

Semantic Data Integration in the Life Sciences

*An Oracle White Paper
September 2005*

Note:

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Semantic Data Integration in the Life Sciences

Note.....	2
Introduction	4
Semantic Web Technology	4
Technology Overview	4
Resource Description Framework.....	5
Web Ontology Language (OWL)	7
Semantic Web Technologies in the Life Sciences	8
Data Aggregation in the Life Sciences.....	8
Ontological Aggregation of Data	9
Data Model Comparison.....	10
Oracle Spatial RDF Data Model.....	11
Model.....	11
Rulebase	13
Rules Index	13
Query	13
Conclusion.....	14

Semantic Data Integration in the Life Sciences

The Semantic Web defines and links data in such a way that it can be used for more effective discovery, automation, integration, and re-use across various applications.

INTRODUCTION

The Semantic Web has been developed as an extension of the current Web. It has been designed to give information well-defined meaning, thereby better enabling computers and people to work in cooperation. This is important as the mix of content on the web is shifting from exclusively human-oriented content to more and more data content. The Semantic Web also brings the idea of having data defined and linked in a way that it can be used for more effective discovery, automation, integration, and re-use across various applications.

SEMANTIC WEB TECHNOLOGY

Technology Overview

The Semantic Web builds upon a number of existing technologies (Figure 1). The foundation layer takes advantage of Uniform Resource Identifiers (URIs), which include Universal Resource Names (URNs) that allow things to be uniquely identified and Universal Resource Locators (URLs) that allow resources to be located. The foundation layer also takes advantage of the Unicode character-encoding scheme.

Moving up the technology stack, the Semantic Web uses XML and the XML extension for Namespaces. In a Namespace, a URI is modeled as a QName, which consists of a qualifier that identifies the vocabulary being used, and a fragment that indicates the element in the vocabulary. The Semantic Web assumes that there will be many different and perhaps overlapping vocabularies, and Namespaces provide a means of uniquely identifying every item in every vocabulary.

RDF Model and Syntax is the next layer in the technology stack, and is the common data format for the Semantic Web, as recommended by W3C. The RDF data structure consists of a triple, where each triple represents a statement about a resource consisting of a subject, a predicate, and an object.

RDF Schema is a language for describing RDF vocabularies. It can be used to describe class hierarchies and property hierarchies, and allows the domain and range of properties to be constrained. RDF Schema can therefore be used for inferencing.

The ontology layer within the Semantic Web stack of expressivity, is currently the highest level that has been well defined. The W3C standard recommendation at this

level is Web Ontology Language (OWL). For the most part, ontological knowledge systems are frame based, graph based or description logics. OWL supports the exchange and use of all forms of ontologies. As this layer can be used to describe cardinality constraints on properties, it can be used for more advanced inferencing than RDF Schema.

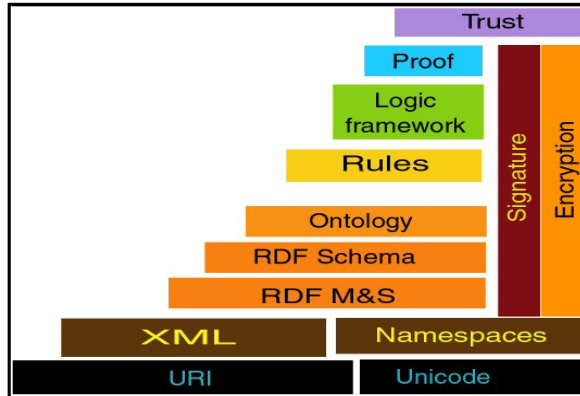


Figure 1. Stack of Expressivity as Defined by the World Wide Web Consortium

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object.

Resource Description Framework

The Resource Description Framework (RDF) is a language for representing information about resources in the Web. It was originally intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource. However, by generalizing the concept of a Web resource, RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web.

RDF is intended for situations in which this information needs to be processed by applications, rather than being only displayed to people. RDF provides a common framework for expressing this information so it can be exchanged between applications without loss of meaning. Since it is a common framework, application designers can leverage the availability of common RDF parsers and processing tools. The ability to exchange information between different applications means that the information may be made available to applications other than those for which it was originally created.

The underlying structure of any expression in RDF is a collection of triples, each consisting of a subject, a predicate and an object. Using a triple as the data structure enables RDF to represent simple statements about resources as a graph of nodes and arcs.

Figure 2 shows an example of an RDF graph that has been designed to represent information about an individual. Properties and values have been created to represent the name and e-mail address of an individual, as well as to highlight a meeting that the individual is attending. Data from different RDF graphs can be merged very effectively. When the blue graph in Figure 2 is merged with a second graph in red, it can be seen that the person is chairing a session at the meeting, and where exactly the meeting is being held. So using RDF, information can be assembled into useful blocks of knowledge. Any information expressed in RDF can be connected to any other information expressed in RDF, in much the same manner that any document expressed in HTML can link to any other document in HTML.

Using a triple as the data structure enables RDF to represent simple statements about resources as a graph of nodes and arcs.

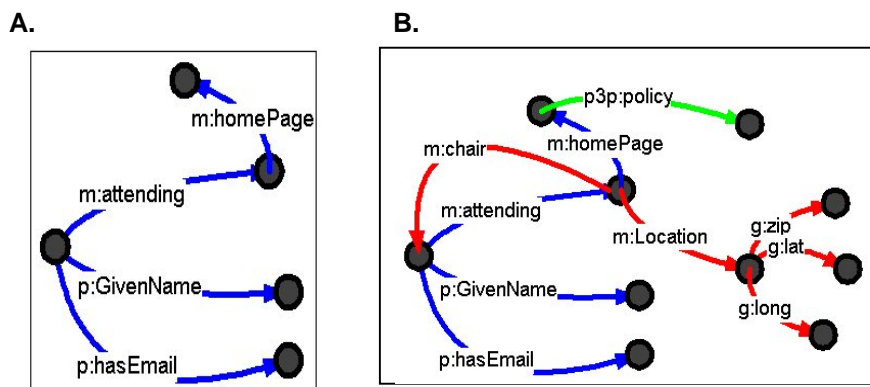


Figure 2. Using RDF Graphs to Represent Information about an Individual. A. The depiction of some basic information about an individual. B. The merging of two RDF graphs to retrieve additional information.

A URI reference or literal can be used to represent nodes. A URI reference used as a predicate identifies a relationship between the things represented by the nodes it connects. A blank node is a node that is not a URI reference or a literal, it is just a unique node that can be used in one or more RDF statements, but has no intrinsic name. RDF provides a built-in vocabulary intended for describing RDF statements. A description of a statement using this vocabulary is called a reification of the statement.

RDF provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs.

RDF provides an XML-based syntax (called RDF/XML) for recording and exchanging these graphs. Like HTML, this RDF/XML is machine processable and, using URIs, can link pieces of information across the Web. However, unlike conventional hypertext, RDF URIs can refer to any identifiable thing, including things that may not be directly retrievable on the Web. The result is that in addition to describing such things as Web pages, RDF can also describe cars, businesses, people, news events, etc. In addition, RDF properties themselves have URIs, to precisely identify the relationships that exist between the linked items.

Classes and properties are described as an RDF vocabulary, using extensions to RDF provided by the RDF Schema.

RDF provides a way to express simple statements about resources, using named properties and values. However, RDF user communities also need the ability to define the vocabularies they intend to use in those statements, specifically, to indicate that they are describing specific kinds or classes of resources, and will use specific properties in describing those resources. RDF itself provides no means for defining such application-specific classes and properties. Instead, such classes and properties are described as an RDF vocabulary, using extensions to RDF provided by the RDF Schema. The RDF Schema facilities are themselves provided in the form of an RDF vocabulary, as a specialized set of predefined RDF resources with their own special meanings.

SPARQL is a query language for querying information in RDF graphs. It provides facilities to extract information in the form of URIs, blank nodes, plain and typed literals. It is also designed to extract RDF subgraphs, and to construct new RDF graphs based on information in the queried graphs.

An ontology is a means of capturing knowledge about a domain, such that it can be used both by humans and computers.

Web Ontology Language (OWL)

An ontology is a means of capturing knowledge about a domain, such that it can be used both by humans and computers. The most important aspect of an ontology is that it creates a shared understanding of a domain. The knowledge is captured in conceptual form; that is, concepts that represent classes or sets of instances in the world. Ontologies relate concepts to one another through relationships, which may have constraints placed upon them. There are many ways of representing ontologies from lists of words; taxonomies, database schema, frame languages and logics. What differs between these forms of ontologies is their level of expressivity.

OWL is an ontology language that has been defined to be compatible with the architecture of the Web in general, and the Semantic Web in particular. OWL enables ontologies to be distributed across many systems, scales to meet the needs of the Web, is compatible with other Web standards, and is open and accessible.

OWL builds on RDF and RDF Schema, and adds more vocabulary for describing properties and classes, for example, relations between classes, cardinality, equality, richer typing of properties, characteristics of properties, and enumerated classes.

OWL is an ontology language that has been defined to be compatible with the architecture of the Web in general, and the Semantic Web in particular.

The use of RDF and OWL will smoothly interconnect personal information management, enterprise application integration, and the global sharing of commercial, scientific and cultural data.

OWL is intended to be used when the information contained in documents needs to be processed by applications, as opposed to situations where the content only needs to be presented to humans. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the Web.

SEMANTIC WEB TECHNOLOGIES IN THE LIFE SCIENCES

Organizations want to make decisions based upon all data. Integrating data is challenging as data is represented in different formats, with different identifiers to reference the same entities, there is acronym collision between the data sets, and many data types are used such as images, text and graphs. The Semantic Web can help organizations overcome these challenges.

Data Aggregation in the Life Sciences

Many people in the life sciences are very excited by the promise of the semantic web. They want to integrate data from many different data sources, so that they can make well-informed decisions, yet data integration has been challenging. The difficulties stem from data being made available in different formats for example different tab-delimited files formats, different XML schemas, and in different relational models. The task is also made harder because the data models frequently change as science progresses, and individuals learn that additional data is also relevant. In addition there is acronym collision across the data sources, and data can be in different data types for example graphs, images, text, and chemical structures.

Many data integration projects currently fail. One of the most common reasons for the failure is the inability to extend the data model to incorporate new data, or the inability to re-use data in ways that it was not originally intended. RDF provides a very flexible model for adding new data to a data model and for re-using data in ways that it was not originally intended. People are beginning to really appreciate the flexible triple syntax, as it is becoming recognized that things are always evolving, that people will always want to extend their system, or to look at data in a different way. Being cognizant of this constant change will be the first step towards companies saving money. People need to be able to re-use data and re-aggregate applications. They also like the idea of the serendipitous discovery of new information.

Screen snapshots of the BioDASH demo for semantic data integration are shown in Figure 3. The demo is available for download from the W3C's Web site (<http://www.w3.org/2005/04/swls/BioDash/Demo/>). In the left hand pane of figure 3A, the protein target GSK3beta is highlighted in red. In the demo, it is possible to click on the icon to find out more information, including the proteins UniProt identifier, and it's classification as a protein target within the RDF Schema ontology. It is also possible to click on the chemical compounds that GSK3beta interacts with to find out more information. The wnt pathway is shown in the right hand pane of figure 3A, and again it is possible to click on the icons to retrieve further information. For example, if you click on Glycogen Synthase Kinase 3 β , it is possible to see the UniProt identifier for the protein. As both data sets are in RDF, it's possible to drag and drop the GSK-3beta protein and it's chemical interactions from the target window in the left pane to the pathway window in the right pane. When the data is merged, it becomes apparent that GSK3beta and Glycogen Synthase Kinase 3 β are the same protein, as a rule was written requesting that merged entities be highlighted (figure 3B). It is also discovered by chance that one of the chemical compounds that interacts with GSK3beta also interacts with a second protein within the wnt pathway. This interaction has implications as to which protein target should be focused on for drug discovery.

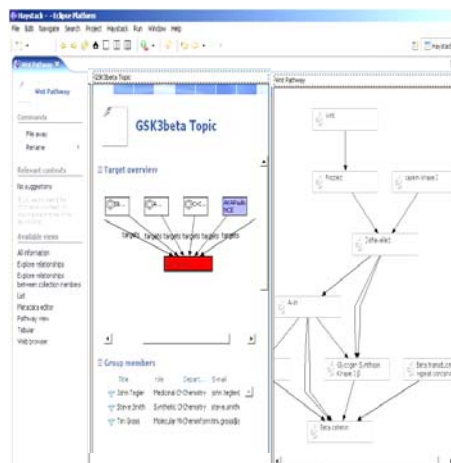
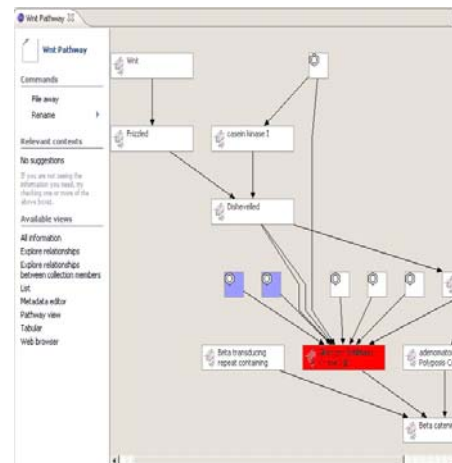
A.**B.**

Figure 3. The BioDASH Demo enables the Dynamic Aggregation of Protein Target Data and Protein Pathways Data

People are beginning to really appreciate the flexible triple syntax of RDF, as it is becoming recognized that things are always evolving, that people will always want to extend their system, or to look at data in a different way. Being cognizant of this constant change will be the first step towards companies saving money.

Ontological Aggregation of Data

The Semantic Web enables people to express well-defined and rich models using RDF schemas or OWL ontologies. These models contain a representation of all of the concepts present in a domain and all of the relationships between the concepts. These relationships can, at their simplest, use IS-A relationships, e.g. 5HT2B IS-A SEROTONIN RECEPTOR, which, when aggregated together, build into a hierarchy or taxonomy. The taxonomies of concepts (and relationships) can be very useful in their own right, especially when the concepts are annotated with properties such as synonyms. This allows users to specify high-level concepts such as GPCR when performing a search or selecting data for analysis. True ontologies however have a very broad range of relationships between concepts such as IS-EXPRESSED-IN, BINDS-TO, HAS-AFFINITY-FOR, IS-USED-FOR-TREATMENT-OF etc. These relationships too include all known synonyms. This allows, for example, all of the many different English variant forms of the BINDS-TO relationship between proteins and compounds to be used to build up a complete and detailed picture of the interactions around a given protein or protein family. This framework enables scientists to perform queries that span both the classes and properties, enabling researchers to ask the questions of interest. BioPAX is an example of such an ontology in OWL within the biological pathway domain (figure 4).

With the semantic web it would be expected that the boundary between classic search engines and inference engines would begin to blur. What is needed now is a system that understands and links meanings, not just matching strings of letters during a query. The inclusion of semantics would greatly enhance the search capabilities of these resources, and allow them to be tied to the other databases with more meaning than just a URL link.

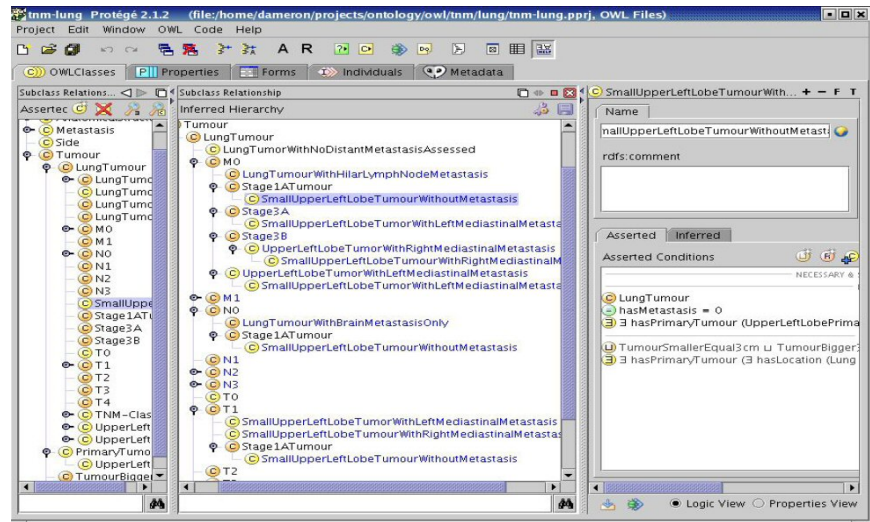


Figure 4. A screen Snapshot of BioPAX in the Protégé Ontology Development Environment. An Oracle RDF Data Model Plugin has been Developed for Protégé.

SQL/RDBMS, XQuery/XML and SPARQL/RDF offer three different ways to query and manage information. By using each of these technologies in different situations, a user can optimize the quality and efficiency of information querying and management. Oracle provides solutions for each of the three methods in order to meet all customer requirements.

DATA MODEL COMPARISON

SQL/RDBMS, XQuery/XML and SPARQL/RDF offer three different ways to query and manage information. Each of the methods serves different, complementary purposes. By using each of these technologies in different situations, a user can optimize the quality and efficiency of information querying and management.

A relational database and SQL are best where concise, efficient transactions are needed. Typically, this occurs within an enterprise application where the user is interacting with the data through a tightly constrained set of forms provided by the application. Given the tightly controlled environment, the application (and the underlying RDBMS) needs a minimal amount of input (e.g. a string, a number, a date) to execute properly. This is because all the metadata about the transaction is embedded or implicit in the application or database schema itself. The benefits of SQL/RDBMS are the low overhead required to execute a transaction and, therefore, the performance and scalability with a known level of quality of service that can be achieved.

However, when executing a transaction across organizational boundaries, the environment is much less tightly controlled. A supplier or customer may use a different application and a different database schema for the same type of transaction. In that case, SQL is at least very difficult to use. For this environment, XQuery/XML combined with Web services is more appropriate, which is why Oracle's products were enhanced to support this technology. XML documents can be used to execute transactions just as with SQL except that XML wraps the metadata about the transaction around the data itself. When an XML document is

sent from one organization to another, an agreed upon schema can be used to decode the metadata about the transaction. This is feasible when you have a well-structured federation of organizations as, for example, in a supply chain. XQuery/XML is not as efficient as SQL/RDBMS but offers much richer transactions and more flexibility for information sharing across applications.

But even XQuery/XML requires some agreement among parties as to the format of documents. Users must know ahead of time how, approximately, the information will be used. In many cases, it is impossible to know who will be looking for information, how they may choose to use it, and how it may be re-used at a later point. SPARQL/RDF is designed for information sharing with ultimate flexibility. By encoding the relationships between data, RDF enables semantics as well as syntax to be embedded in documents. Users can apply arbitrary ontologies to the data and semantics to discover information that may not have even been anticipated by the original data provider. Users with little or no technical knowledge of where the data is located or how it is structured can also formulate queries. This can be particularly powerful for applications on enterprise grids.

Each of the different information management models has distinct strengths. Oracle provides solutions for each of the three methods, in order to meet all customer requirements.

ORACLE SPATIAL RDF DATA MODEL

In Oracle Database 10g Release 2, a new data model has been developed for storing RDF and OWL data. This functionality builds on the recent Oracle Spatial Network Data Model (NDM), which is the Oracle solution for managing graphs within the Oracle Database. The RDF Data Model supports three types of database objects: model (RDF graph consisting of a set of triples), rulebase (set of rules), and rule index (entailed RDF graph).

Model

There is one universe for all RDF data stored in the database. All RDF triples are parsed and stored in the system as entries in tables under the MDSYS schema (figure 5). An RDF triple (subject, predicate, object) is treated as one database object. A single RDF document that contains multiple triples will, therefore, result in many database objects.

RDF_MODEL\$ is a system level table created to store information on all of the RDF and OWL models in the database. When a new RDF Model is created, a new MODEL_ID is automatically generated, and an entry is made into the model table.

The RDF_NODE\$ table stores the VALUE_ID for text values that participate in subjects or objects of statements. The NODE_ID is the same as the VALUE_ID. NODE_ID values are stored once, regardless of the number of subjects or objects they participate in. The node table allows RDF data to be exposed to all of the analytical functions and APIs available in the core NDM.

“The entry of Oracle into the Semantic Web space has already made a big splash, and rightly so. This isn't a big-name player passing off some veneer over an old product as something new; this is a genuine new capability, done well.”

**—Dean Allemang,
Chief Semantic Technology Consultant,
TopQuadrant**

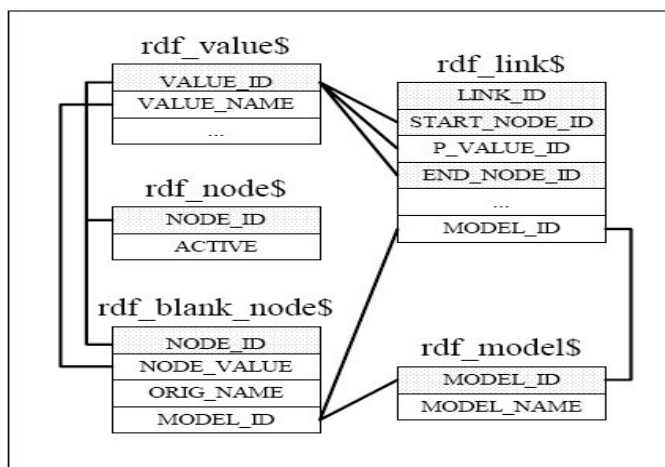


Figure 5. RDF_VALUE\$, RDF_NODE\$, RDF_LINK\$ and RDF_MODEL\$ are the Key Tables for RDF Storage in the Oracle RDF Data Model.

"As an Oracle partner, I am very excited about Oracle Database 10g Release 2. The new RDF capabilities in Oracle Spatial bring enterprise-class scalability and performance to graph databases. Cerebra will work with Oracle to apply these strengths for joint customers using semantic technologies to enable the real-time adaptive enterprise."

**—Jeff Pollock,
VP Technology, Cerebra**

The LINKS\$ table stores the triples for all of the RDF models in the database. Therefore, the MODEL_ID logically partitions the RDF_LINK\$ table. Selecting all of the links for a specified MODEL_ID returns the RDF network for that particular model.

The RDF_VALUE\$ table stores the text values, i.e. the Uniform Resource Identifiers or literals for each part of the triple. Each text value is stored only once, and a unique VALUE_ID is generated for the text entry. URIs, blank nodes, plain literals and typed literals are all possible VALUE_TYPE entries.

Blank nodes are used to represent unknown objects, and when the relationship between a subject node and an object node is n-ary. New blank nodes are automatically generated whenever blank nodes are encountered in triples. However, it is possible for users to re-use blank nodes, for example when inserting data into a containers or collections. The RDF_BLANK_NODE\$ table stores the original names of blank nodes that are to be reused when encountered in triples.

To represent a reified statement a resource is created using the LINK_ID of the triple. The resource can then be used as the subject or object of a statement. To process a reification statement, a triple is first entered with the reified statement's resource as subject, rdf:type as property and rdf:Statement as object. A triple is then entered for each assertion about the reified statement. However, each reified statement will have only one rdf:type to rdf:Statement associated with it, despite the number of assertions made using this resource.

The Oracle RDF Data Model supports containers and collections. A container or collection will have a rdf:type to rdf:container_name or rdf:collection_name associated with it, and a LINK_TYPE of RDF_MEMBER.

“The future of information navigation will enable enterprises to access both structured and unstructured data seamlessly to find the exact information they need. Oracle’s support of the RDF Data Model in its new 10g software is a major step toward this vision. Combined with Siderean’s Seamark Navigation Server, customers can now deliver a new generation of applications, where users navigate uniformly through all digital information, leveraging the inter-relationships of content to pinpoint results.”

**—Bradley Allen,
CTO and Founder, Siderean Software**

Two new object types have been defined for RDF-modeled data.

SDO_RDF_TRIPLE serves as the triple representation of RDF data, whilst SDO_RDF_TRIPLE_S is defined to store persistent data in the database. The GET_RDF_TRIPLE() function can be used to return an SDO_RDF_TRIPLE type.

Rulebase

Each RDF rulebase consists of a set of rules. Each rule is identified by a name, and consists of an ‘IF’ side pattern for the antecedents, an optional filter condition that further restricts the subgraphs, and a ‘THEN’ side pattern for the consequents.

A rule when applied to an RDF model may yield additional triples. An RDF model augmented with a rulebase is equivalent to the original set of triples plus the triples inferred by applying the rulebase to the model. Rules in a rulebase may be applied to the rulebase itself to generate additional triples.

Oracle supplies both an RDF rulebase that implements the RDF entailment rules, and an RDF Schema (RDFS) rulebase that implements the RDFS entailment rules. Both rulebases are automatically created when RDF support is added to the database. It is also possible to create a user-defined rulebase for additional specialized inferencing capabilities.

For each rulebase, a system table is created to hold rules in the rulebase, along with a system view of the rulebase. The view is used to insert, delete and modify rules in the rulebase. Information about all rulebases is maintained in the rulebase information view.

Rules Index

A rules index is an object containing pre-computed triples that can be inferred from applying a specified set of rulebases to a specified set of models. If a graph query refers to any rulebases, a rule index must exist for each rulebase-model combination in the query.

When a rule index is created, a view is also created of the RDF triples associated with the index under the MDSYS schema. This view is visible only to the owner of the rules index and to users with suitable privileges. Information about all rule indexes is maintained in the rule index information view. Information about all database objects, such as models and rulebases, related to rules indexes is maintained in the Rule Index Datasets view.

Query

Use of the SDO_RDF_MATCH table function allows a graph query to be embedded in a SQL query. It has the ability to search for an arbitrary pattern against the RDF data, including inferencing, based on RDF, RDFS, and user-defined rules. It can automatically resolve multiple representations of the same point in value space (e.g. “10” ^^xsd:Integer from “10” ^^xsd:PositiveInteger).

The SDO_RDF_MATCH function has been designed to meet most of the requirements identified by W3C in SPARQL for graph querying.

A Java API is also provided for network representation and network analysis. Analysis capabilities include the ability to find a path between two resources, or to find a path between two resources when the links are of a specified type.

Advantages of the Semantic Web include the ability to integrate heterogeneous data through common explicit semantics, the expression of rich and well-defined models of systems, the formal annotation of findings and interpretations, the ability to embed models and semantics directly within online publications, the application of logic to infer new insights, the ability to search based on term meaning, and it enables data to be machine-processable.

CONCLUSION

Effective, large-scale, data integration has been difficult for many companies to achieve. This is largely attributed to the fact that it's difficult for companies to adapt their data models to changing schema and new data sources, and to integrate many data types into a single warehouse. Increasingly people are becoming concerned that their data warehouses may not be as valuable as they could be due to incorrect data integration as a result of acronym collision, synonyms and homonyms. If the data has not been optimally integrated, then businesses will not gain maximum value from any investment in business intelligence.

Many customers are seeing Semantic Web technologies as a way to overcome these technology hurdles. The flexible structure of RDF alleviates the burden of needing to build a single large data model. It can also be used to tag any Web compatible data format. In addition, it offers the ability for people to clearly define what it is that they are referring to with the use of URIs. Companies are very interested in this capability, as it will significantly increase the quality of the data within the warehouse, thereby allowing far higher quality data analysis.

There are many advantages of the Semantic Web that appeal to organizations in the life sciences. These include the ability to integrate heterogeneous data through common explicit semantics, the expression of rich and well-defined models of systems, the formal annotation of findings and interpretations, the ability to embed models and semantics directly within online publications, the application of logic to infer new insights, the ability to search based on term meaning, and it enables data to be machine-processable.

Oracle now provides solutions to enable organizations to manage data using SQL/RDBMS, XQuery/XML and SPARQL/RDF. Allowing users to select the most appropriate method for managing their data.



Semantic Data Integration in the Life Sciences
September 2005
Author: Susie Stephens

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2005, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.