SOA Grid Mediation:

Service-Based Design for Heterogeneous Geospatial Information Data Interoperability

John Goodwin
Ordnance Survey
John.Goodwin@ordnancesurvey.co.uk

Xavier Lopez
Oracle
Xavier.Lopez@oracle.com

Jeffrey Pollock
Cerebra
Jeff.Pollock@cerebra.com

Abstract

Geospatial data interoperability has been a persistent and ongoing challenge for industry. Whether the scope of data sharing is within organizational boundaries, or between them, traditional efforts have been successful as point solutions only.

Recent progress in two fields of research has produced widely available and vendor supported software solutions that can be aligned to transform the way geographic information system (GIS) data is shared. First, Grid and Service Oriented Architecture (SOA) technology enables an easily accessible and reconfigurable message infrastructure. Second, semantic technologies enable a dynamic and context-driven view into data produced in different environments.

This paper describes the challenges that distributed GIS tools encounter when sharing data, which are in fact common in other industries. The authors demonstrate, with technical detail, how semantic technology standards such as RDF and OWL can help address those challenges. In keeping with the intended focus on semantic mediation of diverse data, detailed aspects of Web service, Grid, and SOA platforms are not discussed at length.

Table of Contents

INTRODUCTION **4**

BUSINESS DEMAND **4**

WHAT IS GEOGRAPHIC DATA? 5
WHAT IS GEOGRAPHIC INFORMATION? 5
HOW GEOGRAPHIC INFORMATION IS USED? 5
Land and Property *6*
Retail *6*
Utilities *6*
Environment *6*
Government *6*
Transportation *6*
Insurance *7*
Wireless and Location-Based Services *7*
SHARING LOCATION AND GEOSPATIAL DATA 7
XML *7*
GIS Spatial Data Semantic Interoperability Challenges *8*
Features *9*
Ontologies *10*

SEMANTIC GRID MEDIATION **10**

SERVICES ON THE GRID 10
CORE SEMANTIC TECHNOLOGY 11
WHAT IS DIFFERENT? 11
SO WHAT? 12

SCALE AND PERFORMANCE **12**

END-TO-END SCALABILITY AND MEDIATION FEATURES 12
LEVEL 2 - RDF AND SPATIAL DATABASE PLATFORM 13
Oracle Network Data Model *13*
Modeling RDF Data in Oracle *13*
Querying RDF Data in Oracle10g *14*
Example: Family Tree in RDF *14*
RDF Schema *15*
Creating Rule Indexes *16*
LEVEL 3 - OWL INFERENCE MEDIATION SERVICE 17
Performance Dial, with Variable Search Clarity *18*
SUMMARY CHARACTERISTICS AND METRICS 22

SOLUTION ARCHITECTURE **23**

GRID USE CASE VIEW 23
LOGICAL INFORMATION VIEW 23
INTERNAL FUNCTIONAL VIEW 24
FEDERATED DATA SECURITY 25
ONTOLOGY ENGINEERING METHODOLOGY 25

BUSINESS VALUE **28**

BENEFITS OF THE GIS GRID MEDIATION APPROACH 29
Data Management Benefits *29*
Organizational Benefits *30*
LOCATION ENABLING THE ENTERPRISE 30

OPERATIONAL GIS BUSINESS CASES **30**

CLASSIFICATION ALIGNMENT 31
ADAPTABLE DATA 31

SEMANTIC GIS QUERIES ON THE GRID 32
OTHER BUSINESS DOMAINS IN NEED 33
Life Sciences and Drug Discovery *33*
Product Master for Pricing and Configuration *33*
Metadata and Content Management *34*

CONCLUSION AND LESSONS LEARNED 34

APPENDIX A: 36

QUERYING RDF SCHEMA AND RULES IN ORACLE10G 36
Exercise 1. Select USStates bordering Colorado *36*
Exercise 2. Select USCities with 37" of annual rainfall. *37*
Exercise 3. RDFS inferencing on the us_territory model *37*
Exercise 4. Inferencing on us_territory model using a user-defined rulebase *38*
Exercise 5. Combine the results of an RDF query with data from a Spatial table *39*

Table of Figures

FIGURE 1: SAMPLE GEOGRAPHIC DATA IN GML/XML 8
FIGURE 2: INFORMATION FORMATS AND INTEROPERABILITY 11
FIGURE 3: RDF GRAPH IN ORACLE 10G 14
FIGURE 4: FAMILY TREE SCHEMA 14
FIGURE 5: INFERENCING WITH RDFS QUERIES IN ORACLE 10G 15
FIGURE 6: USING LITERAL CONSTRAINTS WITH RDFS QUERY 15
FIGURE 7: CREATING A RULES INDEX IN ORACLE 10G 16
FIGURE 8: EXECUTING RULES WITH RDFS IN ORACLE 10G 16
FIGURE 9: SUMMARY RDF AND SPATIAL REQUIREMENTS 17
FIGURE 10: BLOCK ARCHITECTURE OF HYBRID REASONING LAYERS AND XQUERY 19
FIGURE 11: XQUERY FUNCTIONS AT DIFFERENT REASONING FIDELITY 19
FIGURE 12: CREATING AN XQUERY USING OWL INFERENCE WITHIN CEREBRA 20
FIGURE 13: POTENTIAL GEOSPATIAL EXTENSIONS TO MEDIATOR XQUERY OPERATIONS 21
FIGURE 14: SUMMARY OWL MEDIATION REQUIREMENTS 21
FIGURE 15: SUMMARY GIS MEDIATOR FEATURES AND PERFORMANCE METRICS 22
FIGURE 16: SIMPLE CONCEPTUAL VIEW 23
FIGURE 17: GRID USE CASE VIEW 23
FIGURE 18: LOGICAL INFORMATION VIEW 24
FIGURE 19: INTERNAL MEDIATION ARCHITECTURE 25
FIGURE 20: POORLY MODELED GEOGRAPHIC CONCEPTS 26
FIGURE 21: A BETTER MODELED ONTOLOGY 27
FIGURE 22: TWO TAXONOMIC VIEWS OF THE WORLD 31
FIGURE 23: THE ALIGNED TAXONOMIES 31
FIGURE X: US TERRITORIES RDF MODEL 36

Introduction

Geographic data is a common component of up to 80% of all software applications. The breadth of reach for geographic data and geographic information systems is increasingly important for both business and government. Geographic data is central to most modern information technology systems: from disaster planning, utilities management, transportation, and insurance to more commerce-oriented applications such as location-based services and targeted marketing programs. However, to successfully share and reuse geographic data, it must be integrated not only syntactically, but also **semantically**.

The needs of geographic information systems, among others, have driven recent advances in data instance and model formats. Two new XML-based data languages driven by the W3C, the Resource Description Framework (RDF) and the Web Ontology Language (OWL), have proven their ability to resolve the semantic reconciliation challenges posed by geographic information systems. Aided by commercial, off-the-shelf tools from Oracle and Cerebra, this paper puts forth a vision for GIS data interoperability based on a Web services Grid infrastructure.

A familiar mediation architecture, essentially a hub and spoke system for data, is used in conjunction with powerful RDF capabilities from Oracle and OWL-based inferencing from Cerebra, to enable GIS data to be federated, shared, and interoperate with various consumer applications. New capabilities introduced with this approach include the use of highly interconnected instance data, sound and complete reasoning, as well as the mapping and retrieval of legacy data. Importantly, this paper directly addresses questions about performance and scalability for these newer data formats.

Many different business applications, with diverse technical examples, are discussed in this paper to illustrate the tangible business value offered by these semantic technologies.

At the conclusion of this paper, the reader will have a solid idea of how to build a GIS mediation platform and how it can be leveraged for new capabilities. Additionally, the reader will learn how Ordnance Survey, Oracle, and Cerebra have overcome technical challenges to develop and enhance the technology base for commercial use. While not a complete, installable application yet, the notion of a GIS Mediation Service has been demonstrated at multiple levels and shows much promise for future work.

Business Demand

It is a much quoted fact that 80% of all data has one common factor–geography. Data from small businesses, multinational entities, government agencies, local authorities, utilities, etc. will often have a geographic reference. Geographic information is fast becoming part of mainstream Information and Communications Technology (ICT). The more geographic data an organization holds the harder it becomes to manage and interpret. There is now a need for generic information systems (as well as specialized GIS) to be able to handle geographic information. An increasing number of companies and public services are using computer-based GIS and Web-enabled services which allow the rapid integration and analysis of information from many sources, effectively bringing maps to life in an interactive way.

What is Geographic Data?

The most basic type of geographic data offers a topographic representation of the real world. Topography is a graphic representation or precise description of the surface features of a region indicating their relative positions and elevation. These features include everything from forests, roads, utility networks, and rivers down to individual houses, garden plots, and even pillar boxes.

Geographic data models capture a simplified view of reality represented by four basic spatial entities. These four spatial entities are the point, line, area, and surface. For example, a point might represent a junction box; a line might represent a section of cable; an area might represent a building and a surface may represent the land surface through which cables are laid. Attributes can then be used to attach additional information to these spatial entities; for example, one might associate an address with a building.

In addition to this topographic mapping, new layers of information can be added to the data, such as: aerial photographic images which precisely match the mapping; data providing the addresses of all properties; and integrated transport information.

The most basic type of geographic data typically comes in two forms: raster data and vector data. Raster data is based on cells or pixels. Each cell is assigned a single value that represents an attribute for the area of that cell. For example, a cell might have a value of 42 which translates to a particular land cover type such as vegetation. Raster data is also used to store image data such as aerial photography or satellite imagery.

Vector data represents the world using a series of links and nodes. Links between two nodes can form edges, and a series of edges can form an area or polygon. Each link, node and area is given a unique identifier and the topological relationships between them are stored in a database.

What is Geographic Information?

Geospatial or geographic information is any information that can be associated with a specific location. This could include information about crime trends throughout a given neighborhood, the population of a town, or data about the occupants of a particular address. This location would typically be specified by a coordinate (for example latitude and longitude, or a National Grid coordinate), a postcode, or a street address. The element within the data that identifies the location of an object is known as a *geocode*. Typically geospatial data will contain metadata indicating when the information was collected and last updated, what scale the data is in, and how accurate the location information is.

How Geographic Information is Used?

As with most types of information, geographic information is used to support decision making and policy. Geographic information supports decision making where location needs to be taken into account, allowing decision makers to answer questions such as “where are?”, “what is the nearest?”, etc. This section outlines some of the basic uses of geographic information at a fairly high level.

Land and Property

Look beneath the surface of any land or property development and its geology, history, and local surroundings become just as important as its geographical location.

Geographic data is essential to electronically integrating many different sets of information relating to a specific site. Combining third party information with our location data, a user may search, analyze, and visualize their development from the planning stage through to completion and beyond.

Retail

Integrating socio-economic data about potential customers with geographic information allows retailers to find locations suitable for the development of new stores. Additional factors such as drive time to the store's location and vicinity of competing stores can also be taken into account. All of this helps a business to obtain a definitive picture of who their customers are, where they live, what they want, and when they want it. Fully understanding the context of a customer base facilitates more intelligent marketing activity, vital for enabling businesses to increase revenue and profits.

Utilities

Data from utility companies integrated with geographic data allows utilities to manage plat organization such as the location of cables, valves, and so on in their networks. This requires up-to-the-minute, highly accurate, large-scale geographic data.

Environment

At the simplest level, geographic information can be used to help with the modeling of simple environmental processes such as soil erosion or the response of a river basin to a large amount of rainfall.

Geographic information can also be used in all phases of the risk and disaster management process, from prevention to immediate reaction stages. This typically requires the integration of geographic data, meteorological data, various types of environment data, and demographic information about people in risk areas.

Government

Effective policy and planning is dependent on the ability to analyze disparate sets of information. It is vital that the thousands of datasets available from government departments and third party organizations can be effectively "joined-up" to provide a framework for the delivery of e-government.

Geographic information can be used for land search and planning, property and highway management, and emergency services monitoring.

Transportation

Geographic information plays a vital role in the transport sector including road distribution, rail, water transport, automotive, air, and public transport systems. Geographic intelligence gives transport organizations a detailed view of their world. It can enable effective and efficient routing, performance analysis, tracking, supply chain management, and in-vehicle navigation systems.

Insurance

Geographic information is increasingly being used as a profiling tool for risk assessment and insurance purposes, identifying areas of highest and lowest risk. This often requires integrating data from sources as diverse as geology, weather, and property values.

Wireless and Location-Based Services

Location Based Services (LBS) is a technology fuelled by introducing geographic information into the wireless arena. Large scale detailed geographic data can greatly enhance this technology by adding valuable location context to information. This enables the development of services relevant and personal to a user's physical location. LBS assist in personal and commercial route planning, vehicle, asset and personnel tracking, information services, real-world computer gaming, and emergency and breakdown services. Increasingly, large companies are planning to use LBS approaches to deliver highly targeted marketing information to consumers.

Sharing Location and Geospatial Data

Users increasingly need to go beyond the capabilities offered by paper and digital maps. The technology need is for users to integrate their own enterprise and application data with well structured and intelligent geographic information. Currently the costs of data conflation and adaptation are a major barrier to the adoption and efficient exploitation of complex datasets. The data is complex, and often needs to be integrated and adapted to the user's end tasks and IT infrastructure. The key to solving these problems is fully interoperable geographic information.

XML

Significant work is being done by standards bodies, most notably by the OGC (Open Geospatial Consortium), into standard syntactic and structural representations of geographic information. Non-proprietary data standards such as XML provide the first step towards interoperability. In particular, the Geographic Markup Language (GML), provides an application-neutral way to exchange geographic information. Figure 1 shows a sample of GML.

```
<osgb:topographicMember>
<osgb:TopographicLine fid='osgb1000000153442890'>
<osgb:featureCode>10019</osgb:featureCode>
<osgb:theme>Buildings</osgb:theme>
<osgb:theme>Land</osgb:theme>
<osgb:descriptiveGroup>Building</osgb:descriptiveGroup>
<osgb:descriptiveTerm>Outline</osgb:descriptiveTerm>
<osgb:make>Manmade</osgb:make>
<osgb:physicalPresence>Obstructing</osgb:physicalPresence>
<osgb:polyline>
<gml:LineString srsName='osgb:BNG'>
<gml:coordinates>
        461154.890,152807.220 461163.000,152811.300 461165.550,152806.250
        461164.300,152803.800 461161.660,152802.390 </gml:coordinates>
</gml:LineString>
</osgb:polyline>
</osgb:TopographicLine>
</osgb:topographicMember>
```

Figure 1: Sample Geographic Data in GML/XML

Data provided in GML/XML is intended to be human- and machine-readable, but it is not human- and machine-**understandable**. ISO standards and schemas provide a way to ensure data is structured correctly but they do not capture knowledge about the data. Understanding both the actual content of the data, and how it can be used with other data sources remains expensive. ***To reuse and share geo-data successfully, integration has to be realized not just on a syntactic level but also on a semantic level.***

GIS Spatial Data Semantic Interoperability Challenges

A while ago there was a project undertaken by the UK government to estimate the number of people living in a particular area, which would then determine the amount of tax grants local authorities receive. A number of different datasets had to be integrated to complete the calculation. Problems arose when it was assumed that different data providers meant the same thing by "multiple occupancy address". For one data provider "multiple occupancy address" implied multiple post boxes on a given property and vice versa. However, from the other data provider's point of view a property with just one post box (for example a block of flats, shared house, or old people home) could still be multiple occupancy. This semantic mismatch between data providers contributed towards the male population of the UK being underestimated by approximately one million.

When exchanging data it is important that a recipient understands what a supplier is sending, how it differs from what is needed, and how it can be translated to meet the end user requirements (or what the impacts are if the end translation is imperfect)[3].

The reconciliation of different world views is a major barrier to data interchange. This becomes more complicated as more data providers and datasets are involved. There are numerous semantic issues concerned with the nature of the data provided, for example:

- Does the data provider mean "building" when using "roofed structure?"
- Is the meaning of "flood plain" intended by two different data providers the same and is it the same as that understood by the end user?
- The user needs information on "flood defenses", how is data captured against a given specification re-purposed to provide the desired information?
- Is data collected for one purpose useful for another application?
- Two different datasets are classified at different levels of semantic granularity, how can this be resolved?
- Is the meaning and structure of an "address" the same across all the datasets?
- Does the data provided conform to a particular specification?

At present, resolving these kinds of issues is a largely manual task, and often results in an imperfect solution. Typically solutions are hard-coded in applications with developers making assumptions based on ambiguous or implicit information. Furthermore, it seems unlikely that true harmonization will ever be practical between multiple data suppliers. When exchanging information with customers or partners we could assume that everyone is talking about the same thing given a particular term, word, or symbol but history tells us this is unlikely to happen. Efficient sharing of data in a geographically distributed environment with multiple world views

requires formal descriptions of those views in a way that a machine can understand using an agreed structured language. An ontology-driven system provides a solution.

An ontology is a formal way of representing knowledge about concepts and the relationships between those concepts in a particular domain and from a certain point of view. Ontologies allow machines to "understand" what is represented in data, and through machine reasoning, deduce new knowledge about what is represented and create links between different world views. Additionally, abstracting domain and business knowledge to the ontology layer avoids extensive reliance on custom procedural programming and the need to rewrite legacy code every time a data model changes or policies are updated.

Features

Traditionally the basic unit of geographic information within most data models is the "feature". The OGC defines a feature as:

> "A feature is an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth. Vector data consists of geometric and topological primitives used, separately or in combination, to construct objects that express the spatial characteristics of geographic features."

Features can include a wide range of phenomena that can be located in time and space such as buildings, water bodies, towns and villages. Typically a feature encapsulates all that is known, in a given domain, about a specific geographic phenomenon in one entity.

Features can be seen at two levels: feature instances and feature types. Feature instances are the discrete representations of geographic phenomena in a database. Instances can be grouped into common classes that share common characteristics; these are known as the feature types. It is these feature types that we need to encode as concepts in our ontology. In OGC terms feature typing need not be rigid, but can be application-specific. For example one view or application may type a given feature instance as a house, while another types it as a building.

Different organizations tend to use different discrete sets of information about geographic entities and a given feature instance can encapsulate different human concepts in different contexts. For example, consider how many ways one could view a "Pub" and consider the different types of information that could be held about it. A Pub could be viewed as a residence (if this is where the landlord lives), a premises, and of course a type of "eating and drinking establishment". Viewing the pub as premises we might be interested in its address or its geographic extent, and how the premises relate to other objects such as the buildings and beers gardens it is comprised of. Viewing the pub as a type of "eating and drinking establishment" we would be interested in its open hours and the range of beers and wines it serves. Typically this information will be provided by different data suppliers. It can be seen that "features" are not really the atomic units of geographic information exchange, but rather they are aggregates of smaller discrete sets of information.

It is clear that in order to enable the exchange of feature-based geographic information a model is required that allows the comparison and exchange of information at a finer level of detail than that of the feature. An ontology can be used to provide such a model.

Ontologies

A geo-ontology would describe concepts in the following terms:

- A natural language description will be given of the concept in well-structured English
- Physically measurable properties including, for example, geometry
- Relationships to other properties including:
    - Mereology (part-whole relationships, e.g. apartments part of buildings)
    - Topology (configuration of entities, e.g. surface relief and position of features)
    - Taxonomy (specialization and generalization along taxonomies)

Examples of concepts include geographic form (the physical structure of geographic objects, e.g. Building, River, Premises), function (the use of a geographic object e.g. Education, Retail, Industry etc.), addresses, geometries, etc. These concepts can then be combined to define and describe the real world objects that are digitally abstracted as features.

If different data providers capture their knowledge in the form of ontologies, links can be made between the ontologies to align different views of the world. Sharing semantic information using an ontology enables integration and reuse of knowledge and data across various applications such as "joined-up" government, the commercial sector, inter-disciplinary scientific research, or any consumer of GIS data. Today this sort of interoperability is possible across disparate relational databases only "inside the firewall".

Demands for "data on tap" are driving the need for Service Oriented Architectures which allow data to be accessed via Web services. There will be a need to integrate data from geographically disparate sources, some of which is streamed from a Web service and some of which is in a relational database. Recent innovations and new technology standards now allow a new generation of techniques for simplifying how geospatial information can be shared. These advances in technology will be discussed in the follow section.

Semantic Grid Mediation

A recent confluence of two technology bases has led now to an opportunity to address these persistent geographic interoperability challenges with a new kind of solution that is smarter and more change resilient than anything that has come before. Service Oriented Architectures, and the utilities that support them and enable a service Grid, have achieved a maturity level suitable for large production environments. Similarly, the Semantic Web technology-base led by the W3C's RDF and OWL standards can bring radical, new capabilities to federated data sets. When taken together, the resulting system can achieve unprecedented levels of messaging and data configurability, openness, and security in large systems.

Services on the Grid

Grid computing is a new IT architecture that produces more resilient and lower-cost enterprise information systems. With grid computing, groups of independent, modular hardware and software components can be connected and rejoined on demand to meet the changing needs of businesses.[4] Technically speaking, the foundation of Grid computing lies in open and standards-based Web services. The basis of a loosely-coupled message environment for one software system to communicate to another relies on the foundation of a message protocol, a service contract/API, and a directory system for listing services. These are accommodated by the foundational

standards of SOAP, WSDL and UDDI. Collections of application Web services become a true Grid when the deployment of management tools and common service utilities accompany the application services. Thus, an application or infrastructure Grid accounts for the linking of applications like GIS, but not for the reconciliation of data formats and meaning.

Core Semantic Technology

Traditionally, data communicated in any environment, including a Web services Grid, must be reconciled in a labor intensive coding-centric way. In some cases, there are proprietary vendor solutions that can simplify this process, but many businesses find these unattractive because they are expensive and tend to lock-in the data–and the business–to a particular way of working.

Semantic Web technologies have been standardized by the W3C towards a long-term vision for the Internet; intending to make it easier to find and protect data on the Web. But these core W3C technologies are also highly desirable for organizational and community-oriented data exchanges as well. Unlike XML or relational data, RDF and OWL specialize in linking (see Figure 2) data and capturing the intended relationships in a well-defined way. This makes it easier to reuse and disseminate domain knowledge to dispersed communities.

What is Different?

RDF is a graph-based data language. Directed graphs assume very little about the data they contain, and can evolve more easily than relational or hierarchical data storage mechanisms. Composed of triples, an atomic data unit in RDF consists of a subject, an object, and a predicate. Essentially, the predicate is a named relationship that links one piece of information to another. RDF systems can represent any data that is traditionally contained within a tabular (relational) or hierarchical (XML) data management system, but have the advantage of being capable of evolving without disruptive data migration issues. This type of data management system is superior when data needs to be highly interconnected.[5]

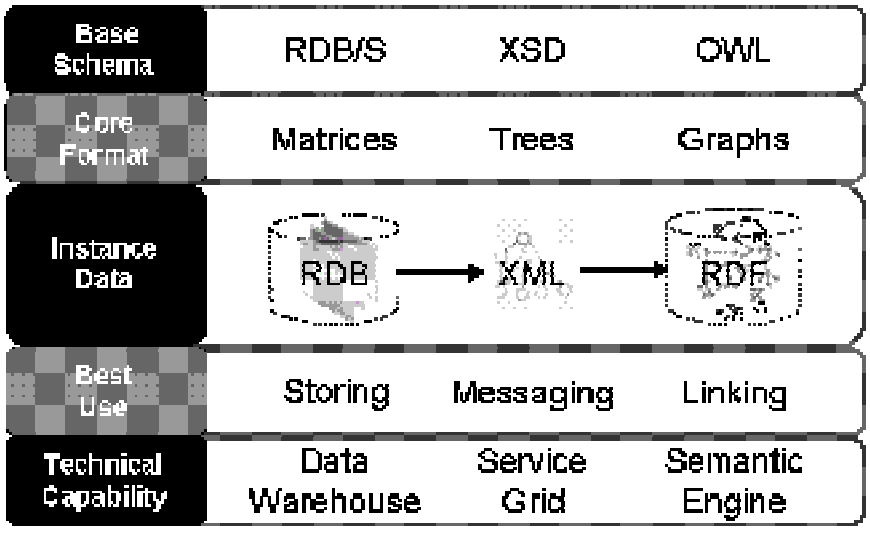


Figure 2: Information Formats and Interoperability

OWL is an unambiguous and expressive schema language for modeling domain information. It uses RDF as a core data format, but constrains the recognized data relationships to a set of named properties that allows a reasoner[6] to derive implicit facts by combining explicitly stated facts with constraints expressed in the schema. In effect, the OWL specification allows IT practitioners to

create metadata that can describe and define data across many different systems, while also allowing an engine to find inferences implied by any given schema, data, or constraints.

But RDF and OWL also provide other key advantages for distributed GIS environments. Both RDF and OWL are specified in XML, which means that existing infrastructure supporting XML can be leveraged toward deploying these newer formats. Reliance on XML also means that developers who are familiar with XML will be capable of learning OWL and RDF without having to learn a new basic syntax. The W3C's primary innovation in these new data formats is the inclusion of the URI as the basic linking system between data sets. For distributed GIS environments, this is a crucial capability because it ensures a deep and fundamental ability to federate information. In practice, communities can join, share, and distribute references to other community data simply by importing other ontologies, referring to external triples, and other extensions, even allowing for the partial inclusion of ontology subsets–enabling communities to use common metadata dictionaries without having to adopt a comprehensive shared data model.

So What?

As it turns out, RDF and OWL address the largest of all business needs: the high cost and inefficiency of sharing enterprise business data. Without a more clever way to link and categorize data from different systems, business and government will always be faced with enormous capital expenditures and debilitating lead-times for gaining access to time-sensitive business information.

In the context of geospatial information, this can have adverse affects on everything from public safety, critical infrastructure management, to mobile computing convenience. Semantic Web technology was designed to address these very challenges.

Scale and Performance

The well-informed software architect will no doubt first question the scalability and performance of a semantics-based solution. It is known that RDF databases could not scale to the level of relational systems, and that the overhead of the algorithms for inference have restricted the quantity of instance data that is tractable for OWL reasoners. Thus, the authors present a stepped approach toward scalability and discuss the associated metrics and limits for each level.

In order to face the volumes of data present in the real-world GIS environments, the design solution must be able to leverage enterprise class software and hardware infrastructure. It must also be capable of interfacing with and searching terabytes of data in a near-real-time (NRT) manner. For the purposes of this paper, NRT is used to mean an unconstrained search time limit that typically responds in single-digit seconds, excluding network latency.

End-to-End Scalability and Mediation Features

In general, it is a fundamental software truth that the more intelligence applied when searching data, the more CPU overhead will be required. For example, a Google search is very simplistic in its word-matching algorithms and can be applied to massive amounts of indexed data. In contrast, a search using an inference system is algorithmically complex for data relationships and usually operates on a much smaller corpus of data. There are many levels of sophistication and search complexity in between.

The end-to-end architecture best suited for a distributed GIS sharing environment has at least three levels:

Level 1 – Billions of Records: Relational and hierarchical data
Level 2 – 10's of Millions of Records: Graph and spatial data
Level 3 – Millions of Records: Ontology and graph data

Interestingly, the Level 3 ontology data can be schematic in nature or instance data. Thus, a GIS mediator service could use inference for categorizing instances in Level 1 & 2 systems for a schema-driven retrieval, and also for providing inference on the physical instance data sourced from Levels 1 & 2, which has previously been loaded into an inference engine. Either method enables valuable use cases that reach far beyond current GIS data sharing capabilities.

The remainder of this section will identify functional requirements and expected performance for Level 2 and Level 3 data ecosystems.

Level 2 - RDF and Spatial Database Platform

Applications developers are now turning to relational database technology for managing the large amount of RDF data generated to support semantic applications. For example, Oracle Database 10g now makes it possible to store, index, and query graph (triples) data using standard object/relational technology and Standard Query Language (SQL). This is a first for the industry. Application developers can now benefit from an open, scalable, secure, and reliable data management platform for RDF-based applications. Based on the underlying Network Data Model, triples data can be persisted, indexed, and queried, similar to other object-relational data types. The Oracle10g database ensures that application developers benefit from the scalability of the Oracle database to deploy scalable semantic-based enterprise applications.

Oracle Network Data Model

A network is a graph representation for modeling objects of interest and their relationships. The Network Data Model (NDM) feature enables graph modeling and analysis in Oracle Database 10g. NDM explicitly stores and maintains connectivity (nodes, links, and paths) for a variety of network structures (such as hierarchical, directed, undirected, random, and scale-free).

It provides network analysis functions including connectivity analysis, shortest path, all paths, accessibility analysis, within cost (distance), nearest neighbor, and minimum cost spanning tree. NDM also provides a framework for applying network constraints and rules (e.g. path length, cost, minimum bounding rectangle, or any application-related constraint, such as links or nodes to avoid). As packaged with Oracle Spatial, additional geometry operations and analysis functions are also included to support geospatial and semantic web applications

Modeling RDF Data in Oracle

The RDF store in Oracle is built on top of the Oracle Spatial NDM, in which a network or graph captures relationships between objects using connectivity. RDF graphs are modeled as a directed logical network. Using a graph, the subjects and objects of triples are mapped to nodes, and predicates (or properties) are mapped to links that have subject start-nodes and object end-nodes. A link, therefore, represents a complete RDF triple. A key feature of RDF storage in Oracle is that nodes are stored only once, regardless of the number of times they participate in triples, but a new link is always created whenever a new triple is inserted. Figure 3 illustrates how an RDF

graph of three RDF triples: <S1, P1, O1>; <S1, P2, O2>; <S2, P2, O2> is modeled. When a triple is deleted from the database, the corresponding link is removed. However, the nodes attached to this link are not removed if there are other links connected to them.

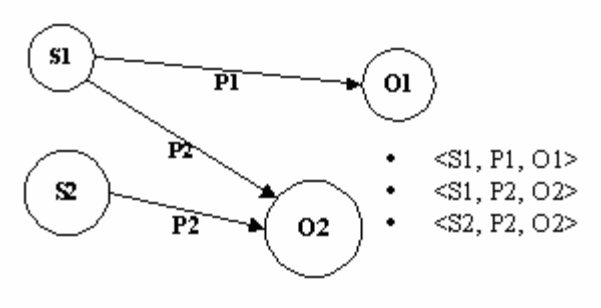


Figure 3: RDF Graph in Oracle 10g

In Oracle, RDF triples are parsed and stored in global tables under a central schema. All the triples for all the RDF graphs (or models) in the database are stored under this schema. Only references (IDs) to these triples are stored in the user-defined application tables that may contain other attributes related to the triples; functions are provided to retrieve the actual triples when necessary.

Storing RDF data in NDM allows use of NDM's sound mathematical model and many graph analysis algorithms applicable to RDF data and subgraph results of SQL-based RDF queries.

Querying RDF Data in Oracle10g

SQL-based querying of RDF data allows arbitrary graph pattern matching to retrieve one or more subgraphs and optionally further process the retrieved data by combining (e.g., using join) with traditional database table data. RDF Schema (RDFS) rules-based inferencing support allows automatic derivation of entailed graphs against which SQL-based RDF graph queries are executed. Storage and SQL-based RDF graph query supports reification (treating RDF statements as resources so that statements can be asserted about them), heterogeneous data types (object column values may be of different data types) and multiple representations of same scalar values (e.g., 10, 010, 0010 all represent the same scalar value 10). Appendix A introduces five simple exercises demonstrating inferencing with RDF/S queries using Oracle10g.

Example: Family Tree in RDF

This section presents a simple example of an RDF model for statements about a family tree (genealogy). The family relationships in this example reflect the tree shown in Figure 4. This figure also shows some of the information directly stated in the example: Cathy is the sister of Jack, both Jack and Tom are male, and Cindy is female.

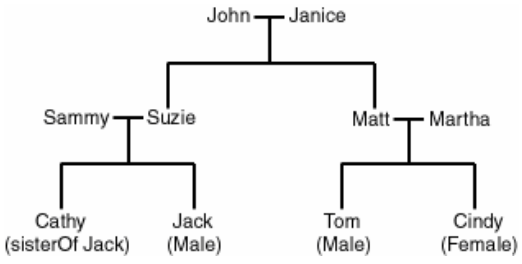


Figure 4: Family Tree Schema

The rows inserted into the FAMILY_RDF_DATA table express the following information: John and Janice have two children, Suzie and Matt. Matt married Martha, and they have two children: Tom (male, height 5.75) and Cindy (female, height 06.00). Suzie married Sammy, and they have two children: Cathy (height 5.8) and Jack (male, height 6). After the data is inserted into the FAMILY_RDF_DATA table, inferencing queries can be executed on this data as shown in the following examples.

```
-- Select all males from the family model
SELECT m MALES
 FROM TABLE(SDO_RDF_MATCH(
  '(?m rdf:type :Male)',
  SDO_RDF_Models('family'), null,
SDO_RDF_Aliases(SDO_RDF_Alias('','http://www.example.org/family/')), null));

MALES
--------------------------------------------------------------------------------
-http://www.example.org/family/Jack
http://www.example.org/family/Tom
```

Figure 5: Inferencing with RDFS queries in Oracle 10g

Oracle's RDF data store allows for the storage of multiple representations of literal values. For example, using SQL insert statements, a developer may make one or more assertions about the heights of Jack, Tom, and Cindy. Jack and Cindy are both 6 feet tall, but the values are entered differently so that lexically they look different. When this value is stored in the RDF store, a generic representation is used to treat them both as equivalent values, but at the same time, the original value is also preserved. The following query displayed in Figure 6 shows how to get both Jack and Cindy with a single query that asks about height. This example also shows how to use literal constants in the RDF query.

```
SELECT p Person, h Height
 FROM TABLE(SDO_RDF_MATCH(
 '(?p :height "06.00"^^xsd:decimal) (?p :height ?h)',
 SDO_RDF_Models('family'), null,
 SDO_RDF_Aliases(SDO_RDF_Alias('','http://www.example.org/family/')), null));

PERSON
--------------------------------------------------------------------------------
HEIGHT
--------------------------------------------------------------------------------
http://www.example.org/family/Jack - 6
http://www.example.org/family/Cindy - 06.00
```

Figure 6: Using literal constraints with RDFS query

RDF Schema

RDF does not make assumptions about any particular application domain, nor does it define the semantics of any domain. For example, in this family model, there is no domain-specific constraint that only males can be the objects of "fatherOf" relationship. RDF Schema lets users

define the domain for an application. This information is represented using the concepts of classes, properties, and attributes.

In the family example, Person is a class that has two subslasses: Male and Female. And parentOf is a property that has two subproperties: fatherOf and motherOf. And siblingOf is a property that has two subproperties: brotherOf and sisterOf. The properties fatherOf and brotherOf have the domain Male, and motherOf and sisterOf have the domain Female. Any new triples that are inserted into the FAMILY RDF model will take advantage of the RDFS constraints.

Creating Rule Indexes

Inferencing constraints can be used in answering RDF queries by enabling the RDFS rules for a given set of RDF models. A rule index is created for a set of RDF models and at query time, users can optionally specify the use of RDFS rules in answering RDF queries. The rules index will pre-generate (forward chain) all the triples which can be inferred using the RDFS rules. This pre-generation process is done by creating a rules index as shown in this example:

```
-- RDFS inferencing in the family model
BEGIN
 SDO_RDF_INFERENCE.CREATE_RULES_INDEX(
  'rdfs_rix_family',
  SDO_RDF_Models('family'), SDO_RDF_Rulebases('RDFS'));
END;
/
```

Figure 7: Creating a rules index in Oracle 10g

Once this rules index is created, the inference query to find all the males from the family model can be executed with the option of using the RDFS inferencing in answering the following query.

```
-- Select all males from the family model, with RDFS inferencing.
SELECT m MALES
 FROM TABLE(SDO_RDF_MATCH(
  '(?m rdf:type :Male)',
  SDO_RDF_Models('family'), SDO_RDF_Rulebases('RDFS'),
SDO_RDF_Aliases(SDO_RDF_Alias('','http://www.example.org/family/')),
  null));

MALES
--------------------------------------------------------------------
http://www.example.org/family/John
http://www.example.org/family/Matt
http://www.example.org/family/Sammy
http://www.example.org/family/Jack
http://www.example.org/family/Tom
```

Figure 8: Executing Rules with RDFS in Oracle 10g

In addition to the base RDFS rules, users can create their own set of rule bases and use them in inferencing queries. For example, for the family RDF model, users can define a "grandParentOf" rule to define the grand parent property. A rules index is then created to enable this rule for use in inferencing queries.

Capabilities	Description	Metrics/Technical

Capabilities	Description	Metrics/Technical
RDF Data Management	Manage GIS graph data over full lifecycle	
Query	Runtime query listening over federated sources	SQL + RDF_MATCH
Inference	Set notifications based on service events	RDF subsumption
Security	Standard RDBMS security management	RDB ACL
Clustering	Standard database clustering	Oracle
Partitioning	Standard database partitioning	Oracle
Loading	Data loading and transformation into RDF	Oracle
Scale and Performance	Attributes of GIS data population	
Inference (Subsumption)	SubClassOf and SubPropertyOf axiom support	~30,000,000 instance @ 1s.
Storage	Size object population in the RDF store	10's of Terabytes
Spatial NDM	Network Data Model, core Spatial features	
Modeling	Hierarchical and constraint-based modeling	
Analysis	Transitive closure and path computation	
Constraints	Rules (network constraints) and cost-based queries	
Visualization	Logical views of graph data	

Figure 9: Summary RDF and Spatial Requirements

An important benefit of using a RDF data model in standard DBMS is dramatic improvement in server performance, reliability, scalability, and security, these benefits are clearly identifiable in Figure 9. RDF data-support is simply an extension of a well-defined evolution by mainstream IT vendors to manage abstract data types (spatial, imagery, video, audio, time series, text) in standard commercial database technology. Likewise, application developers can benefit from a wide range of supported tools and application servers supporting this enhanced database platform.

Level 3 - OWL Inference Mediation Service

Applications developers are now beginning to deploy model-driven applications using off-the-shelf and standards-based tools. For example, Cerebra Server makes it possible to load, manipulate, infer, and query logical (ontology) data using standard XML technology, including XQuery. This is a first for the industry. Application developers can now benefit from an open, scalable, secure, and reliable mediation platform for OWL-based applications. Natively based on OWL, models can be persisted, indexed, and queried, similar to any XML data type. The Cerebra Server platform ensures that application developers benefit from the flexibility inherent in inference-driven tools.

The OWL Inference Mediation Service (Mediation Service) is, at a business level, a way to interpret information correctly. Just as interpreters broker discussions at a United Nations conference table, the Mediation Service brokers data formats and structure wherever it needs to be localized into a system.

For GIS the notion of a Mediation Service is not new[7]. The reasons that GIS needs to mediate between different interpretations of data should be clear: different consumers of geographic data use different notions of what specific features are understood to be. The classic, albeit simplistic, example of this is in the distinction between a "Lake" and a "Pond." Depending on how a modeler chooses to classify a "Body of Water," a single feature could be either a "Lake" or "Pond." Thus, the act of communicating GIS information often embeds the *folksonomy* (a term used to describe a local taxonomy of information that is valid within a fairly narrow context) in

the physical data exchange. Overcoming these localized vocabularies has historically required specialized application code to transform data, which is not only expensive, but often so time-consuming as to make the process impractical for large communities. GIS owners have the disadvantage of belonging to a global community with many small, specialized dialects that result in particularly difficult data exchange scenarios.

In practice, the Mediation Service is a Grid-enabled service that natively speaks XML in both query and response. The core capability of the Mediation Service is to provide the interpretation necessary to provide accurate information and disambiguate queries to federated information. For example, asking for the set of all "Lakes," the Mediation Service would be capable of understanding that, in a local vocabulary, a "Lake" is a "Body of Water" that has a specific volume, depth, or dimension. It would retrieve the appropriate data regardless of whether the original modeler deemed some features to be "Ponds" or "Lakes". Technically speaking, the Mediation Service would be capable of doing this by using OWL Restriction Classes and inference during a query.

Inference systems come in a variety of flavors, each of which is best-suited for different things. The core inference capability is enabled by the W3C standard OWL Description Logics (OWL DL) approach, which is a reliable and consistent approach to inferencing on data. The reliability and consistency of standard OWL DL is an essential requirement because without it different communities might interpret information differently, defeating the purpose of a Mediation Service in the first place.

Performance Dial, with Variable Search Clarity

As previously mentioned, OWL inference systems can typically operate on millions of data points at one time. They are actually capable of much more than this, but the time required for the algorithms to complete makes very large data sets (over a million instances) impractical in all but a few cases. An essential performance feature for a Mediation Service inference system is the ability to adjust the fidelity of the information. Put another way, the ability to dial-in the specific inferencing capabilities has a direct impact on the performance and the clarity of the information that has been queried. Usually, a system that has been designed to provide very deep levels of inferencing (supporting the complete OWL DL specification for example) will result in slower overall query times. The inverse is true for less intense inference levels – performance on query time may improve dramatically.

A hybrid reasoning capability (Figure 10) enabling adjustable search fidelity is a natural feature for the Grid Mediator. Users and applications can determine the tradeoffs for certain search types, or even analyze the differences between result sets for analytic reports. A Grid Mediator, as conceived by this approach would span the range of inference from explicit (SQL/XML) to lightweight (OWL Tiny, OWL Lite) to more heavyweight (OWL DL) and, as the technology evolves, to generally undecideable forms of inference (OWL Full, FOL).

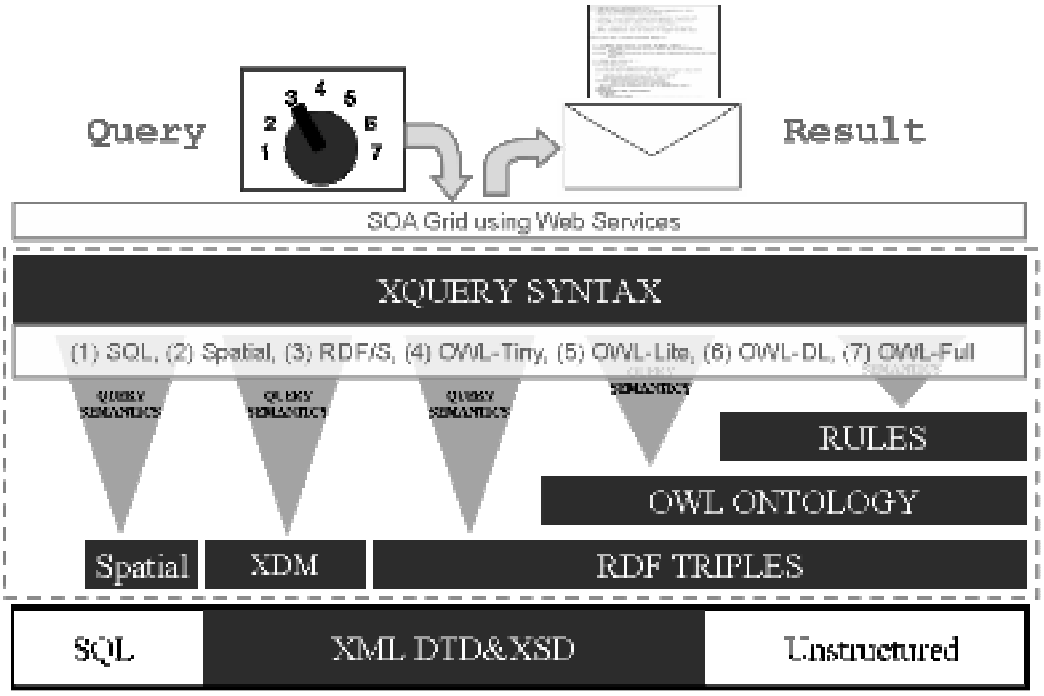


Figure 10: Block Architecture of Hybrid Reasoning Layers and XQuery

Querying information with a hybrid facility would be a user-selectable (figure 11) capability, which could be changed for different queries on the same data, or specific query fidelity on specific data. The approach pioneered by Cerebra is to leverage XQuery functions to change fidelity on an ad hoc basis and apply changes to properties files that control the execution of core Tableaux reasoning algorithms. To date, the implementation is basic, but functional.

It is important to note, that in a mathematical sense, there is really only a binary option: to choose a query approach either with or without correctness guarantees. In all cases, it is possible to select algorithms that have known consequences to query retrieval time, but once in the non-guaranteed nferencing domain, there is no way of telling "how incorrect" the answers may be. This provides yet another point of emphasis on why a hybrid approach, grounded in correct OWL reasoning, with user selectable query levels, is desirable for application developers.

XQuery Hybrid Reasoning Function 1:

```
  member($C, "direct|all", "DL|HYBRID|RDFS")

Returns: member instances of a specific class description $C, "all"
returns all members of $C, depending upon reasoning mode, including
complex expressions and sub class membership. Intuitively, "direct"
returns only instances that are directly members of $C.
```

XQuery Hybrid Reasoning Function 2:

```
  related($C, $P, $D, "DL|HYBRID|RDFS")

Returns: instances matching a combination of classes ($C), individuals,
literals, datatypes ($P) and RDF properties ($P).
```

Figure 11: XQuery Functions at Different Reasoning Fidelity

As a query language, XQuery is quite similar to other languages, but has the distinct advantage of allowing for extendible functions (Figure 12) and native XML document handling. The approach used for Mediation Service implementations to date has been to forego the inclusion of any specific XPath or XQuery Data Model support, and to instead optimize the handler for OWL- and RDF-based data formats. However the grammar, functions, and XML result-set handling are largely intact–even in early implementations.

```
declare default function namespace
       "http://www.cerebra.com/cerebra/functions/3.1"
declare default element namespace
       "http://localhost:8080/sanitized/geo-test.owl"
for $i in instances()
for $j in instances()
for $p in instances()
for $cn in datavalues()
where every $model in models() satisfies
 member($i, "Feature", $model)
 and member($j, "Obstruction", $model)
 and related($j, "hasWeight", $i, $model)
 and related($j, "hasConceptSources", $p, $model)
 and related($p, "movable", $cn, $model)
 and matches($cn, "xs:string", "C01_ConceptName")
      return <osgb:topographicMember><osgb:TOID>{$i}</ogsb:TOID>

FLOOD DETERENTS
-----------------------------------
<osgb:topographicMember><osgb:TOID>00213</ogsb:TOID>
```

Figure 12: Creating an XQuery using OWL inference within Cerebra

The power of the hybrid XQuery approach really shines when we start to envision further extensions to the hybrid capabilities, to take advantage of spatial reasoning operations for example. In the same way that we have demonstrated how a single query layer can simultaneously leverage direct RDF retrieval and OWL DL inference, we envision cases where additional operators can be inserted to directly query a spatial system (Figure 13). The combined power of multiple inference types on single queries can have numerous advantages to business and developers who need answers to complex questions, simply and reliably.

```
declare default function namespace
      "http://www.cerebra.com/cerebra/functions/4.1"
declare default element namespace
      "http://localhost:8080/sanitized/geo-test.owl"
for $i in instances()
for $j in instances()
for $p in instances()
for $cn in datavalues()
where every $model in models() satisfies
 member($i, "EmergencyAccommodation", $model)
 and member($j, "PrimaryCare", $model)
 and outsideRegion($p, "FloodZone", $model)
 and insideRegion($p, within("miles",20,"Floodzone"), $model) return
<EmergencyCare><Primary>{$i}</Primary></EmergencyCare>

EMERGENCY ACCOMMODATION
-----------------------------------
<EmergencyCare><Primary>Hayfield High School</Primary></EmergencyCare>
<EmergencyCare><Primary>Anytown Local Church</Primary></EmergencyCare>
<EmergencyCare><Primary>Town Center Clinic</Primary></EmergencyCare>
```

Figure 13: Potential Geospatial Extensions to Mediator XQuery Operations

Despite the heavy emphasis thus far in explaining the core inference and query capabilities, there are other important features for a GIS Mediation Service. For instance, the registration and discovery of available geographic data can be accomplished internal to a Mediation Service using an ontology (having the benefit of inference), or externally using a service registry such as UDDI or ebXML (no inference for discovery). A Spatial Assembly Capability[8] can assemble results into a common XML document for consumption by the requesting service. Finally, a publish-and-subscribe-based event broker provides for system-wide notifications to be sent out to subscribers about events internal to the Mediation Service – such as new data becoming available.

Capabilities	Description	Metrics/Technical
Mediation Features	Reconcile different GIS semantics on the information	
Query Broker	Runtime query listening over federated sources	XQuery
Pub/Sub Events	Set notifications based on service events	JMS via SOAP/WSDL
Publishing	Assemble heterogeneous data into consumable sets	XML content assembly
Registry/Discovery	Discover what services are available via descriptions	OWL Directories
Security	Multi-level role and user based restrictions	OWL Control Lists
Inference	(used by all) Resolve formal Semantics	Description Logics
Inference Features	Disambiguate complex information/relationships	
TBox	For concept-level reasoning on schema	~500,000 concepts @ .5ms
ABox	For complete DL reasoning on instances	~250,000 instances @ 1s
DBox	For datatype reasoning on abstract class types	Same as TBox
Instance Store	For tune-able (less complex) reasoning on instances	~2,000,000 instances @ 1s

Figure 14: Summary OWL Mediation Requirements

For the geospatial community, the Mediation Service would provide the technical capabilities and performance characteristics described in Figure 14 in order to facilitate rapid, sensible, and simpler exchange of geographic data to those that need it. The technology employed is more complicated than traditional approaches, but the user interfaces (e.g.: query, API, or online composite services) would be much simpler than the brittle techniques commonly employed in non-Mediation-Service environments.

Summary Characteristics and Metrics

Thus far, the discussion of GIS mediation has been partitioned into two primary threads:

1. Understanding the role and capabilities of an RDF spatial platform
2. Understanding the role and capabilities of an OWL mediation platform

It will be useful now to examine how these two focal points can be converged into a single GIS mediation platform. The next section, "Solution Architecture", will describe the architectural alignment between these two components and define the primary interfaces and connections to external systems such as Grids and relational databases. Figure 15, a summary features and performance table, describes the complete set of attributes for the GIS Grid Mediator Service.

Capabilities	Description	Metrics/Technical
Mediation Features	Reconcile different GIS semantics on the information	
Query Broker	Runtime query listening over federated sources	XQuery
Pub/Sub Events	Set notifications based on service events	JMS via SOAP/WSDL
Publishing	Assemble heterogeneous data into consumable sets	XML content assembly
Registry/Discovery	Discover what services are available via descriptions	OWL Directories
Security	Multi-level role and user based restrictions	OWL Control Lists
Inference	(used by all) Resolve formal Semantics	Description Logics
Inference Metrics	Disambiguate complex information/relationships	
TBox	For concept-level reasoning on schema	~500,000 concepts @ <1ms
DBox	For datatype reasoning on abstract class types	Same as TBox
ABox	For complete DL reasoning on instances	~250,000 instances @ 1s
Instance Store	For tune-able (less complex) reasoning on instances	~10,000,000 instances @ 1 s
RDF (Subsumption)	SubClassOf and SubPropertyOf axiom support	~30,000,000 instance @ 1s.
RDF Management	Manage GIS graph data over full lifecycle	
Query	Runtime query listening over federated sources	SQL + RDF_MATCH
Inference	Set notifications based on service events	RDF subsumption
Security	Standard RDBMS security management	RDB ACL
Clustering	Standard database clustering	Oracle RDBMS
Partitioning	Standard database partitioning	Oracle RDBMS
Loading	Data loading and transformation into RDF	Oracle RDBMS
OWL Management	Attributes of GIS data population	
Query & Inference	Multi-user, multi-session, hybrid reasoning	Cerebra
Modeling	Team-based, secure, versioned, object-level	Cerebra
Clustering	Engine-level, multi-threaded	Cerebra
Mapping	External retrieval (SOA, DBMS, Spatial)	Cerebra
Spatial NDM	Network Data Model, core Spatial features	
Modeling	Hierarchical and constraint-based modeling	Oracle Spatial
Types and Functions	Spatial data management	Oracle Spatial
Spatial Analysis	Spatial intersect, union, buffer, overlap, etc.	Oracle Spatial
Network Analysis	Transitive closure and path computation	Oracle Spatial
Constraints	Rules (network constraints) and cost-based queries	Oracle Spatial
Visualization	Graph, spatial, raster and topology information	Oracle Spatial

Figure 15: Summary GIS Mediator Features and Performance Metrics

Given these attributes in Figure 15, how can a GIS mediator be constructed and used in actual practice? The next section describes an approach.

Solution Architecture

In the most general sense, it is appropriate to think of RDF and OWL as a metadata layer on top of existing legacy systems. OWL will behave as a function schema to provide views into legacy information and RDF will behave as a highly flexible data format for reconfiguring information into the appropriate sets. Simplistically, the Mediation Service will use the information contained within OWL and RDF to retrieve legacy data from its native source, or convert legacy data into graph data so that it can be easily disassembled and reassembled into different structures as driven by inference techniques. It is in this manner that geospatial data consumers can gain access to a wide variety of content without having to hard-wire connections to the specific legacy sources–the mediator handles the content assembly for them.

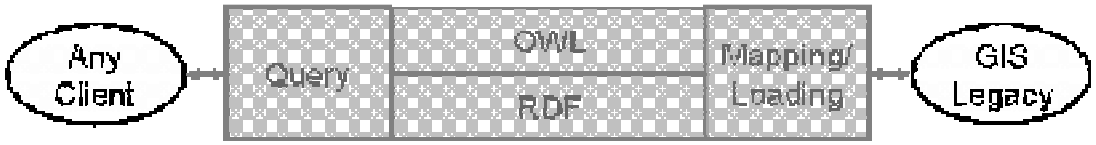


Figure 16: Simple Conceptual View

Grid Use Case View

The Grid component exposes several APIs to the SOA Grid. There are multiple access points to fulfill specific functions. The typical access layers are Grid-friendly XQuery for OWL-based queries and XML document responses. Alternately WSDL or Java APIs can be used to accomplish similar behavior over different protocols. In principle, this service is a simple I/O-based software service, with powerful behind-the-scenes, black-box inference capabilities.

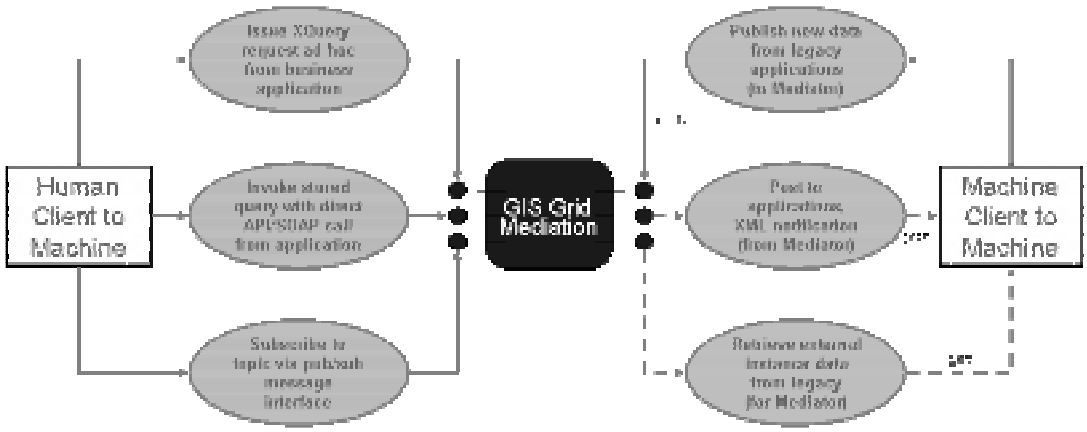


Figure 17: Grid Use Case View

Logical Information View

The following simplified logical diagram in Figure 19 shows how OWL may be used to logically connect underlying data from rule systems, XML documents (Web services), relational data (relational databases), and RDF. The approach for connecting external data sources to OWL typically involves a mapping annotation to relate individual data objects to concepts within a given ontology. With OWL, the system can perform inferencing at varying granularity, all the

way down to pure RDF levels. Alternately, data can be retrieved directly from the spatial sub-system when local functions are required.

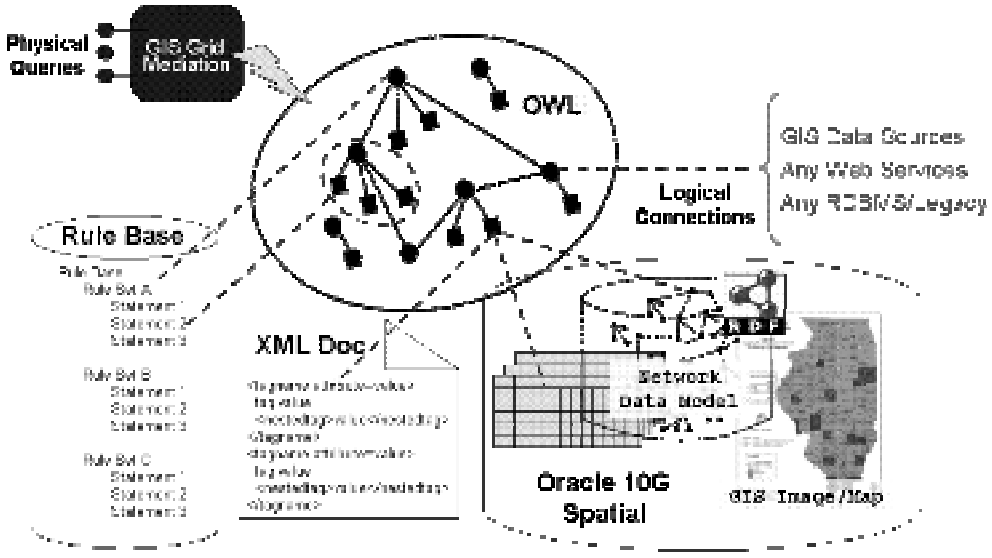


Figure 18: Logical Information View

Internal Functional View

Specific details as to the internal functional integration of the OWL and RDF systems are a proprietary approach developed by Oracle and Cerebra, thus these details are omitted. In principle, the basic idea is to expose an XQuery engine via WSDL/SOAP interfaces. The functional behavior of XQuery is enhanced to handle inference/reasoning tasks with an OWL DL inference engine. The OWL inference engine has external handles to legacy sources (via Service Oriented Architecture) and the RDF Spatial store, it can assemble queries to these external sources according to the inferences made at runtime. The Oracle 10g platform provides a single framework for accommodating relational, RDF, and spatial data persistence–which is used as a hub for migrated data. Typically, users would migrate data into RDF and Spatial formats when graph optimizations and instance-level reasoning needs to be accomplished. Additionally, the OWL platform can call out to Grid services, behaving as a virtual hub, when schema (ontology) level inference is all that is required.

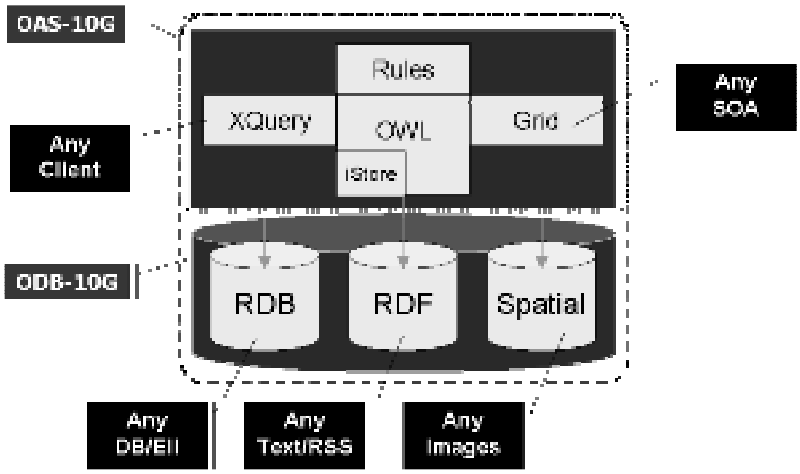


Figure 19: Internal Mediation Architecture

Federated Data Security

Security is an important and complex subject for any software system. To be secure in a holistic sense, a multi-level security approach must be taken; this should include applications, protocols, databases, and infrastructure-level security. Typical solutions would likely include network firewalls, application firewalls, encryption, and application authentication and authorization. For the purposes of this paper, security is considered within the context of the Mediation Service, emphasizing an approach suitable for the federated nature of this solution.

The innovative approach taken here[9] is commensurate with the rest of the solution and draws from experience deploying dozens of inference-based solutions. An ontology is used to capture Access Control List (ACL) data, which is in turn used to constrain the views available to particular roles and users of the mediator. There are two key advantages to using an inference-driven approach for security: (1) access is derived via classification, which allows for more nuanced categorization of user rights and (2) the security checks are validated with inference, which ensures a global guarantee of correctness when using OWL-based ACLs.

A simple example of these benefits would be to consider a user, JohnS, who is assigned to the Analyst role. In the domain ontology, Analyst roles have been prevented from seeing geographic features that are associated with ClearanceLevel2 realms by using native OWL axioms. If an Administrator were to attempt to override role-level assignments by associating JohnS with a geographic feature that was part of a taxonomy restricted to ClearanceLevel2 privileges, the inference engine would generate an "inconsistency" warning when asserting the new assignment, thus allowing the system to reject the change and give the Administrator specific options to correct the error. This kind of consistency checking is native to all OWL DL inference engines.

While the Mediation Service uses ontology-based ACLs for monitoring user access from the Grid, the RDF database system uses security controls identical to traditional database systems. Thus, the Grid service can connect to and leverage any RDF database on the network so long as it has the appropriate read/write privileges necessary to perform its business functions. In this way, the internal Grid Mediator architecture is quite similar to a classic 3-tier application server based system. In essence, the OWL models represent the domain security logic and the RDF system provides a persistence mechanism for operating with all the data.

Ontology Engineering Methodology

Knowledge engineering for a domain as complex and detailed as geographic can be a difficult process to get right and, in most organizations, there is rarely one individual that is both a domain expert and an ontology expert. To this end it is important to have a rigorous and methodical approach to capturing knowledge and constructing domain ontologies to ensure it is done correctly. A two-stage approach was adopted for constructing the ontologies.

In the first step[10] a "conceptual ontology" is developed. The conceptual ontology is intended to be primarily for human consumption: it attempts to balance the need for maximal formality of the ontology whilst retaining clear human comprehension. It is produced by domain experts, enabling them to structure and classify their knowledge of the domain in an explicit human-

readable model as well-structured English. This is not merely a glossary, which typically captures the meanings of words in a vocabulary, but a documentation of the concepts in the domain of interest.

The second stage develops a machine-interpretable representation of the conceptual ontology using OWL DL. This process often involves extensive discussion between the domain modeler and the OWL expert to ensure intended meaning is captured correctly.

If an ontology is to be useful, it is important that time is taken to construct it correctly. One major criticism of many existing ontologies found on the Web is that they tend to hide semantics in concept names, and hence not expose them to the machine. Furthermore, people tend to misuse subsumption hierarchies. Hierarchies often contain somewhat meaningless concepts at a higher level. One example in a geographic ontology might be the tendency to split all geographic features into "natural feature" and "man-made feature" at a high level. In the real world it is unlikely that many concepts fit unambiguously under either "natural feature" or "man-made feature". There is a tendency to try to artificially shoehorn concepts in a hierarchy under meaningless high-level concepts. Errors can also creep in when hierarchies are too deep. Figure 20 shows an example taxonomy containing some of the modeling errors discussed.

Initial Hierarchies

```
Substance     GeographicObject
 Water         WaterBody
  FreshWater      NonTidalWaterBody
  SaltWater       River
              FreshWaterBody
               River
              NaturalWaterBody
               River
             ManMadeStructure
              Building
               School
                RidingSchool
               Church
             EducationService
              School
               RidingSchool
```

Figure 20: Poorly Modeled Geographic Concepts

A good methodology should make the modeler think more carefully about how subsumption hierarchies are formed. In Figure 20, notice how the concepts "River" and "School" are specialized along a number of different conceptual axes. For example, "School" is specialized according to both its structure and its function. This highlights yet another problem: people tend to confuse "partonomies" with hierarchies. In reality, a school has parts that are buildings, but a school is never actually a subclass of a building.

Hierarchies are developed so that each concept only has at most one parent, and specialization down a hierarchy is subject to exactly one implicit differentiating notion per concept (Figure 21). The top-level concepts of each hierarchy are mutually disjoint, and all sibling classes in the hierarchy are disjoint. This approach also helped us identify key concepts in our ontology. For example a school may have functions other than that of education so we differentiate "School" (the real world object) from "SchoolRole" (a concept representing the intended use of a school). Any multiple inheritance can be inferred from the ontology model using an inference engine. For example, given the ontology in Figure 20 it can be inferred that a River is a subclass of a FreshWaterBody.

Typically geographic concepts in a domain ontology are described in terms of their "form and function", their physically measurable properties (e.g. geometry), taxonomic relationships, mereology, and topology.

An initial ontology skeleton is built from primitive concepts representing the physical forms of geographic objects (e.g. structure, building, rivers, land, etc.) and functional uses of geographic objects (e.g. industry, place of worship, education, recreation, etc.–these are basically the "roles"). These forms and functions can then be combined to describe and define real world geographic objects (e.g. School, Church, etc.). Figure 21 shows a fragment of an ontology built from the concepts discussed so far.

Normalized Skeleton Taxonomies

Substance	Form	Function
Water	WaterBody	EducationServiceRole
FreshWater	River	SchoolRole
SaltWater	Structure	UniversityRole
	Building	PlaceOfWorshipRole

Linking Axioms

- School equivalentClass hasForm someValuesFrom (hasPart someValuesFrom Building) and hasFunction someValuesFrom SchoolFunction
- Church equivalentClass hasForm someValuesFrom Building and hasFunction someValuesFrom PlaceOfWorshipRole
- FreshWaterBody equivalentClass WaterBody and contains someValuesFrom FreshWater
- River subClassOfWaterBody and contains someValues from FreshWater

Figure 21: A Better Modeled Ontology

Mereology is important when describing geographic objects. Mereology describes geographic concepts in terms of their component parts, for example one might say that a river is made up of a river mouth, a river source, and river stretches. It can also be used to specify dependencies of existence, for example a flat or apartment must be part of at least one building.

Obviously a very important aspect of a geographic ontology is describing the topological and spatial relationships between various concepts. For example one might want to encode information like:

- A river is spatially adjacent to a river bank
- A river stretch spatially overlaps a river mouth
- A dam traverses a river
- A bridge crosses a river

The conceptual and logical ontologies are intended to complement each other. There is still clearly room for ambiguity in this conceptual ontology, indeed, the authors have found that formal logical ontologies are only unambiguous to a human if they are familiar with that formal language. However, in our experience this is not the case for many domain experts. Thus, in addition to using a well developed methodology, it is important to annotate concepts in an OWL ontology with descriptions given in the conceptual ontology to ensure the intended meaning of individual concepts is captured in a human-readable dictionary form.

Business Value

Organizations ranging from life sciences, to defense/intelligence, to manufacturing are realizing the power of semantic technologies. However, they are also recognizing that full value of semantic technologies can only be recognized when the semantic applications are an integral component of IT systems. Until semantic applications are readily integrated with existing (non-semantic) business and geospatial applications, there is a high risk of recreating islands of non-interoperable RDF data, ontological artifacts, geospatial data, and business data. Hence, the promise of semantic technologies will be more fully realized as they are increasingly integrated with mainstream IT technology. For the IT practitioner, this means that RDF and OWL data must be managed just like other data types using standard RDBMSs; semantic mediation tools must be readily incorporated into middle tier application servers, and Extract, Transform and Load (ETL) tools must evolve to ensure that unstructured data can be easily loaded into RDF databases.

The authors are seeing a rapid uptake in the amount and usage of geospatial data in business and government applications. Over the past few years there have been three major advances for geospatial data-sharing using Grid-based mediation services.

The first breakthrough has been the ability for IT and GIS applications to share and make use of spatial and related information in a seamless manner. Openness doesn't simply mean across GIS; it means across all classes of applications. This is enabled through open geospatial data types (vectors, rasters, network and topology models) to be accessible through standard programming conventions (SQL, Java, and XML). This is essential for any production quality, scalable, computationally intensive application like those found in utilities, energy, modern land management, and telecommunications applications where security, robustness, and the ability to extend and adapt applications is essential. Virtually all enterprise systems in government, “traditional GIS”, and operational ERP software in business require and use this new IT-based infrastructure for open geospatial information.

The second breakthrough has been the extension of Web services and Grid technology to enhance access to geospatial information. Web services are excellent for accessing information in consumer and constituent applications. A growing number of government and consumer applications that have emerged are now deploying a Web services model. Examples such as the US Geological Survey's Geospatial One-Stop portal, Google Maps, and MapQuest are examples of government and commercial Web services that others are increasing embedding into their own

applications and services. However, as the need to access and combine data from multiple Web services and local enterprise databases grows, the importance of mediating the data across these sources becomes paramount. Mediation engines use ontologies to overcome the complex application-specific or Web-services-specific mappings common when integrating multiple data sources and data types into one coherent compound service.

This leads to a third, more recent, advance enabling the integration of business and specialist applications and services. The challenge here is that the flow of information is unremitting, the data types are heterogeneous and it is not clear what information resources need to be integrated a priori. Currently, different users of geospatial applications might use different syntactic standards, organize information in different ways, and even use different terminologies to refer to the same information. Successful access to, and integration, of these information resources hinges on the ability to address these different kinds of heterogeneity. In the past, this has meant having to create brittle application-to-application connectors. Also, mental models and keywords for accessing data generally diverge between subject areas and communities; hence, the need to manage different ontologies.

RDF, along with geospatial ontologies, introduces a means to semantically describe data for the purpose of semantic interoperability. These descriptions provide the basis for improved automation of the tasks of service discovery, composition, mediation, and invocation. In short, these advances provide the framework for geospatial information retrieval based on the semantics of spatial and business ontologies. By explicitly representing the role of semantics in different components of the information retrieval process (people, interfaces, search systems, and information resources), the mediation approach enables users to more precisely retrieve the data they need, based on the semantics associated with these data[11].

Benefits of the GIS Grid Mediation Approach

There are at least two dimensions along which to measure the value of a given software approach: technical and business. Typically the former has value to the CIO organization, where the latter has value to the business as a whole.

Data Management Benefits

- Greatly reduced complexity of systems management
  - Especially in heterogeneous environments
  - No universal agreement or global model is required
- Support for leading standards: W3C and Open Geospatial Consortium (OGC)
  - Standard OWL/RDF/XML and spatial data types
  - Proprietary data structures and formats are avoided
  - Universal, standards-based interfaces – Java APIs, XQuery, XML, and SQL
- Advanced, more automated, intelligent data functions
  - Smart query (graph traversal, implied constraints, hybrid reasoning)
  - Hybrid reasoning (multiple inheritance, intersection, union, transitivity)
  - Native spatial functions (union, nearest, within-distance, graph match)
- Support by leading semantic and geospatial tool vendors
  - RDF, OWL, and Spatial data can be managed just like regular data
  - Scalability, data security, replication, partitioning, parallel queries, bulk load
  - Support for both short and long transactions (version management)

Organizational Benefits

Equally important to the data management benefits of Spatial data within the DBMS are the organizational benefits. Examples of how DBMS-centric spatial technology increases operational efficiency while reducing cost include:

- **Low Cost of Ownership:** Combined semantic and geospatial applications can be deployed on a corporate level with data stored centrally, lowering ownership costs. Beyond the advantage of central data storage and query, Service Oriented Architectures eliminate the need to install and maintain client-side software on the desktop, and store and manage data separately, outside of the corporate database.
- **Low Risk:** RDF and spatial information can be integrated directly into the corporate DBMS. This results in scalable, secure, high-performance applications. Customers can choose to deploy on any server platform (Unix, Linux, or NT) using existing IT resources to manage these applications.
- **High Value:** Using the Internet, much larger numbers of users can access the application at virtually no additional costs to the organization. This means that all the users that need to access mission critical information can do so 24x365.
- **Increased Usability:** There is the perception that semantic and geographic information systems are complex and difficult to use. While this was true in the past, it is now possible to build semantically and location enhanced business applications that are accessible through intuitive web-based client applications.

Location Enabling the Enterprise

The emergence of location based services is now viewed more broadly than simply the delivery of maps and driving directions to end users. Businesses, government agencies, and nonprofits are routinely incorporating location into their front and back office applications. This helps to make their business more efficient and creates a positive business experience for customers and partners. By integrating enterprise information with location-enhanced customer information, organizations obtain more comprehensive business intelligence. Integrating Web mapping services, Web feature services, and tracking capabilities along with enterprise applications is a powerful method for improving the business intelligence. For example, wireless carriers can reuse customers' calling location information to analyze and better support a variety internal operations ranging from billing, customer care, network load monitoring, predictive analysis, and network planning. This use of location information, combined with the increased availability of location and mapping services is rapidly propelling location-based services into the mainstream. Other types of business operations like fleet management, asset tracking, knowledge management, and supply chain operations are similarly benefiting from location-enabling their activities.

Operational GIS Business Cases

The combination of complicated data, geographic distribution, and cultural heterogeneity poses major challenges that are typically solvable with only brute force methods, and require massive efforts to change. The following section will explore increasingly complex examples from Ordnance Survey as a way to describe the business case using a geospatial example.

Classification Alignment

Different organizations tend to have different views of the world. Often the meaning of a word can differ subtly or classification schemes can be at different levels of semantic granularity. Figure 22 contrasts two fragments of classification schemes from the United Kingdom Ordnance Survey and UK Valuation Office.

Valuation Office	**Ordnance Survey**
School_and_Premises	EducationServices
LocalAuthoritySchool	School
IndependentSchool	PrimarySchool
	SecondarySchool
	PrivatePrimarySchool
	PrivateSecondarySchool

Figure 22: Two Taxonomic Views of the World

Both classification schemes provide information on schools, but they are clearly structured differently. Here we can use a simple set of axioms (see Figure 23) to provide semantic interoperability between the Ordnance Survey and Valuation Office data. However, this is a relatively simply example. Current research at Ordnance Survey is looking at how to best interoperate between two, potentially very large, ontologies where the mappings might not be quite as straightforward as those shown in Figure 23. We do not propose that ontologies will solve all of our interoperability problems, but they are certainly an important part of the solution.

Valuation Office	**Ordnance Survey**
School_and_Premises	EducationServices
LocalAuthoritySchool	School
IndependentSchool	PrimarySchool
	SecondarySchool
	PrivatePrimarySchool
	PrivateSecondarySchool

Taxonomy Alignment

School *equivalentTo* School_and_Premises
PrimarySchool *subclassOf* LocalAuthoritySchool
SecondarySchool *subclassOf* LocalAuthoritySchool
PrivatePrimarySchool *subclassOf* IndependentSchool
PrivateSecondarySchool *subclassOf* IndependentSchool

Figure 23: the aligned taxonomies

Adaptable Data

After structuring and aligning the ontology, the database then has to be queried against the physical schema. This can often cause much confusion. Ontologies enable business, domain, and policy knowledge to be captured in a far more intuative and portable information model, creating

a view into the data that can be easily tailored and adapted for different user needs. To achieve this we envisage having a layer of at least three ontologies sitting on top of a database.

The first ontology, the data ontology, essentially provides an interface between the database schema and the domain ontology. Often, the data ontology will be structurally similar to the relational database it is intended to interface with, this is done to facilitate the mapping process.

The second ontology, the domain ontology, describes the geographic and topographic domain from the Ordnance Survey view of the world. With these two layers in place, we query our database through the concepts described in our ontology instead of the traditional symbol matching approach. This allows us to more easily specialize and generalize our queries. For example, we could ask our database for all the "Education Services" in Southampton and it could return all the schools, colleges, and universities even though nothing is explicitly typed as "Education Service" in the database.

The third ontology layer, "the application ontology", describes a more specialized ontology based on either a particular application or a third party view of the world. Here new concepts can be constructed from, or mapped to, concepts in the domain ontology. One might imagine in a flood disaster management scenario defining new concepts such as "emergency accomodation" from the concepts "school" and "hospital". Instances in the database can then be dynamically reclassified based on the axioms in the application ontology. The authors have witnessed this stacked ontology approach help real-world data become more adaptable for different needs.

Semantic GIS Queries on the Grid

Ordnance Survey is interested in combining semantic technologies with spatial technologies to provide really powerful location based services (LBS). Spatial databases allow one to use spatial functions and operators to answer queries like:

- "list all school zones crossed by this railway line"
- "find all pizza parlors within this area of interest"
- "return the 10 hotels which are closest to the airport, and the distance to each in miles"

So one can imagine that a simple example of combining spatial queries with semantics would be generalizing the query "find all pizza parlors within this area of interest" to, say, "find all restaurants within this area of interest".

A more interesting application, going back to the scenario of flood disaster management, is trying to find all emergency accomodation in the event of a flood. One could define "potential emergency accomodation" as being all buildings that are within 10 miles of a hospital. The inference engine will figure out which objects in the database are buildings (as they may well be classified at a lower level as schools, churches, etc.) and the spatial engine will then determine which of these are with 10 miles of a hospital. Objects in a database could then be dynamically reclassified as emergency accomodation according to these rules.

Some may argue that finding all emergency accomodation might be done more easily using standard SQL with spatial operators. However, the issue of ambigious classification of geospatial objectes is well known[12], and we expect it to be more useful to have information about, for example, "emergency accomdation" captured in the portable and reusable form of an ontology.

Furthermore, additional considerations such as the emergency accomodation's location relative to flood zones and the requirement for buildings to accomodate a certain number of people might require semantic interoperability with additional data sources.

Other Business Domains in Need

It should be clear by now that many industries suffer from the same challenges that have been discussed in this paper, some of these opportunities are discussed below.

Life Sciences and Drug Discovery

RDF and OWL-based ontologies are rapidly gaining ground in life sciences and medical diagnostics. They range from full-fledged ontologies such as Gene Ontology (Consortium), and BioPAX (funded by the Department of Energy), to controlled vocabularies such as National Library of Medicine's SNOMED Clinical Terms.

For example, a graph data model is well-suited to storing and analyzing metabolic pathways. Data from the KEGG metabolic pathway database from the University of Kyoto is managed in Oracle10g as a directed network. The chemical compounds are stored as nodes and the enzymes are stored as links. A Java visualization tool is used to view the metabolic pathways and to interrogate the data using analysis including shortest path and all paths. Metabolic pathway analysis can guide researchers as to which drug discovery targets are likely to yield the most beneficial results.

In the case of gene expression, gene expression profiles provide useful insights into the underlying mechanisms of disease. Signaling pathways are stored in a graph model with genes represented as the nodes and interactions between the genes as the edges. Gene expression data from biological samples can be stored in a separate relational table. Queries can be performed that span both the signaling pathway and the gene expression data, enabling researchers to identify subsets of the signaling pathway that are likely to be impacted by the changes in the gene expression profile.

Product Master for Pricing and Configuration

Several Fortune 500 manufacturing companies have made significant investments in RDF- and OWL-based IT systems to help manage the complexity of their product data. In effect, IT groups are converging on the notion of RDF/OWL systems becoming the nerve center, or Product Master, of their complicated data environments.

The reason for this shift is simple; it is easier to repurpose data that has been described by RDF and OWL metadata. An example is a company that has to generate quarterly financial reports about sales figures for their product line. Historically, the issue had been that different internal business groups (Finance, Marketing, Manufacturing, and IT) used different taxonomies for organizing their product. This is not unusual for a company with over 40,000 products, never mind the fact that these are electronic systems that share parts and have complex engineering processes for constructing them. A mediator style solution allows this company to create multiple views into common data, without migrating instances or creating new database schema, while also allowing for "what-if" analysis into hypothetical reporting hierarchies.

Another top-flight manufacturer, this one a software manufacturer, faced persistent product and license program launch barriers due to inefficient pricing processes. Their discount system had

become too complex because of the sheer number of products and programs they needed to support. An OWL/RDF-based system allows them to infer the implicit discount levels for programs and products with guaranteed computational accuracy, thus reducing the manual nature of classifying the price-points for the millions of potential product and program combinations within their software systems.

Metadata and Content Management

A critical sister-component of the Grid Mediator is the Grid Repository. Largely unaddressed in this paper, the repository allows for metadata management, schema management, and management of various content objects within a business. Likewise, the repository typically exposes a registry interface (such as UDDI or ebXML) for the discovery of both content and services within a given IT domain.

This Grid Repository would provide the mediator with up-to-date versions of ontology, XML schema, and other content objects and it would provide the basis for finding external services that were available for the Mediator to retrieve data from. The Grid Repository can also act as an overlay to traditional Content Management Systems (CMS) such as Documentum.

Although we have not covered its operation in this paper, there are several authoritative sources[13] that describe the role and functions of next-generation metadata and content management systems making use of OWL/RDF technology.

Conclusion and Lessons Learned

In this paper the authors have provided the reader with an informative overview of the challenges faced when attempting to share geospatial information, by analogy these challenges should clearly resonate as common within and across other industry domains.

After many years of building, deploying, and working with RDF and OWL based systems, the authors have drawn many lessons about semantic technologies, but perhaps none is more important than the single largest motivating factor for this paper: ***they work***.

Clearly, the goal of this paper was to articulate a design solution for a semantics-driven mediation engine, one which is commercially supported with vendor tools, capable of enterprise class performance and scalability, as well as powerful enough to provide dramatic new efficiencies for managing data within the Grid, and more generally, a Service-Oriented Architecture.

The authors have certainly focused on a technology discussion for this paper, but as practitioners, we have duly noted the equal importance of people and process. After many implementations it is clear that there is no replacing a good data modeler when it comes to working with RDF and OWL. Likewise, a good data modeler is always made better while working within a sound process. No technology is a substitution for careful staffing and guidance within an IT organization. But great technology can make the laborious more efficient and turn the impossible into the possible.

RDF and OWL technologies have exited the ivory tower. While the grand vision of the Semantic Web may still be several years off, we hope to have shown the reader how these semantic technologies can begin to change organizations today.

Appendix A:

Querying RDF Schema and Rules in Oracle10g

This section presents introduces five simple exercises combine spatial and semantic queries using Oracle10g. These examples use a US territories model showing hierarchical relationships reflected on the graph in Figure X. This figure also highlights some of the relations between the territorial boundaries.

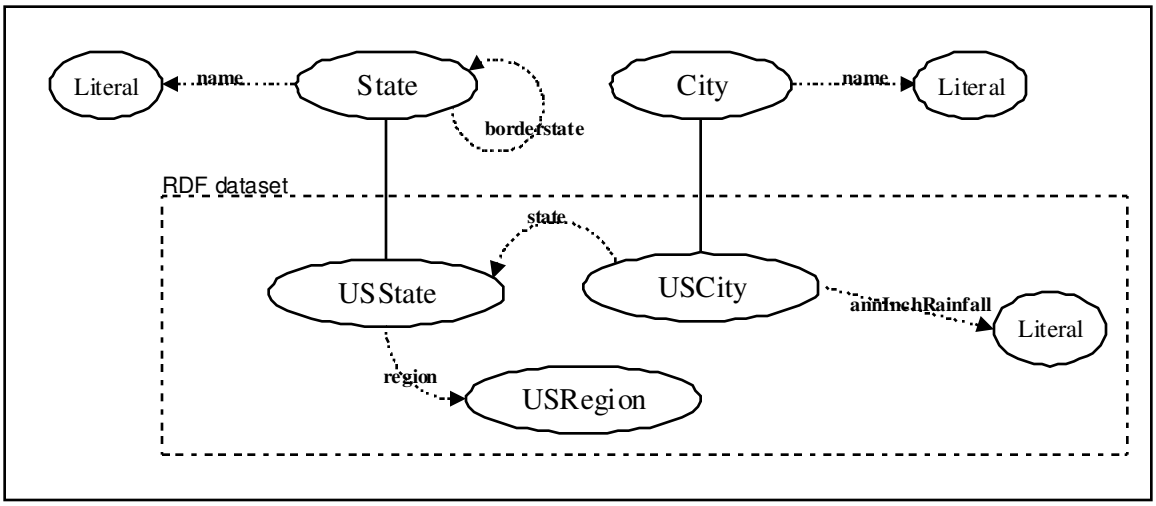


Figure X: US Territories RDF Model

Data Sources for these Exercise:

1. http://www.rdfdata.org/data.html#geo
2. http://www.daml.ri.cmu.edu/ont/USCity.daml
3. http://www.daml.ri.cmu.edu/ont/USRegionState.daml
4. http://www.ncdc.noaa.gov/oa/climate/online/ccd/nrmpcp.txt

Exercise 1. Select USStates bordering Colorado.

```
SELECT s states_bordering_co
FROM TABLE(SDO_RDF_MATCH(
'(?s usrs:borderstate usrs:CO)',
SDO_RDF_Models('us_territory'),
null,
SDO_RDF_Aliases(SDO_RDF_Alias('usrs','http://www.daml.ri.cmu.edu/ont/US
RegionState.daml#')),
null));

/*
STATES_BORDERING_CO
--------------------------------------------------
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#AZ
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#KS
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#NE
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#NM
```

```
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#OK
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#UT
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#WY
*/

SET LINES 100;
COLUMN city format a40;
COLUMN rainfall format a10;
```

Exercise 2. Select USCities with 37" of annual rainfall.

```
SELECT n city, r rainfall
FROM TABLE(SDO_RDF_MATCH(
'(?c noaa:annInchRainfall "37.0"^^xsd:decimal) (?c noaa:annInchRainfall
?r) (?c usc:name ?n)',
SDO_RDF_Models('us_territory'),
null,
SDO_RDF_Aliases(SDO_RDF_Alias
('noaa','http://www.ncdc.noaa.gov/oa/climate/online/data#'),
                SDO_RDF_Alias
('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')),
null));

/*
CITY                                     RAINFALL
---------------------------------------- ----------
Grand Rapids                             37
Rockford                                 37
Portland                                 37.0
*/
```

Exercise 3. RDFS inferencing on the us_territory model.

Step 1: Create a rules index. The rules index pre-computes triples that can be inferred from applying the rulebases. The RDFS rulebase implements RDFS entailment rules and is Oracle-supplied.

```
BEGIN
SDO_RDF_INFERENCE.CREATE_RULES_INDEX(
'us_territory_rdfs_rbidx',
SDO_RDF_Models('us_territory'),
SDO_RDF_Rulebases('RDFS'));
END;
/
```

Step 2: Inferencing query with the RDFS rulebase. For this example we will SELECT all the Cities in ME. Please note that the dataset does not contain an explicit City class. Using the RDFS rules usc:USCity is an rdfs:subClassOf ac:City; a usc:USCity is therefore a type of ac:City.

```
SELECT cn ME_CITIES
FROM TABLE(SDO_RDF_MATCH(
'(?n rdf:type ac:City) (?n usc:state usrs:ME) (?n usc:name ?cn)',
SDO_RDF_Models('us_territory'),
```

```
SDO_RDF_Rulebases('RDFS'),
SDO_RDF_Aliases(SDO_RDF_Alias
('ac','http://www.daml.ri.cmu.edu/ont/City.daml#'),
                SDO_RDF_Alias
('usrs','http://www.daml.ri.cmu.edu/ont/USRegionState.daml#'),
                 SDO_RDF_Alias
('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')),
null));

/*
ME_CITIES
------------
Augusta
Lewiston
Portland
York
*/
```

Step 3: Perform query without RDFS inferencing. In this example, Selecting all the cities in ME without using RDFS inferencing will return a null set (e.g., rdf:type->ac:City is not found).

```
SELECT cn ME_CITIES
FROM TABLE(SDO_RDF_MATCH(
'(?n rdf:type ac:City) (?n usc:state usrs:ME) (?n usc:name ?cn)',
SDO_RDF_Models('us_territory'),
null,
SDO_RDF_Aliases(SDO_RDF_Alias
('ac','http://www.daml.ri.cmu.edu/ont/City.daml#'),
                SDO_RDF_Alias
('usrs','http://www.daml.ri.cmu.edu/ont/USRegionState.daml#'),
                 SDO_RDF_Alias
('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')),
null));

/* no rows selected */
```

Exercise 4. Inferencing on us_territory model using a user-defined rulebase.

Step 1: Create a rulebase. Rulebases contain a set of rules that will be applied to queries.

```
EXECUTE SDO_RDF_INFERENCE.CREATE_RULEBASE('us_territory_rb');
```

Step 2: Insert a rule into the rulebase. Rules are stipulations that are applied to draw inferences from the data. For example: IF city x is in state y; and y is in region z; THEN city x is in region z.

```
INSERT INTO mdsys.rdfr_us_territory_rb VALUES(
'city_region',
'(?x usc:state ?y) (?y usrs:region ?z)',
 NULL,
'(?x usc:cityRegion ?z)',
SDO_RDF_Aliases(SDO_RDF_Alias
('usrs','http://www.daml.ri.cmu.edu/ont/USRegionState.daml#'),
```

```
                SDO_RDF_Alias
('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')));

COMMIT;
```

Step 3: Create a rules index with the RDFS and us_territory rulebases

```
BEGIN
SDO_RDF_INFERENCE.CREATE_RULES_INDEX(
'us_territory_rbidx',
SDO_RDF_Models('us_territory'),
SDO_RDF_Rulebases('RDFS','us_territory_rb'));
END;
/
```

Step 4: Perform inferencing query using RDFS and us_territory_rb rulebases.

```
-- Example: SELECT the states and regions with cities called "Augusta"
-- note: the dataset does not explicitly contain a usc:cityRegion
property, this relationship is inferred.

SET LINES 125;
COLUMN state format a60;
COLUMN region format a60;

SELECT s state, rg region
FROM TABLE(SDO_RDF_MATCH(
'(?c usc:cityRegion ?rg) (?c usc:state ?s) (?c usc:name "Augusta")',
SDO_RDF_Models('us_territory'),
SDO_RDF_Rulebases('RDFS','us_territory_rb'),
SDO_RDF_Aliases(SDO_RDF_Alias
('usrs','http://www.daml.ri.cmu.edu/ont/USRegionState.daml#'),
                SDO_RDF_Alias
('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')),
null));

/*
STATE                                                        REGION
------------------------------------------------------------ ----------
------------------------------------------------
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#GA
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#DeepSouth
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#ME
http://www.daml.ri.cmu.edu/ont/USRegionState.daml#NewEngland
*/
```

Exercise 5. Combine the results of an RDF query with data from a Spatial table

The following example contains a combination of spatial and rdf query capabilities of 10gR2. The query is list the population of cities in the New England region. There are three datasets used in answering this query. The first is the RDF, RDFS dataset to determine which states constitute the New England Region. The other two states, and cities dataset contains spatial and population information. These datasets are used to determine those cities that lie in the New England states and lists their 1990 census population value.

The query and results are:

```
 SELECT c.pop90 Population, c.city || ', ' || c.state_abrv City
 from cities c, states s
 where s.state in
 (SELECT a.name state_name
 FROM TABLE(SDO_RDF_MATCH(
 '(usrs:NewEngland usrs:memberstate ?s) (?s usrs:name ?name)',
 SDO_RDF_Models('us_territory'),
 SDO_RDF_Rulebases('RDFS','us_territory_rb'),
 SDO_RDF_Aliases(SDO_RDF_Alias
('usrs','http://www.daml.ri.cmu.edu/ont/USRegionState.daml#'),
                SDO_RDF_Alias ('usc','http://www.daml.ri.cmu.edu/ont/USCity.daml#')),
 null)) a)
 AND
 sdo_inside(c.location, s.geom)='TRUE'
 order by population desc ;

/*
POPULATION CITY
---------- ----------------------------------
    574283 Boston, MA
    169759 Worcester, MA
    160728 Providence, RI
    156983 Springfield, MA
    141686 Bridgeport, CT
    139739 Hartford, CT
    130474 New Haven, CT
    108961 Waterbury, CT
    108056 Stamford, CT
    103439 Lowell, MA

10 rows selected.

*/
```

[13] Joe and Glen Paper
[4] Reference to Oracle Grid whitepaper
[5] This is not to imply that graph systems are superior in all cases – relational and XML based systems still have advantages (speed and simplicity respectively) that make them superior in many scenarios.
[6] Describe a reasoner
[7] Reference to Zaslavesky paper and others?
[8] Zaslavesky
[9] Reference to Cerebra whitepaper with OWL security
[10] Reference to Hayley
[11] Egenhofer 2004
[12] Reference to Zaslavesky paper and others?
[13] Reference to Cerebra whitepaper on "Active Metadata Management"