach

XQuery language, recommended by W3C, is only applicable to non-spatial data. To query spatial data in GML, XQuery is expanded in data model, algebra, functions and operations, and formal semantics to achieve a GML query processor. The processor can deal with non-spatial and spatial queries, and the results will be outputted in GML.

Defined data types are: Geometry, Coord, Coordinates, Point, LineString, LinearRing, Polygon, Box, GeometryCollection, MultiPoint, MultiLineString, and MultiPolygon and spatial operations are: simple geometry operation, spatial relation operation, spatial analysis operation and specific geometry operation. JTS is called for handle operations

A Specification of a Spatial Query Language over GML

Besides GML is used to represent and exchange the geographic data, it also benefits its XML-based model in interoperability, and more importantly, could be queried. In this study, the data model and algebra behind the language, based on a previously proposed model in which components and interrelations are represented as a directed graph. (The model includes new type of vertex for geometry types and its properties. Egdes represent elements containment, relationship between elements, and values.) With a data model and algebra, data types, and operation, the features that the language needs.

XQuery As a Spatial Query Language

Xquery is not a sutable language to query GML, since spatial related data types and semantics need to be treated different from XML. The purpose of this study is to define expand Xquery not for predefined GML elements, but more flexible. A set of operators and functions on GML data types that cover the most typical queries over spatial data. The difference with the previous one is that this language is applicable not for predefined GML types, but to any type.

GXQuery: A GML Spatial Data Query Language

Integration of GML and Xquery. Adding spatial data types (Point, LineString, LinerRing, Polygon, MultiPoint, MultiLineString, MultiLinerRing and MultiPolygon) and spatial operation functions (basic, relational, analysis,.. functions), based on OGC Simple Features Specification for SQL, to XQuery can achieve GML spatial data query. Moreover to the previous studies in this subject GML spatial data query language based on XQuery, problem of GML spatial data query, reasons for extending XQuery to support GML spatial data query, features of GML spatial data query language, content of XQuery spatial extension, architecture of GXQuery, implementation methods of GXQuery, and query examples of GXQuery were detially studied and discussed.

```
FOR $var1 IN doc("CDUTCampus.gml")//Building,
$var2 IN doc("CDUTCampus.gml")//Building
WHERE $var1/gml:name/node() = "CDUT Palestra"
AND $var2/gml:name/node() = "CDUT Gymnasium"
RETURN geo:Contains(geo:Envelope($var1), $var2)
```

GML-QL: A Spatial Query Language Specification for GML

Expanding data types and spatial operation in Query:

```
FOR $c IN document("Country.xml") //country,
RETURN <Country> <gml:name> $c/Name/text() </gml:name>
          <pop> $c/pop </pop>
           <area> Area($c/shape) </area>
```

</Country>

Development of a Query Language for GML based on Xpath
A semantic version of Xpath language is defined, not based on the tree-based (syntactic) structure of the GML, based on the semantic structure of it.
A system is developed to store GML by PostGIS RDMS and then Xpath queries are translated into SQL, considering the GML schema. Fnaly, the result is represented in the system in KML.

XPath for Querying GML-Based Representation of Urban Maps
A system developed that stores GML by means of PostGIS, and translates Xpath to SQL.
The results would be exported into KML to be visualized.

Querying Heterogeneous Spatial Databases: Combining an Ontology with Similarity Functions
A knowledge-based approach is used to querying heterogeneous spatial databases based on an ontology and conceptual and attribute similarities. The ontology, which may be independent of the databases, expands and filters a user query. Then, queries are translated into a formal specification of entity classes, which are compared against definitions in databases. This process is carried out by determining the conceptual similarity between entities in a user ontology and by comparing these entities in the ontology with entities in the conceptual models of databases. In addition, the specification of a query is done not only by identifying entity classes but also by considering constraints based on attribute values.

Studying an Approach to Query Spatial XML
Towards integrating spatial and non-spatial data, it's necessary to develop an integration sys. For querying the data in different sources. Here a prototype of a mediation system for querying XML spatial resources in GML is studied. The main task of this approach is to provide users with a unique interface for querying spatial XML resources with different schemas, independently of their organization and location. It provides the infrastructure for formulating structured spatial queries by taking into consideration the conceptual representation of a specific domain in the form of an ontology. The resources are integrated using RDF. The most novel and critical feature of this approach is the querying of spatial XML resources, because it uses a different way from that of querying and relating non-spatial resources.

Towards Similarity-based Topological Query Languages
Base on the problem posed in different representation of spatial data in various resources (For example, one dataset M1 may represent roads and bridges as regions, another dataset M2 may represent roads as regions and bridges as lines, a third dataset M3 may represent both as lines), or even in integration scenarios or architectures, a possible solution is to introduce some mechanism of query relaxation, by which approximated answers are returned to the user. In this study, the relaxation problem for spatial topological queries is considered. In particular, some relaxed topological predicates are presented and is show in which application contexts they can be significantly used. In order to make such predicates effectively usable, the way that GQuery, an XML-based spatial query language, can be extended to support similarity-based queries through the proposed operators is also discussed.

Determine all roads overlapping some bridge.

```
for $x in document(bridge.xml), $y in document(road.xml)
where overlap($x/geometry, $y/geometry) = true
```

```
return $x
```

Determine all roads overlapping some bridge, up to a 22% error.

```
for $x in document(bridge.xml), $y in document(road.xml)
where overlapw($x/geometry, $y/geometry,R,L,0.22) = true
return $x
```

Integration of Spatial XML Documents with RDF

An approach to integrate Geospatial data on the Web, storing them in GML, and using a query language. An ontology is used to solve the semantic heterogeneity of different GML documents.

Other

A Library for Managing Spatial Context Using Arbitrary Coordinate Systems
Numerous coordinate systems are used commonly. Conversion between them in order to communicate, inter-operate, and integration of data between various applications and systems is a necessity. A library, using JTS, based on the OGC is developed to cope with different CSs.

Point collection partitioning in MongoDB Cluster

Multi-Criteria Path Finding
An algorithm to find the shortest path between start and end point, based on different criteria, which is compatible with database operations. The result can be stored in DB, also based on the criterias, for future use.

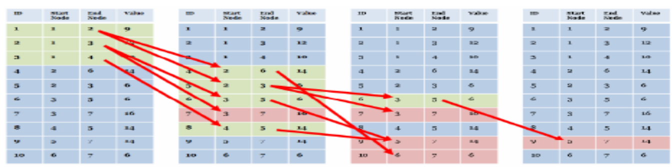

Figure 3. Process of finding all possible paths from 1 to 7

Ontology-based Geospatial Data Query and Integration**
In the geospatial data sharing, GML and data access protocols like Web Future Service (WFS) help enabling clients to access heterogeneous data. Usability of the access is limited, because of the lack of data semantics encoded in WFS feature types. new method to provide integrated access to distributed geospatial data using RDF ontology and query rewriting. This method is more efficient than converting all data to ontology instances because it avoids the costs and consistency problems of data replication. Also, ontology interface can still use existing tools for conducting spatial query on geometries and relational queries on non-spatial data, and for rendering spatial features encoded in GML. Using the proposed method, user queries can be more straightforward because the application ontology can encode data semantics not available in WFS feature types or database schemas. Our method uses RDF ontology but since RDF is a subset of OWL, the method can be directly applied to OWL ontology though extension may be needed to take advantage of the semantic constraints of OWL. OWL has more semantic constructs such as class equivalence/intersection/union, transitive/reflexive object properties, and the restrictions of universal/existential quantification on object properties. Our query rewriting algorithm needs to be extended to handle semantic constraints encoded with these constructs. However, it may be possible to use some existing tools such as Jena in this extension.

A Query-Friendly Compression for GML Document
Since GML documents are mostly in big size, a query-friendly compression method is proposed. The GML Doc. In SAX document parsing are transformed into a compact representation, encompassing an event dictionary, the event hierarchy, a binary event tree, and the doc. Content blocks before compressed using a general compressing utility. The method supports direct path queries and spatial queries, without the requirement of a full decompresion.

Advances in GML for Geospatial Applications*

A study on GML from different views for using it in spatial applications, such as storing, querying, parsing, and visualizing. Also, using it in mobile devices and web services are surveyed.