GEO-OBJECT CATALOGS TO ENABLE GEOGRAPHIC DATABASES INTEROPERABILITY

Daniela F. Brauner, Marco A. Casanova and Carlos J. P. de Lucena
Departamento de Informática – PUC-Rio, Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ, Brasil - CEP 22453-900

Abstract: This paper introduces the concept of Ontology based Geo-Object Catalog (OGOC). An OGOC acts as a focal point for a federation of independent geo-object sources, providing consolidation services for the geo-objects represented in several, as well as access services to query geo-objects and metadata. To accomplish this task, an OGOC stores a reference ontology, which also includes reference geo-object instances, and ontologies representing the conceptual schemas of the data sources. The paper also considers a framework for the creation of personalized OGOCs.

Key words: ontology, catalog, database interoperability.

1. INTRODUCTION

The growing number of independent object sources, easily accessible through the Web, has only exacerbated the heterogeneity problem. More precisely, accessing independent object sources requires remapping structurally heterogeneous objects, that is, objects that are organized following different conceptual schemas, as well as semantically heterogeneous objects that is, objects with distinct meanings. Structural and semantic heterogeneity problems are challenges that the distributed databases community has long faced (Özsu and Valduriez, 1999).

To enable interoperability, remote systems must be able to locate and access object sources, and to interpret and process the objects. One solution proposed by the database community relies on generating mappings between pairs of conceptual schemas. However, this pairwise mapping between schemas becomes impracticable if the number of object sources is fairly large. Another solution consists in adopting a global schema. In this case, each conceptual schema has to be mapped into the global schema. Yet another approach, advocated more recently, is to use ontologies to expose implicit knowledge (Wache et al., 2001; Uschold and Grüninger, 2001) thereby enabling interoperability.

Mena et al. (2000), proposes a semantic based integration approach that uses multiple ontologies, instead of an integrated view. In this context, ontologies are virtually linked by interontology relationships, which are then used to indirectly support query processing.

Reed and Strongin (2004) propose a new service for generalized distributed data sharing and mediation using XRIs (eXtensible Resource Identifiers). The goal of XDI is to enable data from any data source to be identified, exchanged, linked and synchronized into a machine-readable *dataweb* using XML documents.

However, independently of the approach adopted to map schemas or ontologies, it might be impossible to define mappings between objects from distinct sources. For instance, consider two object sources about enterprise installations, maintaining information about buildings, industrial installations, etc. Suppose that each source uses its own installation identifier, say, one uses the address as identifier, and the other uses an installation code. Suppose also that neither stores both the address and the code of the installations. It is then obvious that, given an installation identified by its address, it becomes impossible to locate the same installation by its code, and vice-versa. Note that this is true even if one aligns the objects classes in both sources. In fact, without explicit mappings between the object instances, these two data sources cannot interoperate. This problem can only be addressed if an object catalog is defined that explicitly stores object instance mappings.

This paper then proposes the concept of Ontology-based Geo-Object Catalog (OGOC), as a strategy to address the interoperability problem between geographic object sources. We mean by geographic object, or *geo-object*, any data that has some information about its spatial location (we will avoid using the term "feature" in this paper). The OGOC will act as a generalized mediator for a federation of geo-object sources, providing services to access and search for federated data and metadata. To meet this requirement, the catalog will store: (1) a *reference ontology*, similar to a global conceptual schema; (2) *local ontologies* describing object sources; (3) *ontology mappings* from the local ontologies to the reference ontology; (4) sets of instances of geo-objects, acting as *standard geo-objects*: (5) *instance* *mappings* from reference geo-objects to the geo-objects stored in each source.

In short, an OGOC enables interoperability among the federated geo-object sources on both the data and the metadata levels.

The OGOC concept generalizes and combines the OpenGIS Consortium (OGC)[1] catalog and *gazetteer* notions, referenced in this text, respectively, as OGC Catalog and OGC Gazetteer.

An OGC Catalog is a collection of descriptive information (metadata) about data stored in a geographic database (Nebert, 2002). Thus, metadata describes the properties that can be queried and requested through catalog services. An OGC Catalog provides discovery, access and management services, allowing the user to locate and modify metadata, and to request services on the data.

The OGC Gazetteer (Atkinson and Fitzke, 2002) is a spatial dictionary of objects with geographic attributes. Each instance of a *gazetteer* service typically covers a limited region, such as a country, and has an associated vocabulary of geo-object identifiers. An OGC Gazetteer provides operations to retrieve:

- the service description and geo-object types that it can handle (getCapabilities);
- the schema definition of a geo-object type (describeFeatureType);
- sets of geo-objects (getFeature).

This paper is organized as follows. Section 2 introduces the concept of Ontology-based Geo-Object Catalog. Section 3 presents a generic architecture for an OGOC, briefly introduces a framework to generate customized OGOCs, and describes scenarios where an OGOC can profitably be used. Section 4 addresses additional functionalities that an OGOC should have. Finally, Section 5 contains the conclusions and suggestions for future work.

2. ONTOLOGY-BASED GEO-OBJECT CATALOGS

An Ontology-based Geo-Object Catalog (OGOC) acts as a focal point for a federation of independent geo-object sources, providing consolidation services for the geo-objects represented in several sources, as well as access services to query objects and metadata.

To accomplish this task, an OGOC stores a *reference ontology* and ontologies representing the conceptual schemas of the geo-object sources. The reference ontology may include a set of reference instances.

[1] OpenGIS Consortium: http://www.opengis.org

A *reference instance* represents, in a consolidated way, a collection of geo-objects, stored in distinct sources, that refer to the same real-world object. Each reference instance belongs to a class in the reference ontology and carries attributes that identify the equivalent geo-object in each source. In this way, the catalog provides means to determine equivalent classes and equivalent geo-objects from distinct sources.

Figure 1 schematically shows the use of the reference ontology (R_O) and of the reference instances (I_R) to map, respectively, the local ontologies and the local geo-objects from two federated geo-object sources.

The mapping between the local ontologies and the reference ontology will be based on generic relationships, such as subClassOf and sameClassAs, and on relationships that depend on the application domain.

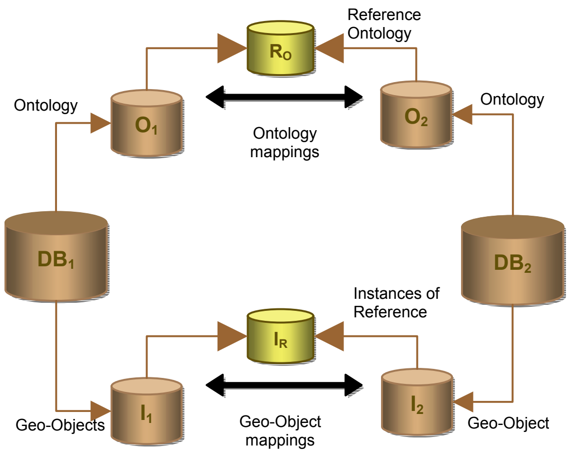

Figure 1 - Geo-objects and ontologies mappings.

An OGOC may be used to mediate access to the geo-object sources, acting as the pivot of a federation of objects sources. To be part of the federation, each source must follow the *participation protocol*: (1) register the source in the OGOC, informing the source's conceptual schema (an ontology, in the OGOC ontology description language); (3) map the source ontology to the reference ontology; and (4) provide the set of geo-objects, with its basic attributes, that the source wishes to share, in the context of the federation.

To summarize, an OGOC:

- covers a specific application domain, where interoperability among geo-object sources is needed;
- provides a reference ontology for the application domain, with classes, basic properties and relationships definitions;

- implements an ontology repository to store and manage the ontologies describing the federated geo-object sources, as well as the reference ontology;
- offers a query language to query and manipulate ontologies and instances.

3. OGOCS REFERENCE ARCHITECTURE AND FRAMEWORK

3.1 OGOCs Reference Architecture

The OGOCs reference architecture, illustrated in Figure 2, is composed of the following modules:

- **Interface:** exposes the OGOC operations.
- **Ontology Manipulation Module:** implements OGOC operations to manipulate ontologies (metadata) of federated geo-object sources. The operations will be defined as a generalization of the OGC Catalog service interface specification:
  - **Discovery services:** provide methods to locate ontologies on the OGOC repository;
  - **Access services:** provide methods to request services on data;
  - **Management services:** define methods to update ontologies in the repository;
- **Geo-Object Manipulation Module:** implements the OGOC operations to manipulate reference instances. The operations will be defined as a generalization of the OGC Gazetteer interface specification:
  - **GetCapabilities:** returns the service description provided by the OGOC, including a list of supported object types and its respective supported operations;
  - **DescribeObjectType:** returns a description of an object type;

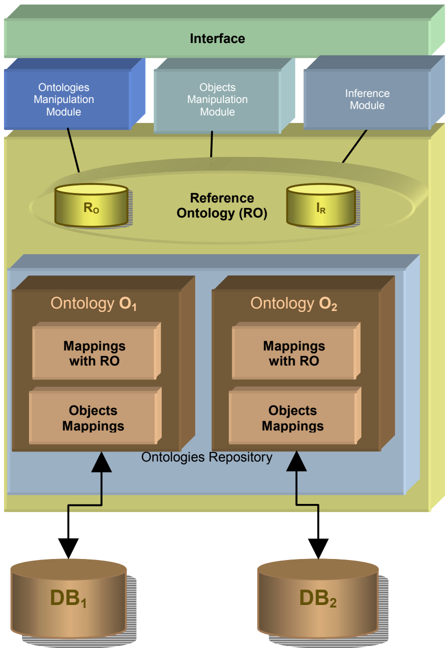

Figure 2 - OGOC generic architecture.

- **GetObject:** provides queries to sets of objects, defined using filters, based on the properties of the geo-object types. A special parameter may be added to enable the use of an ontology-driven cooperative behaviour strategy that helps the requester achieve its goals, by allowing the system to apply transformations to queries and answers.
- **Inference Module:** provides inference services that allow more sophisticated queries to be formulated against the catalog geo-objects and metadata.
- **Reference Ontology:** describes, in the federation application domain, classes, basic properties and equivalence relationships that provide an integrated vision of the ontologies of the federated geo-object sources.
- **Ontology Repository:** stores and manages ontologies that describe federated geo-object sources and OGOC reference ontology.

3.2 A Framework for Customized Catalogs

The generic architecture described in the previous section suggests the definition of a framework (Fayad and Schmidt, 1999) that generates customized catalogs by instantiation.

The frozen-spots and hot-spots of the framework are defined to accommodate changeable characteristics of the application domain.

The proposed framework has the following frozen-spots:

- **Ontology Language:** OWL[2] is the ontology language used to formalize the ontologies stored in the catalog;
- **Ontology access and management operations:** operations to access and manage ontologies will be implemented from a generalization of the OGC Catalog service interface specification to metadata manipulation;
- **Objects access and management operations:** operations to access and manage objects (instances) will be implemented from a generalization of the OGC Gazetteer service interface specification to data manipulation;

The framework hot-spots are:

- **Catalog access interface:** the interface to access the catalog services;
- **Ontology repository:** the implementation of the ontology repository.
- **Query language:** the query language used to query ontologies in the ontology repository;
- **Reference ontology:** the reference ontology, specific to the catalog application domain.

4. OGOC USAGE SCENARIOS

In this section we describe three usage scenarios of OGOCs, using the geographic application domain.

Scenario 1:

In this scenario, we describe the use of the OGOC as a mediator for federated geo-object sources. Consider a user who wants to know facts about the city of "Rio de Janeiro". The catalog is invoked through the *GetObject* operation using the place name as the geo-object identifier in the user query (passed as a parameter of *GetObject*). The catalog will execute the query against its ontologies repository, locating all objects from the "city" type that have "Rio de Janeiro" as place name. The answer may include objects from distinct sources registered in the catalog. This is possible because the catalog stores mappings linking equivalent geo-objects from distinct sources, even if they pertain to different classes.

Scenario 2:

This scenario covers the case of a user that needs to discover geo-object sources that have classes that map to a specific geo-object class *G* of the reference ontology. In this scenario, the catalog is invoked through the *DescribeObjectType* operation, having *G* as one of its parameters. The expected answer is a XML representation of a set of triples of the form *(G, p, R)*, where *p* is a property that maps *G* into range *R*, defined in the reference

[2] OWL - Web Ontology Language: http://www.w3.org/TR/owl-ref/

ontology. In particular, p may represent a relationship between G and some class H defined in the ontology of one of the geo-object sources.

For instance, consider a user that needs information about "cities" in the world. The catalog will execute the query against the ontology repository to discover where to find objects of the class "city". The answer of the *DescribeObjectType* operation will be metadata about "city", as explained above, that enables the user to locate and access all geo-object sources, registered in the catalog, that contain information about cities.

Scenario 3:

The catalog can be used to provide standard metadata to a user who is designing a new geo-object source that will participate in the federation. The usage scenario is very similar to the second scenario and will not be repeated here.

5. ADDITIONAL CAPABILITIES

All scenarios described in section 3.3 address the use of an OGOC to enable interoperability between geographic databases. They also suggest additional capabilities that can be added to the catalog enabling a cooperative behavior.

Indeed, the catalog may help users achieve their goals through request and answer transformations. This cooperative behavior will be useful to correct or reformulate user queries and system answers.

The cooperative behavior may be accomplished with the help of the ontologies stored in the catalog, enhanced with an inference engine.

Among other functions, a cooperative environment may (Hemerly et al., 1993):

- Correct a query;
- Resolve ambiguities;
- Generate alternatives to accomplish, as much as possible, the same purpose as the original query, when it fails;
- Provide explanations about an answer;
- Complement a query by supplying additional information.

Consider a catalog about geographic places in the World. An example of an ambiguous query would be "Find Rio de Janeiro". Figure 3 illustrates a city and a state with this name in Brazil. Figure *4* shows a second city, called "Rio de Janeiro", this time in Peru.

Figure 3 - Example of "Rio de Janeiro" in Brazil.

Figure 4 - Example of "Rio de Janeiro" in Peru.

SOURCE: http://www.expedia.com/

Indeed, a request for the place name "Rio de Janeiro" sent to the ADL Gazetteer Server Client[3] will generate an ambiguous answer, which requires user intervention to be disambiguated. Table 1 shows the answer returned by the ADL Gazetteer, with the eight different feature types associated with places named "Rio de Janeiro".

However, note that the answer could be automatically disambiguated using an associated vocabulary, such as a *feature type* ontology.

[3] http://middleware.alexandria.ucsb.edu/client/gaz/adl/

Table 1 - Feature types of "Rio de Janeiro" instance in ADL Gazetteer Server.

Place Names	Feature Type
Rio de Janeiro, Igarape - Acre, Estado do - Brazil	streams
Rio de Janeiro - Brazil	populated places
Janeiro, Rio de - Brazil	streams
Rio de Janeiro - Loreto, Departamento de - Peru	populated places
Rio de Janeiro, Serra - Paraiba, Estado da - Brazil	mountains
Rio de Janeiro, Estado do - Brazil	administrative areas
Janeiro, Rio de - Brazil	streams
Rio de Janeiro, Serra do – Brazil	mountains

SOURCE: ADL Gazetteer Server Client

As a second example, we illustrate how to correct queries, using the reference ontology.

Consider the following user query: " *Give me the State of Peru that has a City called Rio de Janeiro located in"*. Suppose that we use RDQL[4] (RDF Data Query Language) as the catalog query language.

We note that one of the mayor advantages of RDQL is its ability to mix both metadata and instances in the same query.

Then, the above query would be formulated in RDQL as follows:

```
SELECT ?state
WHERE   (?id, <geo:name>, "Rio de Janeiro")
   (?id, <geo:isLocatedIn>, ?state)
   (?state, <rdf:type>, <geo:State>)
   (?state, <geo:isLocatedIn>, ?country)
   (?country, <rdf:type>, <geo:Country>)
   (?country, <geo:name>, "Peru")
```

The answer to this query will be NULL, because the first order division of Peru is not by states, but by "Departmentos".

This failure can be avoided using a relaxation technique (Chu and Mao, 2000), applied to ontologies. The relaxation technique consists in navigating thru the class hierarchy to generalize or specialize the query terms, expanding or reducing the scope of the user query.

The modified query will then be " *Give me the Departamento of Peru that has a City called Rio de Janeiro located in"*, which in RDQL is:

```
SELECT ?state
WHERE   (?id, <geo:name>, "Rio de Janeiro")
   (?id, <geo:isLocatedIn>, ?state)
```

[4] RDQL: http://jena.sourceforge.net/tutorial/RDQL/

```
(?state, <rdf:type>, <geo:Departamento>)
(?state, <geo:isLocatedIn>, ?country)
(?country, <rdf:type>, <geo:Country>)
(?country, <geo:name>, "Peru")
```

(Note that “?state” is a variable in RDQL and, hence need not be changed).

6. CONCLUSION

In this paper, we introduced the concept of an Ontology-based Object Catalog (OGOC) as the pivot of a federation of independent geo-object sources. The paper presented a generic architecture and a framework that facilitates generating customized catalogs for a specific application domain.

An OGOC stores a reference ontology and ontologies representing the conceptual schemas of the sources. The reference ontology may include a set of reference instances, which are sometimes essential to guarantee interoperability. In this context, the proposed concept of catalog is similar to a gazetteer, as in geographic applications.

We are currently specifying the framework. We intend to test the framework architecture by instantiating the hot-spots as follows:

- Reference ontology application domain: geographic application domain;
- Catalog access interface: Web services;
- Ontology repository: implemented using API Jena[5];
- Query language: RDQL;

ACKNOWLEDGMENTS

We gratefully acknowledge the helpful suggestions received from Nicolau Meisel. This work is partially supported by CAPES/Brazil and CNPq/Brazil grant 552068/2002-0 and 552040/2002-9.

REFERENCES

Atkinson, R., and Fitzke, J., 2002, Gazetteer Service Profile of the Web Feature Service Implementation Specification, *Version 0.9, OGC 02-076r3, OpenGIS© Discussion Paper, OpenGIS Consortium*; http://www.opengis.org/docs/02-076r3.pdf

[5] Jena API Framework: http://jena.sourceforge.net/

Berners-Lee, T., Hendler, J., and Lassila, O., 2001, The Semantic Web, *Scientific American*, May, 2001.

Chu, W. W., and Mao, W., 2000, CoSent: A Cooperative Sentinel for Intelligent Information Systems, *In: Proceedings of 2000 Int'l Conference on Artificial Intelligence* (ICAI), Las Vegas, NV.

Fayad, M. E., and Schmidt, D. C., 1999, Building Application Frameworks: Object-Oriented Foundations of Framework Design, Wiley Computer Publishing.

Florescu, D., Levy, A., and Mendelzon, A., 1998, Database Techniques for the World-Wide Web: A Survey, *SIGMOD Record*, 3(27): 59--74.

Guarino, N., 1998, Formal Ontology and Information Systems, *In: Proceedings of FOIS'98,* Trento, Italy, Amsterdam, IOS Press, 6-8 June, 1998, 3--15.

Gupta, A., Marciano, R., Zaslavsky, I., and Baru, C., 1999, Integrating GIS and Imagery through XML-Based Information Mediation**,** *NSF International Workshop on Integrated Spatial Databases: Digital Images and GIS;* http://www.npaci.edu/DICE/Pubs/isd99.pdf

Hakimpour, F., and Geppert, A., 2001, Resolving semantic heterogeneity in schema integration, In: Proceedings of the International Conference on Formal Ontology in Information Systems, Ogunquit, Maine, USA, October, ACM Press, 297--308.

Hemerly, A. S., Furtado, A. L., and Casanova, M. A., 1993, Towards Cooperativeness in Geographic Databases, *In: Proc. 4th Int'l. Conf. on Databases and Expert systems Applications* (DEXA 93), Prague.

Hernandez, M. A., Miller, R. J., and Haas, L. M., 2001, Clio: A Semi-Automatic Tool For Schema Mapping, *In: Proceedings ACM SIGMOD Conference 2001,* Santa Barbara, CA, USA; http://www.almaden.ibm.com/software/km/clio/sigmod02demo.pdf

Linthicum, D., 2004, Semantic Mapping, Ontologies, and XML Standards (August 20, 2004); http://*www.XML-JOURNAL.com*

Mena, E., Illarramendi, A., Kashyap, V., and Sheth, A., 2000, OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies, *In: International journal on Distributed And Parallel Databases* (DAPD), ISSN 0926-8782, 8(2):223-272.

Nebert, D., 2002, Catalog Services Specification, *Version 1.1.1, OpenGIS© Implementation Specification, OpenGIS Consortium;* http://www.opengis.org/docs/02-087r3.pdf

OpenGIS, 2004, OpenGIS Consortium Inc. (August 4, 2004); http://www.opengis.org/

Özsu, M. T., and Valduriez, P., 1999, Principles of distributed database systems, 2nd Edition, Prentice-Hall, Inc.

Reed, D., and Strongin, G., 2004, The Dataweb: An Introduction to XDI. *White paper of the OASIS XDI Technical Committee*, v.2.

Uschold, M., and Grüninger, M., 2001, Ontologies: Principles, methods and applications**.** *Knowledge Engineering Review*, 11(2): 93--155.

Vretanos, A. P., 2002, Web Feature Service Implementation Specification, *Version 1.0.0, OpenGIS© Implementation Specification, OpenGIS Consortium;* http://www.opengis.org/docs/02-058.pdf

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., and Hübner, S., 2001, Ontology-Based Integration of Information – A Survey of Existing Approaches, *In: Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing, Seattle, WA*, 108--117.