# STUDY OF DIFFERENT APPROACHES TO THE INTEGRATION OF SPATIAL XML WEB RESOURCES

José E. Córcoles, Pascual González
*Departamento de Informática Universidad de Castilla-La Mancha Campus Universitario s/n.02071.Albacete. Spain.*

Keywords: Semantic Web, Integration, Spatial XML

Abstract: The research community has begun to investigate foundations for the next stage of the Web, called *Semantic Web*. Current efforts include the Extensible Markup Language XML, the Resource description Framework, Topic Maps and the DARPA Agent Markup Language DAML+OIL. A rich domain that requires special attention is the Geospatial Semantic Web. However, in order to approach the Geospatial Semantic Web, it is necessary to solve the problem of developing an integration system for querying spatial resources stored in different sources. In this paper, we study two different approaches to integrating spatial and non-spatial information represented in the Geographical Markup Language (GML). The approaches studied follow LAV (Local as View) integration. With this study we obtain the best approach to developing a real system for querying GML resources stored in different sources.

## 1 INTRODUCTION

A domain that requires special attention is the Geospatial Semantic Web (Egenhofer et al. 2002). The enormous variety of encoding of geospatial semantics makes it particularly challenging to process requests for geospatial information. Work led by the OpenGIS Consortium (OpenGIS, 1999) addressed some basic issues, primarily related to the geometry of geospatial features.

In order to approach the Semantic Geospatial Web, it is necessary to solve the problem of developing an integration system for querying spatial resources stored in difference sources. The user should view a *virtual* spatial data repository in a given domain without knowledge of the source in which each item of data is located.

Tackling the integration of spatial information on the Web is not a simple task since, for example, the sources may store large amounts of incomplete spatial data, which may make it necessary to join the results of queries with spatial joins. Thus, it is necessary to apply efficient query processing strategies that allow spatial joins to be made on the Web (Shahabi et al. 2003). Therefore, the application of spatial operators means a different treatment with respect to the integration of XML resources with only alphanumeric (non-spatial) data.

The main aim of this paper is: (1) to study two architectures for integrating Spatial XML resources (GML) obtained by modifying two existing approaches, and (2) to compare each one, with the aim of obtaining the best approach for developing a real system for querying GML resources stored in different sources. In our study the spatial information is represented in the sources by GML because it is an XML encoding for the transport and storage of spatial/geographic information, including both spatial features and non-spatial features. The mechanisms and syntax that GML uses to encode spatial information in XML are defined in the specification of OpenGIS (OpenGIS, 2003). Thus, GML allows a more homogeneous and flexible representation of the spatial information.

Query mediation has been extensively studied in the literature for different kinds of mediation models and for the capabilities of various sources: in the field of non-spatial integration there are several approaches such as *Tsimmis* (Papakonstantinou et al. 1995), *Information Manifold* (Levy et al. 1996). More directly concerned with the integration of



E. Córcoles J. and González P. (2004).
STUDY OF DIFFERENT APPROACHES TO THE INTEGRATION OF SPATIAL XML WEB RESOURCES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 364-369
DOI: 10.5220/0002613303640369

XML resources, it is worth noting C-Web Portal (Amann et al. 2001) and (Amann et al. 2002). C-Web Portal supports the integration of non-spatial resources on the Web, and C-Web provides the infrastructure for (1) publishing information sources and (2) formulating structured queries by taking into consideration the conceptual representation of a specific domain in the form of an ontology. On the other hand, (Amann et al. 2002) proposes a *mediator* architecture for the querying and integration of Web-accessible XML data resources (non spatial data). Its contribution is the definition of a simple but expressive mapping language, following a *local as view* approach and describing XML resources as local views of some global schema.

In relation to spatial data integration, there are approaches developed by (Gupta et al. 1999) and (Boucelma et al. 2002). (Gupta et al. 1999) extends the MIX wrapper-mediator architecture for integrating information from spatial information systems and searchable databases of geo-referenced imagery. (Boucelma et al. 2002) presents a mediation system that addresses the integration of GIS data tools, following a GAV(Global as View) approach.

In order to design approaches for querying spatial GML resources, we have based our work on two existing studies: (Amann et al. 2002) and (Amann et al. 2001) mentioned above. We have selected the first approach ((Amann et al. 2002)) because it is focused on integrating XML resources, and it can be extended in a simple way to query GML resources with spatial operators. The second approach has been selected because it is an interesting approach that makes it possible to query different resources on the Web. By this we mean that modifying it adds the possibility of querying GML resources. In addition, both approaches follow a LAV (Local as View) integration. The LAV approach facilitates the maintenance of the integrated schema and mediation, although query evaluation is far more complex than the global-as-view approach where the integrated schema is defined in terms of source schemas. The LAV approach is therefore favoured in the context of the integration of resources that change significantly over time, such as Web resources. In short, by modifying these contrasted approaches we exploit the solution to query XML data on the Web.

The overview of both architectures and the modification applied to query spatial information are shown in Section 2 and Section 3. In Section 4 we conclude with a comparison of the two approaches, emphasising the most important advantages and disadvantages, in order to obtain the best approach for developing a real system for querying GML resources stored in different sources.

# 2 APPROACH BASED ON RDF

A *Community Web Portal*(Karvounarakis et al. 2000)*(C-Web)* essentially provides the means to select, classify and access, in a semantically meaningful and ubiquitous way, various information resources (sites, documents, data) for diverse target audiences (corporate, inter-enterprise, …). The core Portal component is a *Catalog* holding descriptions, i.e. metadata, of the resources available to the community members. In order to effectively disseminate community knowledge, *Portal Catalog* organises and gathers information in a *multitude of ways*, which are far more flexible and complex than those provided by standard (relational or object) databases. It uses the Resource Description Framework (RDF) standard (Brickley et al. 2000) proposed by W3C, designed to facilitate the creation and exchange of resource descriptions between Community Webs. In order to query the *Catalog*, a query language, called RQL, is presented in (Alexaki et al. 2001) which allows semistructured RDF descriptions to be queried using taxonomies of node and edge labels defined in the RDF schema.

In order to integrate the spatial information of several spatial XML documents (GML), we have based our work on the *Community Web Portal* concept (Amann et al. 2001) with RDF and RQL, a declarative language for querying both RDF descriptions and related schemas. To perform this integration, it is necessary to make some modifications to the original approach because in the original approach the *Catalog* is considered as a collection of resources identified by URIs and it is described using properties. However, it does not need to use operator over the resources, only over the properties.

GML documents (or part of) are a resource. Unlike the original approach, it is possible to apply spatial operators (comparatives: cross, overlap, touch; analysis: Area, Length) over the resources provided they represent geometry information with GML. In order to take advantage of this fact, we have designed two modifications with respect to the original approach:

Extension of RQL to support spatial operators over the resources that represent spatial documents or part of spatial documents. These operators must be the same as those defined in (Corcoles et al. 2001) for a query language over GML. There are two types of operators: methods for testing Spatial Relations and methods that support Spatial Analysis. This extension is not dealt with in this paper.

Extension of the *Community Web Portal* architecture to support the application of the spatial operators

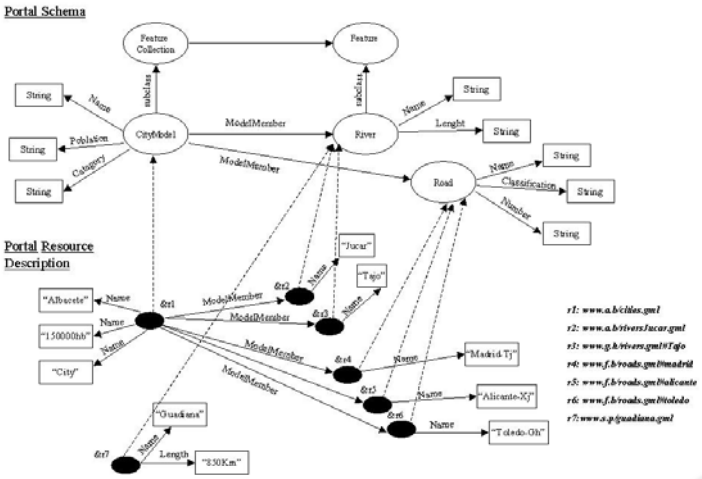


Figure 1: Portion of Catalog of a CityModel.

over the resources involved in the query, and the integration of all information to be returned.

## 2.1 An Overview

In this section, an overview of the application of the mediation system is given, looking at the system from the point of view of the user. In Figure 1, an example of the Portal Schema and its instances is shown. The example has been obtained from the specification documents of GML. Due to RDF's capability for adding new feature and geometry types in a clear and formal manner, this example has been carried out extending the geospatial ontology defined by OpenGIS, where the class (Geometry, LineString, etc) and properties (coordinates, PolygonMember, etc) are defined. The example shows an extension of a Catalog for a CityModel proposed by (OpenGIS, 2003) and called Cambridge.rdfs.

This is a well-known example used in all specifications over GML and developed by OpenGIS. The Cambridge example has a single feature collection of type 'CityModel' and contains two features using a containment relationship called 'modelMember'. The model member can be Rivers that run through the City or Roads belonging to the City.

An example of a query expressed in RQL extended with spatial operators may be as follows:

"Find all Road resources that belong to Albacete City and which are within 50 meters of a River of this city ".

In RQL:

Select Z
From {X}ModelMember{Y:River}, {W}ModelMember{Z:Road}, {P}name{Q}
Where X=W and X=P and Q="Albacete" and Crosses(Buffer(Y,50),Z)

In a different way to the original approach, this query has a part that is executed directly over the *Catalog*, and another part that is executed over the objects *Road* and *River* stored in the respective sources. In order to do this, it is necessary to establish a spatial query plan. On the other hand, if the query uses operators like *Area, Length, Union, Intersection*, etc., this approach manages the new created resources. The results could be the resources (&r2 ,&r3).

Although the resources may be of different types (documents, HTML files, Raster image,..), in this approach the semantic of the spatial operators is only applied over the geometry objects based on the OpenGIS specification (OpenGIS, 1999).

This approach can be studied at length in [CG03]).

# 3 APPROACH BASED ON MAPPING

In this section we present a general overview of the second approach. This work has been inspired by (Amann et al. 2002), which proposes a *mediator* architecture for the querying and integration of Web-accessible XML data resources (non spatial data). Its contribution is the definition of a simple but expressive mapping language, following a *local as view* approach and describing XML resources as

local views of some global schema. This approach offers its users a *virtual* data repository in a given domain. This repository is *virtual* because the real data resides in some external sources. However, the users of the repository are not concerned with the source location and source data organisation.
Obviously, our aims (integrate spatial sources) are different from those of the previous work by (Amann et al. 2002) (integrate non-spatial sources). For this reason, (Amann et al. 2002) has been extended in order to satisfy our requirements.
We present in the following section an overview of the system architecture offered by (Amann et al. 2002). In subsection 3.2, we detail the main modification of this approach to achieve spatial queries.

## 3.1 Overview

The main task of an integration mediator is to provide users with a unique interface for querying the data, independently of its actual organisation and location. This interface, or global schema, is described as an *ontology*. As used here, an *ontology* denotes a light-weight conceptual model and not a hierarchy of terms or a hierarchy of concepts (in the same way as in the first approach). The global schema can be viewed as a simple object-oriented data model. Hence, a global schema can be viewed as defining a database of objects, connected by roles, with the concept extents related by subset relationships as per the *isA* links in the schema. Since it is an integration schema, this is a virtual database. The actual materialisation exists in the sources.
To evaluate a user query expressed in terms of the ontology, the approach translates it into one or more queries on the XML sources. For this purpose, we need to establish a correspondence between each source and the global ontology. This correspondence is described by a *mapping*, which is a collection of *mapping rules* (path-to-path).
The description of the global schema in terms of the ontology allows users to formulate structured queries, without being aware of the source specific structure. (Amann et al. 2002) illustrates querying with the query language defining *tree queries*. Although these queries have some limitations (e.g. joins between variable are not allowed (Amann et al. 2002)), they are sufficiently powerful to illustrate the issues of answering queries from XML source data (but it is not sufficient to answer spatial queries).
In order to process each query, two query processing cases are possible. In the first case, the solution to a query is the union of the complete answers from individual sources. If no complete answer can be obtained from a source, then the source is abandoned. This simple strategy has obvious advantages, as it only needs a variable binding algorithm and a simple query execution plan for searching all sources for which there exists a full binding. In contrast, the second case also allows for incomplete answers from a given source. If a source *s* can only partially answer a query, then the query is decomposed into two parts, one to be fully answered by *s* and the other part being sent to the other sources. In this case, it needs a variable binding algorithm and a query execution plan that includes query decomposition for searching all sources for which there exists a full/partial binding.

## 3.2 Modification to query spatial resources

Our modification includes two new components: (i) *spatial system*, used to make the joins between the results of spatial queries and (ii) *Spatial DBMS*, where the spatial GML documents are stored over ORDBMS (in the same way as the first approach). In addition, other modifications have been made to the functionality of the existing components: (iii) extends the features of the query languages including spatial operators and including spatial and non-spatial joins, and (iv) extends the query execution plan to query spatial joins (using the *spatial system* component).
The query language used for querying in the original approach should support spatial and non-spatial operators with a user-friendly interface. Therefore, in our modification we use another spatial query language for GML (Corcoles et al. 2001) as the query language used by the users and for querying the different sources. Thus we have simplified the translation between the user query and the queries executed in each source. It is an advantage with respect to (Amann et al. 2002). (Note that in the original approach, a query language based on OQL is shown.) A query is a simple *tree query*, based on **select-from-where** clauses. We assume queries satisfy the following restrictions regarding the original algebra.
First, over the variables in the *select* clause it is possible to apply spatial operators supported in the original algebra: methods for testing Spatial Relations and methods that support Spatial Analysis(Corcoles et al. 2001). Second, as is mentioned above, in the original approach the **where** clause is a conjunction of simple predicates, where a simple predicate is of the form $x\ \theta\ d$ in which $\theta \in \{=, <, >, \leq, \geq\}$ and $d$ is an atomic value. Thus, it is not possible to express joins by equalities between

variables, i.e. by predicates of the form $x_i = x_j$. In our case, this limitation restricts the expressive power of the query language and of this application. For this reason, we have incorporated the possibility of including spatial joins in the *where* clause ($x_i \phi x_j$ | $\phi$ is a spatial join operator defined by (Corcoles et al. 2001); cyclic joins are not allowed). This facility makes it more difficult to evaluate the queries than in the original approach. For this reason, the definition of a new algorithm (shown below) for evaluating the query is necessary. Third, Spatial Operators *intersection*, *union*, *difference and symdifference* are not included. This restricts the power of the query language but simplifies the evaluation of queries. Last, the language has no quantifiers, aggregates, or subqueries.

The result of such a query is a set of tuples of the form {[$a_i$, $a_j$, …$a_k$]}where $a_i$, $a_j$, …$a_k$ are instances of the variables in the query's select clause and can be either atomic values or GML fragments.

Due to the extension of the original approach with spatial and non-spatial *joins*, a third query processing case should be added in this modification. This case allows for incomplete answers from a given source and all variables involved in *spatial join operators* (or non spatial) are found in different sources. It is the most complete and complex case. It needs (i) a query execution plan that includes query decomposition for searching all sources for which there exists a full/partial binding, (ii) a solution for joining the results of each partial query, and (iii) a strategy for performing spatial joins on the web sources. For this reason we have developed a new query execution plan, which is described in the following section.

## 3.3 Query Execution Plan

This Section describes a query execution plan that includes query decomposition for searching all sources for which there exists a full/partial binding and a strategy for performing spatial joins on the web sources.

In a distributed environment with restricted access to the remote servers (the remote servers are read-only), performing spatial join queries must be simulated by spatial select operations after transferring whole/partial data sets from the local to the remote server. In addition, the query response time is a function of size and complexity of the data transferred between the servers. (Shahabi et al. 2003) provides an approach to performing spatial joins in a Web environment. Based on this work, we have adopted the following query processing strategy to solve spatial joins. *Local* refers to the *mediator* and *Remote* refers to the remote sources. (i) Local: Local to Remote Transfer {*Dinamic-MBR*}, (ii) Remote: Spatial Selection {Window-Selection}, (iii) Remote: Send to Local {Candidate Objects} and, (iv) Local: Refinement {Pipelined}.

Given a set of sources S and a query Q, the algorithm P(Q) shown in figure 2 computes a Query

```
Input: a query Q and a set of sources S
Output: a query execution plan for Q
Algorithm:  QEP(Q,S) = 0;
  For all sources s ∈ S {
    If B(Q,s) <> 0 {
      /* There exist at least one maximal binding for Q in s */
      for all bindings β ∈ B(Q,s) {
        if β is full binding P(β) : = Q;
        else { P(β) : = Qp(β); //Qp is the prefix Query
          for all Spatial join divided in Qp
              set SJ = SJ ∪ [v1,v2,Op]
          for all suffix queries Q´∈ QS(β){
            if P(β) <> 0
              /*there exists a non-empty query plan for all subqueries up to Q´*/
              if (QEP(Q',S) <>0
                /*there exist a query plan of Q'*/
                if (QEP(Q',S) satisfy some join in SJ)
                    P(β) = P(β) ⊗k ∧ joins  QEP(Q',S)
                else
                    P(β) = P(β) ⊗k  QEP(Q',S)
              Else P(β) = 0
        }
        QEP(Q,S) = QEP(Q,S) ∪ P(β)
      }
    }
  }
  return QEP(Q,S)
```

Figure 2: Query Execution Plan Generation.

Execution Plan for Q. For each source *s* and a maximal binding $\beta \in B(Q,s)$, a QEP P(β) is computed: if β is a full binding (i.e complete answers are obtained), the result is query Q. Otherwise, if β is a partial binding, then query Q is decomposed into a prefix query $Q_p(\beta)$ and a set of suffix queries QS(β). All spatial joins divided in the prefix query are registered. It is necessary to know when a partial query satisfies a spatial join and then carry out the join in the QEP. The query execution plan of Q against source *s* is obtained by joining $Q_p(\beta)$ with the query execution plan for each suffix query $Q' \in QS(\beta)$ (variable *k* denotes the key query variables of Q´). To calculate the query execution plan of a suffix query Q', the algorithm is called recursively. Finally, the plan obtained is added to the existing plan by union.

Note that there are two reasons for interrupting the calculation of a query execution plan for a given source *s* and binding β. The most trivial case is that there exists no maximal binding for Q in *s*. The second reason is that there exists at least one suffix query which cannot be satisfied (empty query execution plan).

# 4 CONCLUSIONS

Each approach has advantages and disadvantages. With the modification incorporated in the first approach it is possible to obtain an architecture that enables the integration of different kinds of

resources (documents, sites, GML resources,...) using a unique query language RQL that allows querying of a *Catalog* with references to all resources. In addition, this approach (and the second approach) allows GML resources to be queried efficiently in each source, because they store the GML documents in ORDBMS(Corcoles et al. 2002). These are just some of the features offered by the first approach.

No doubt, this approach is the best approximation to the Geospatial Semantic Web, querying all kinds of resources in the same way. However, this alternative has several disadvantages: (1) the query language used by the user (RQL) is different from the query language used by the *wrappers*. For this reason, several conversions are necessary between query languages (RQL to QL over GML and QL over GML to QL of the DBMS); (2) the rewriting algorithm to translate the RQL query to XML queries over the local sources is complex to implement.

The second modification offers a more powerful alternative for querying spatial resources. It solves some disadvantage of the first approach: (1) the use of only one query language to query the ontology (users) and to query the GML resources in the wrappers. The approach architecture allows a more powerful query language than the modification of RQL; (2) different strategies are contemplated in this approach, including spatial joins between objects localised in resources of different sources. This approach uses strategies for carrying out joins efficiently between large spatial data on the Web.

Finally, both approaches also has the disadvantage of administering the mapping rules between the ontology and the sources.

In conclusion, an approach with advantages of both alternatives is the more powerful alternative for developing a system to query spatial XML resources on the Web. The implementation of a usable prototype has already been achieved.

## REFERENCES


Amann B., et al.. 2002 Ontology-Based Integration of XML Web Resources. In International Semantic Web Conference (ISWC), Sardinia, Italy.

Alexaki. S et al. 2001. "The ICSFORTH RDFSuite: Managing Voluminous RDF Description Bases". In Proceedings of the 2nd International Workshop on the Semantic Web (SemWeb'01), in conjunction with WWW10, pp. 1-13, Hong Kong.

Amann, B. et al. 2001. Mapping XML Fragments to Community Web Ontologies. In Proc. Fourth International Workshop on the Web and Databases.

Boucelma, O, et al. 2002. A WFS-Based Mediation System for GIS Interoperability. ACM-GIS 2002. 10th ACM International Symposium on Advances in Geographic Information Systems. McLean (USA).

Brickley, D , et al. 2000. Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation. Technical Report CR-rdf-schema-20000327, W3C, Available at http://www.w3.org/TR/rdf-schema.

Córcoles J, et al. 2001. A Specification of a Spatial Query Language over GML. ACM-GIS 2001. 9th ACM International Symposium on Advances in Geographic Information Systems. Atlanta (USA).

Córcoles J, et al. 2002. Analysis of Different Approaches for Storing GML Documents ACM-GIS 2002. 10th ACM International Symposium on Advances in Geographic Information Systems. McLean. (USA).

Córcoles J, et al. 2003. Querying Spatial Resources. An Approach to the Semantic Geospatial Web. CAiSE'03 workshop (WES2003). To Appear in Lecture Notes in Computer Science (LNCS) by Springer-Verlag.

Egenhofer M. 2002. Toward the Semantic Geospatial Web. ACM-GIS 2002. 10th ACM International Symposium on Advances in Geographic Information Systems. McLean (USA).

OpenGIS. 2003. Geography Markup Language (GML) v3.0. http://www.opengis.org/techno/documents/02-023r4.pdf.

Gupta, A, et al.. 1999. Integrating GIS and Imagenery through XML based information Mediation. Integrated Spatial Databases: DigitalImages and GIS. Lecture Notesin Computer Science. Vol1737. Pp. 211-234. Springer-Verlag.

Karvounarakis, G, et al. 2000. Querying community web portals. Technical report, Institute of Computer Science, FORTH,Heraklion, Greece. http://www.ics.forth.gr/proj/isst/RDF/ RQL/rql.pdf.

Levy,A. Y, 1996. Querying Heterogeneous Information Sources Using Source Description. In Proc. of the Int. Conference on Very Large Databases, pp.25-262. India.

OpenGis Consortium. 1999. Specifications. http://www. opengis.org/ techno/specs.htm

Papakonstantinou, Y, et al. 1995. Object Exchange Across Heterogeneous Information Sources. In Proc. ICDE Conf. TSIMMIS project: http://www-db.standford,edu/tsimmis.

Shahabi, c, et al. 2003. Alternative Strategies for Performing Spatial Joins on Web Sources, To Appear in Journal of Knowledge and Information Systems (KAIS) by Springer-Verlag,