UMRR: Towards an Enterprise-Wide Web of Models

Jing Mei[1], Guotong Xie[1], Lei Zhang[1], Shengping Liu[1], Bob Schloss[2], Yue Pan[1], Yuan Ni[1]

[1]IBM China Research Laboratory
Zhongguancun Software Park, Beijing, 100193, China

{meijing, xieguot, lzhangl, liusp, panyue, niyuan}@cn.ibm.com

[2]IBM Watson Research Center
P.O.Box 704, Yorktown Heights, NY 10598, USA

rschloss@us.ibm.com

ABSTRACT

Metadata that describes the structure and semantics of data sources takes a significant role in enterprise information integration. Enterprise information integration always involves an increasing set of types of metadata that are dispersed in various repositories, modeled by various tools, represented in various formats. There is a crucial requirement to break the "Tower of Babel" among different types of metadata to enable better understanding, more comprehensive governance and analysis. Towards the goal of a simplified framework for metadata representation, federation, search and analysis, in the UMRR (Unified Metadata Registry and Repository) project at IBM, we propose and implement an open Web architecture for universal metadata management, where the Resource Description Framework (RDF) is adopted to represent the underlying metadata that are in various formats and the "Linked Data" method is leveraged to build a web of models. Our demonstration illustrates the effectiveness of this architecture to enable the metadata federation such that the global query, search and analysis on the metadata are feasible. Additionally, we also demonstrate that the proposed architecture could easily leverage Web 2.0 technologies, such as social bookmaking, tagging and RSS feeds, etc. for collaborative metadata management.

1. INTRODUCTION

There are many kinds of metadata in Enterprise Information Integration and many different ways to classify them. On the basis of the intended users, metadata can be classified into two categories: technical metadata and business metadata [2]. In this proposal, we also refer to these metadata as models and use the two terms in an interchangeable way. These models are used by different roles, e.g. Business Analyst, Data Architect, Data Analyst, Database Administrator and Database Developer etc. These models are themselves produced by various tools and stored in distributed metadata repositories. Considering the diversity of tools from different vendors and the lack of standards beyond SQL, the model management issue becomes very challenging.

Before listing the real challenges, let's take a look at Figure 1 which shows a typical information integration scenario where enterprise-wide model management capability is required. An insurance company wants to transform its IT infrastructure to the SOA architecture by publishing information services, which offer the key business data of the company, e.g. claim and customer, to internal and external systems via Web Services interfaces and insurance data standards like ACORD (http://www.acord.org/).
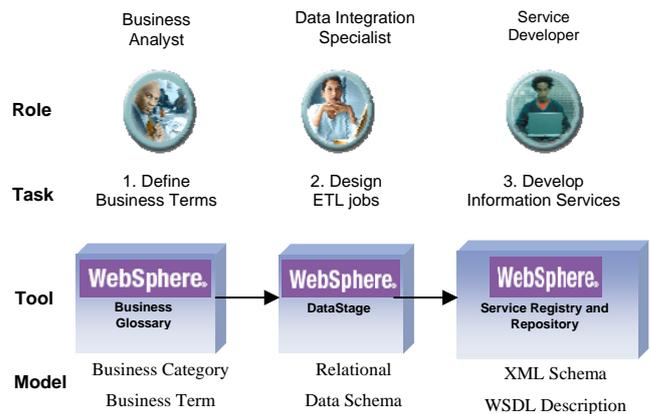

Figure 1. Enterprise-wide model management scenario in enterprise information integration

In a simplified IBM solution to this scenario, there are three roles. Business Analysts use WebSphere Business Glossary (WBG, http://www-306.ibm.com/software/data/integration/business_glossary/) to organize business glossaries in the insurance domain and manage the relationships among business terms and technical metadata like relational data schemas or WSDL service descriptions, to simplify the integration task by leveraging better business-IT alignment. Data Integration Specialists use WebSphere DataStage (http://www-306.ibm.com/software/data/integration/datastage/) to develop ETL jobs to extract data from operational systems, clean them and generate the integrated business data. Service Developers use WebSphere Service Registry and Repository (WSRR, http://www-306.ibm.com/software/integration/wsrr/) to develop information services and publish their WSDL descriptions. Please note that in real situations, insurance companies may use tools from different vendors, e.g. ETL tool from Informatica (http://www.informatica.com/) or service registry from Sun (http://www.sun.com/products/soa/registry/).

There are several metadata management issues in the scenario. Firstly, it is unable to represent or record cross-repository linkage. For example, a business analyst could not "classify" a WSDL service, *CustomerInformationService*, in the remote WSRR server by using the business term *Customer_ID* in WBG, since the WSDL service is managed by a remote server registry and the metamodel and access method of the service registry might be unknown to WBG. It is especially so when the service registry is developed by a different vendor than the vendor which built WBG.

Secondly, it is difficult to perform global search and global query. For example, a service developer who intends to implement the *CustomerInformationService* may want to use the keyword "customer" to retrieve all related metadata information such as relational table *DW_CUSTOMER* that stores customer profile data, ETL job *Job4_DS_Create_Customer* that creates a file for the integrated customer data, and the WSDL service *Customer-ScoringService* that returns customer rating information. If the service developer has more knowledge about metadata types or structures, he may want to find more accurate results by issuing a structure query, e.g. "show me all the relational tables that are relevant to 'customer', and each table MUST have at least one column that is relevant to *ID*". The issue here is that different tools are offering different search capabilities, query languages and metamodels, which makes it more complicated to do global search and query.

Thirdly, it is difficult to do global analysis. For example, suppose a data integration specialist wants to change the definition of an ETL job *Job3_QS_Match*. Before changing the definition, he needs to know which models (elements) are "dependent" on this ETL job, e.g. *Job4_DS_Create_Customer* that is the next stage in the whole ETL process, *DW_CUSTOMER* that is populated with data published by *Job4_DS_Create_Customer*, and *Customer-InformationService* that offers services by exporting data from *DW_CUSTOMER*. The issue here is that metadata repositories are offering "silo" metadata in different metamodels and definitions of "dependent" vary with the types and structures of the metadata involved which makes it hard to do global analysis.

Finally, the data integration specialist may want to "bookmark" all ETL jobs he cares and give them a few tags for future lookup, or "subscribe" to all changes of relational tables in a database. The issue here is how to leverage existing Web 2.0 technologies, e.g. del.icio.us and RSS feed readers etc., to avoid reinventing the wheel and forcing people to switch among different tools.

Towards the goal of a simplified framework to address the above issues, we propose "an enterprise-wide Web of models" for metadata management, which is based on the "Semantic Web" infrastructure and the "Linked Data" method. By transforming existing metadata repositories to Linked Data servers, enterprise-wide models can be managed autonomously but accessed uniformly via lightweight protocols like HTTP and RESTful SPARQL queries. It eliminates the barrier imposed by different representations, formats and access protocols of multiple silo model repositories across the enterprise. Then metadata search engine can collect all metadata in the enterprise from Linked Data interfaces, and build search indices for the keyword search and structural query. Eventually, semantic reasoning can be performed over collected metadata for comprehensive metadata analysis.

2. ENTERPRISE-WIDE WEB OF MODELS

The models and their links inspired us to consider the Web architecture, which has been proven to be scalable and robust to handle heterogeneous inter-related data that are evolving continuously and autonomously. However, unlike web pages, the models to be represented are typically structured data, and structural query capability is desired. Fortunately, we have Semantic Web and recent Linked Data technologies to represent, access, link, and query structured data on the web [1]. Our main idea is illustrated in Figure 2.
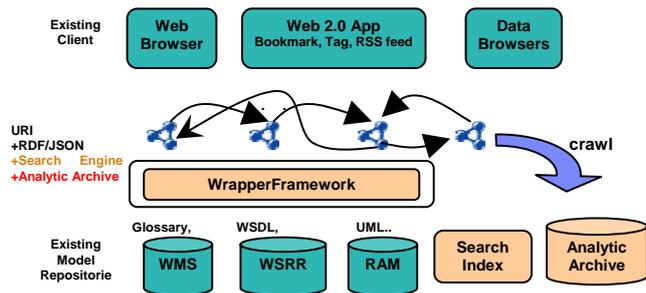

Figure 2. UMRR Prototype Architecture

In a wrapper framework, existing model repositories are published as Linked Data Servers, where URIs are assigned to models and fine-grained model elements. As shown in Figure 2, there are three running prototypes, one for WBG, one for WSRR, and the other for RAM (Rational Asset Management). Both WBG and WSRR are wrapped by using the D2R-like solution to publish relational databases, while RAM is accessible by API. On top of wrapped repositories, model (element) URIs are dereferenceable via HTTP GET requests, and an RDF description for a given URI will be returned. The returned RDF description follows an OWL ontology defined for meta models, including links to model (element) URIs in remote servers.

The web of models also involves crawlers and search engines that collects RDF descriptions of all metadata from distributed model repositories via Linked Data interfaces or semantic sitemap. The collected RDF descriptions are stored in an analytic archive we developed on SOR [4]. Leveraging the powerful query and reasoning features of SOR, we can do global metadata query and analysis easily. The archive is also indexed by an RDF search engine we developed [3] to provide functions ranging from simple keywords-based to complex structural and semantics-based search.

Now, via dereferenceable URIs on Web of Models, existing Web 2.0 methods can be imposed to collaboratively manage the enterprise models. More interestingly, a SPARQL query is also a dereferenceable URI and we could bookmark an interesting query and share the query result with others. So far, we implemented a transformer that could convert a SPARQL query result to an RSS feed. Thus, users can monitor a specific query result to get notified once it changes using their favorite RSS readers.

Briefly, this Web of Models architecture integrates innovations from open Semantic Web standards, 3rd party components, and components developed by our team. Our contribution is, for the first time, solidifying the architecture and applies it to solve the acute metadata management problem in the enterprise. Our evaluations show that this is a very promising approach.

3. REFERENCES

[1] Chris Bizer, Richard Cyganiak and Tom Heath, How to Publish Linked Data on the Web (tutorial), July 2007.

[2] Ganesan Shankaranarayanan and Adir Even, The Metadata Enigma, CACM, 49(2), Feb. 2006.

[3] Lei Zhang, et., al., Semplore: An IR Approach to Scalable Hybrid Query of Semantic Web Data. Proc. of ISWC 2007.

[4] Li Ma, et., al., Effective and Efficient Semantic Web Data Management over DB2, to appear in Proc. of SIGMOD 2008.