Mediating the Semantic Web

Georges Gardarin, Tuyet-Tram Dang-Ngoc

PRiSM Laboratory
University of Versailles
45, avenue des Etats-Unis. 78035 Versailles CEDEX
FRANCE
georges@gardarin.org http://www.gardarin.org
dntt@prism.uvsq.fr http://www.prism.uvsq.fr/users/dntt/

Résumé. Cet article développe une extension d'une architecture de médiation pour intégrer le Web sémantique. Plus précisément, XLive est un médiateur tout XML développé à PRiSM. Il permet d'exécuter des XQuery sur des sources de données hétérogènes. Après une rapide présentation de XLive et du Web sémantique, une architecture à trois niveaux d'ontologies et de schémas est introduite pour connecter des adaptateurs pour le Web sémantique. Cette architecture vise à intégrer des sources de type Web service d'information conformément à une ontologie globale de référence. Elle conduit à étendre XLive avec le support de vues, un outil de conception de vues et de mappings, et des adaptateurs pour les Web services.

1. Introduction

Typical information integration systems have adopted a wrapper-mediator architecture [Wiederhold, 1992]. In this architecture, mediators provide a uniform user interface to query integrated views of heterogeneous information sources. Wrappers provide local views of data sources in a global data model. The local views can be queried in a limited way according to wrapper capabilities. While in the 90's most studies were based on using the object model as data integration model, the focus has come to XML as global model at the beginning of the new century. Mediator architects are used to distinguish the local as view (LAV) approach versus the global as views (GAV) approach, in which the integrated views are designed in terms of the local views of sources. To meet a given ontology in a given domain, the LAV approach seems more appropriate, but the GAV approach makes easier to take into account existing data sources schemas. Thus, mixed approaches are possible supported by schema design tools [Haas, 1999]. Well-known research projects and prototypes in mediation include Garlic [Haas, 1999], Tsimmis [Garcia-Molina, 1997], Tukwila [Ives, 1999].], Artemis [Castano, 2000], Enosys EXIP[Papakonstantinou, 2003], and XLive [Dang-Ngoc, 2003], a descendant of e-XML mediator [Gardarin, 2002].

Tim Berners-Lee, the inventor of the WWW, thought up the Semantic Web. The goal is to give well-defined meaning to Web information, better enabling intelligent Web applications requiring searching or reasoning, make them exchange meaning rather than information. There are a lot of (sometimes confusing) activities at W3C along this line. The most prominent results are RDF, RDF Schema (RDFS), and OWL (Web Ontology Language). RDF provides a semantic data model based on triple <resource, property, value>

to describe Web resources [RDF, 2002]. RDFS brings a data typing model for RDF. Using RDF Schema, the users can specify types of resources, create properties and classes with sub-classes, as well as defining ranges and domains for properties. OWL is a language to develop ontologies [OWL, 2003]. An ontology defines a common vocabulary to share information in a domain. It includes formal definitions of concepts in the domain and relations among them. In practical terms, OWL includes constructs to define classes, subclass–superclass hierarchies, properties and values plus constructs to define constraints and rules between classes and properties.

The problem addressed in this paper is how to query the semantic Web using a mediation architecture. In the XLive project, we have developed a full XML mediator to query semi-structured data sources including relational databases mapped in XML and XML files stored in Xyleme, a warehousing system for XML [Abiteboul, 1999]. We are currently extending the XLive mediator to query and integrate Web services. We assume that each site provides a Web service API to retrieve Web pages. First, using an extractor we built a local view of the site as a collection of XML documents that can be queried on certain elements. Then, based on several layers (in general, 2) of ontologies, we define integrated views of sources in a given domain (e.g., tourism). Accordingly to the classification of [Wache, 2001], we implement a hybrid approach where the semantics of each source is described by its own ontology and the semantics of integrated data is defined by some domain ontology. Each ontology is associated to an XML view defined in XQuery of the extracted data.

This paper is organized as follows. In section 2, we recall some backgrounds on XQuery and XML mediation, then, we describe the XLive mediator. We focus on the XLive architecture and metadata management. In section 3, we recall some backgrounds on the semantic Web, introducing OWL and Web Services, which are strong foundations for the XLive project. In section 4, we introduce the XLive extensions for querying the semantic Web. These are mainly the introduction of ontological views, the development of Web service wrappers, and a design toolkit for mapping views to ontologies and vice versa.

2. XML Mediation

Mediation technology based on XML and XQuery is under development. Some products are already available. We survey this new technology and describe our XLive mediator (see www.xquark.org for an industrial open source version).

2.1 Basics and Backgrounds

With the advent of XQuery as a standard for querying XML collections [XQuery, 2003], several mediator systems have been developed using XQuery and XML schema as pivot language and model. Examples of full XML mediators are the Enosys XML Integration Platform (EXIP), the Software A.G. EntireX XML Mediator, the Liquid Data mediator of BEA derived from EXIP [Papakonstantinou, 2003], the e-XMLMedia XML Mediator, a predecessor of our current XLive project [Gardarin, 2002]. XML Mediators are focusing on supporting the XQuery query language on XML views of heterogeneous data sources. The data are integrated dynamically from multiple information sources. Queries are used as view

definitions. During run-time, the application issues XML queries against the views. Queries and views are translated into some XML algebra and are combined into single algebra query plans. Sub-queries are sent to local wrappers that process them locally and return XML results. Finally, the global query processor evaluates the result, using appropriate integration and reconstruction algorithms.

XQuery is a powerful language, which encompasses SQL and much more. Notably, it is able to query rich and extensible data types; it is a functional language, so that any valid expression applied to a valid expression is a valid query; it will soon incorporate XQuery Text for full text queries. XQuery Text shall provide functionalities as single-word search, phrase search, support for stop words, search on prefix, postfix, infix, proximity searching, word normalization, diacritics, ranking and relevance. All that features will make XQuery an ideal language for querying the semantics Web.

XQuery mediators are interesting components to perform semantic integration of Web information. Web mediators aim at transferring Web information from different Web sites in the same domain (e.g., tourism) to a high level domain-specific common information model, making document semantic machine understandable. Web mediators should integrate three kinds of advanced technologies: ontology based description of Web sources, Web data extraction and mapping, and distributed query processing based on mediators.

2.2 Overview of the XLive Mediator

In the XLive project, we use a mediation architecture to support web information integration and semantic search as shown in Figure 1. It follows the classical wrapper-mediator architecture as defined in [Wiederhold, 1992]. The communication between wrappers and mediator follows a common interface, which is defined by an applicative Java or Web service interface named XML/DBC. With XML/DBC, requests are defined in XQuery and results are returned in XML format.
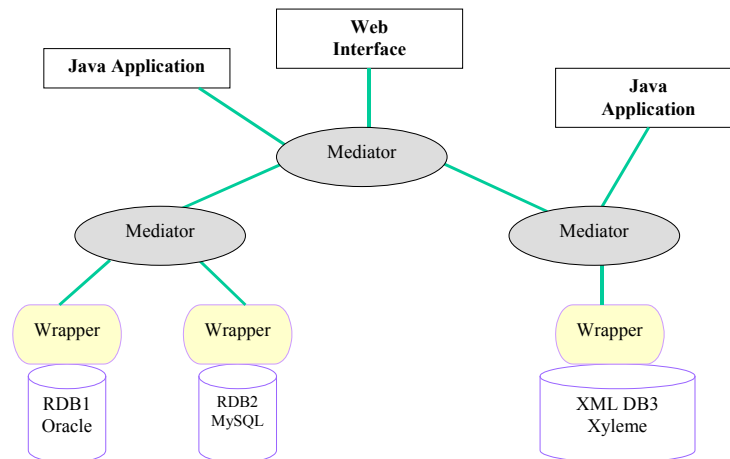

FIG. 1 - XLive Architecture

Our architecture is composed of mediators that deal with distributed XML sources and wrappers that cope with the heterogeneity of the sources (DBMS, Web pages, etc.). The XLive mediator is a data integration middleware managing XML views of heterogeneous data sources. Using XLive mediator one can integrate heterogeneous data sources without replicating their data while the sources remain autonomous. XLive mediator is entirely based on W3C standard technology: XML, XQuery, XML-Schema, SAX, DOM and SOAP. All information exchanges rely on XML format. XML-Schema is used for metadata representation. Wrappers provide schemas to export information about local data structures. XQuery is employed for querying both the mediator and the wrappers. Connectivity of mediator and wrappers relies on the XML/DBC programming interface, an extension of JDBC to integrate XQuery. More information about the XLive mediator can be found in [Dang-Ngoc, 2003].

To integrate a new source into the mediation architecture, a wrapper must be built. It has to implement the XML/DBC programming interface, process some XQuery requests, and return results in XML format. DBMS are data oriented sources and metadata are provided to describe sources and mappings. DBMS wrappers translate data sources in XML and process a possibly reduced set of XQuery on the source data. In the case of Web source, the wrapper brings more intelligence. It aims at semantically integrating Web information in a common model accessible to programs.

2.3 Metadata in the XLive Mediator

The XLive mediator maintains metadata in a simple way. Each time a source connects to a mediator, it provides the structural tree of the XML documents it is able to retrieve. More precisely, a mediator maintains the set of XML paths that can be queried and retrieved from each wrapper it is connected to. It is called the path set and can be seen as a weak schema or a summarizing DTD of the source. Query processing uses the path set to check query validity and to determine the relevant data sources for a query. The path set is also used to expand incomplete paths. Skeletons of schemas can also be displayed to the final user in order to help him to formulate queries. In the sequel, we are going to discuss how to extend these metadata to meet ontologies.

3. Semantic Web

In this section, we present some recalls on the Semantic Web. We also discuss the link with Web Services.

3.1 Basics and Backgrounds

Let us start with a definition from W3C: « The Semantic Web is the representation of data on the World Wide Web. It is a collaborative effort led by W3C with participation from a large number of researchers and industrial partners. » It is based on the Resource Description Framework [RDF, 2003], which integrates a variety of applications using XML for syntax and URIs for naming. RDF descriptions express semantic metadata over document

contents (Web pages, etc.) using triplets <Resource, Property, Value>. Resources located by URIs are described by properties and values, a value being a resource or a literal.

RDF is used to annotate documents with semantic descriptions. Descriptions are expressed in terms defined in ontologies. An ontology is a definition of the concepts and relationships that exist for a domain. It is generally expressed in terms of class hierarchies, literal or class valued properties, with formal rules linking the defined classes and properties (e.g, class equivalence, property transitivity, etc.) [OWL, 2003]. Ontologies are definitions of metadata about data (e.g., Web pages) that can be used for reasoning. RDF schema is a first level language to write definition of metadata in the RDF Vocabulary Description Language proposed by W3C. When RDF metadata are based on an RDF schema, RDF properties should reference the base RDF schema through a namespace. RDF schema does not provide reasoning capabilities. In addition, an ontology includes reasoning capabilities about things that may help in understanding what computers are dealing with when processing information in search engine, query systems, mediation systems, intelligent agents, etc.

3.2 OWL to Define Ontologies

OWL (Ontology Web language) is a language to define ontologies on the Web based on RDF schema. It extends the basic constructs of RDF schema to provide more interoperability (e.g., equivalences), more reasoning facilities (e.g., description logic), and more evolution support (e.g., integration and version). It is inspired from DAML (DARPA) and OIL (EEC). OWL incorporates the basic features of RDFS (RDF Schema): definitions of class, property, domain, range, sub-class and sub-property. A property relates a class to a class or a literal. The XML schema data types are supported for typing literal properties. Individual can also be defined as member of a class. In addition, OWL provides constructs to define relationships between classes such as equivalence and set expressions (e.g., A is B union C minus D). Properties can also be qualified as functional, symmetric, or transitive; two properties can be declared as inverse. Cardinalities are extended and restrictions can be imposed to property values. Finally, versioning information and annotations (e.g., comments) can be defined for ontologies.

3.3 Place of Web Services

Web services are applications whose logic and functions are published on the Web in UDDI registries and accessed through standard XML messages encoded in SOAP. A further step in the development of software components (previous steps have been object component development as EJB or Active X), Web Services represents abstract functions that can be reused without worrying about how the services are implemented. The interface of a Web Service is defined strictly in terms of messages the service accepts and produces. Applications using Web Services can be implemented on any platform in any programming language, as long as they can create and consume messages defined for the service interface. A Web Service can also be a composition of other services to provide higher-level functionalities. Language for composing and orchestrating Web Services are on their way towards standardization (for example, BPML, BPEL, BPSS).

Web Services and the Semantic Web are complementary. Web Services aim to standardize and automate the protocol to transmit requests and answers to active Web resources, whereas the semantic Web aims to standardize and automate semantic descriptions of Web resources for information integration and reasoning. The Semantic Web addresses the semantic description of the contents while the Web Services addresses the syntactic exchange of the contents. Integrating Web services and the Semantic Web in a mediating architecture is another issue discussed below.

4. Mediating the Semantic Web

Several architectures of mediators have been proposed to query the Semantic Web. We propose an hybrid architecture with ontologies defining vocabularies of schemas describing information. This architecture is currently being implemented both as a schema design tool and runtime view support in the XLive project.

4.1 Hybrid Mapping Architecture Overview

Most Web mediators implement the LAV approach to design local source schemas as views of some pre-existing global schema often called ontology. There is often some confusions between a schema and an ontology. In our view, an ontology define concepts and relationships between concepts. These concepts are used to annotate information and define conceptual view of information. A (XML) schema defines the structures and types of document collections. One of the problems for mediating the Web is to map (generally poor) schemas of documents to (generally rich) ontology and vice versa.

Most mediators work with schemas of collections, not with ontological descriptions of information. The latter superimpose conceptual schemas and constraints on information using ontology-based annotations. To integrate the Semantic Web in our mediator architecture, we propose an hybrid mapping scheme, where:
(i)   Local sources are mapped to one or more local ontologies using a GAV approach; local annotations elaborated in conformance to the source ontologies can be used to support the mapping.
(ii) Domain ontologies are mapped to local ontologies using a LAV approach; annotations of local sources can also be added to help the mapping process.
We clearly distinguish schema describing view of information and ontologies giving set of concepts to annotate information and helping in building view. We define all mappings in XQuery, which is a powerful language to map information and annotations in an integrated way. This is feasible as everything is XML and can be queried based-on XQuery, possibly with specific functions as allowed in XQuery.

Figure 2 gives an overview of the proposed schema and ontology architecture at the mediator layer. A global ontology is defined in OWL. A global ontological view corresponds to an integrated schema; it defines a logical collection in terms (i.e, with a mark up) defined in the global ontology. Semantic annotations can be added to the local view in terms of the global ontology. Local views are derived from global views using XQuery following the

LAV definition scheme. Mapping from local ontological view instances to global ontological views is through an XQuery computed at design time as the inverse of the view definitions.
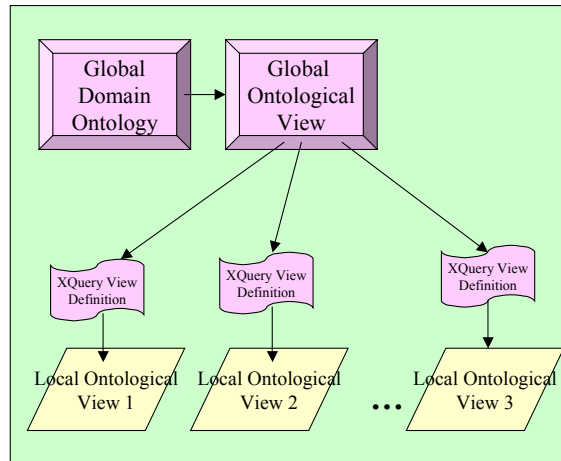

FIG. 2 – View and Ontology at mediator layer.

Figure 3 gives an overview of the proposed schema and ontology architecture at the wrapper layer. Local ontologies are also defined in OWL. A local ontological view is defined by a path set and a query ; it corresponds to a logical collection expressed in terms defined in the local ontology. Semantic annotations can be added to a Web site in terms of the local ontology. Local ontological view instances are derived from local logical views and annotations using XQuery. Local logical views instances are derived from Web Services calls or local source queries (e.g., SQL queries).
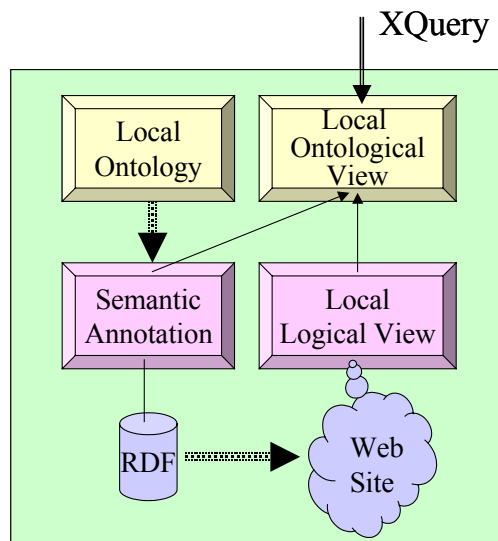

*FIG. 3 – View and Ontology at wrapper layer.*

4.2 Wrapping Web Services as Local Logical Views

   Web sites become more and more open to Web services. Good examples are Amazon.com and Google. They both provide a Web service API to query their catalogs. While the Amazon catalog describes products, Google catalog describes the World Wide Web sites. With Google, the main Web service operation is the search request. It submits a query string and a set of parameters to the Google Web APIs service and receives in return a set of search results (see Figure 4). Search results are XML documents derived from Google's index of over 2 billion Web pages. They are composed of various elements, the most important being an array of <ResultElement>'s, each item giving a summary, a URL, a snippet, a title, a size, the host name and some others information (see Figure 5).
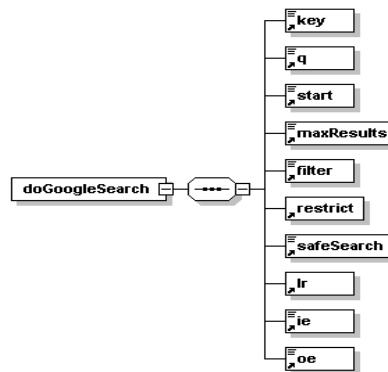
FIG. 4 - Google query path set.
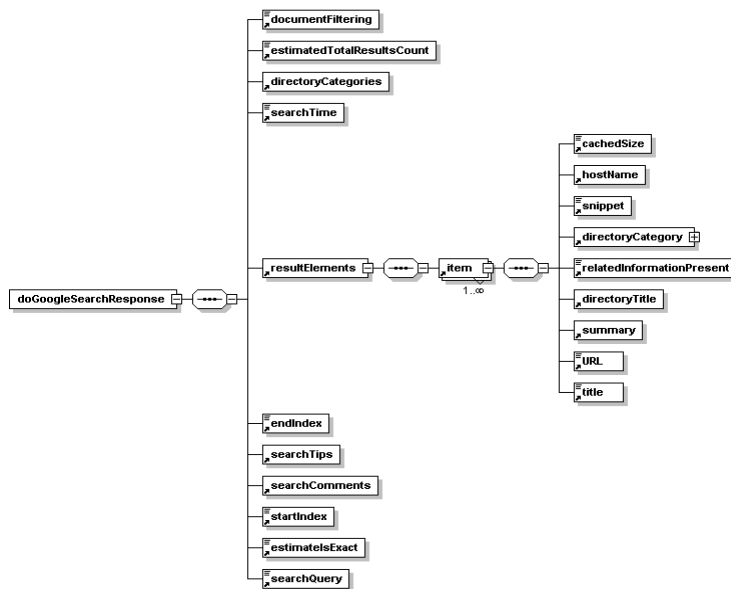
FIG. 5 - Google response path set.

To wrap Web services to an XQuery mediator, it is necessary to define a representation of the Web service as one or more collections of XML documents. The path set of the collection(s) should cover both the query string and the search results. For simplicity, we propose to define one collection per service operation modelling the invocation and response messages. A collection derived from an operation holds a path set covering the DTDs of all documents that can be queried and retrieved using the operation. For example Google search could be seen as a collection of path sets portrayed in Figure 4 and 5. Queries could then be submitted involving the collection using XQuery with text extensions, as foreseen in XQuery Text. We call that view of a Web service the **local logical view**, defined as follows:

Definition: Local Logical View
Representation of a Web service function as a partially queryable collection whose path set is the set of paths defined in the function call and return messages.

The collection schema is derived from the WSDL service description. All queries are not feasible on the local logical view. Capabilities should be associated to the wrapper to define what queries are possible. For example, with Google search queries must instantiate the q element with some text search expressions to be valid. Other elements can be given or generated by default.

In summary, information searching Web services can be modelled as queryable collections corresponding to operation messages. This simple mapping makes possible the translation of search XQuery in Web service invocation. The mapping defines the virtual view of a Web service as a collection of XML virtual documents. The translation of XQueries involving virtual collections in Web service invocation has to be developed on a per case basis, depending on the Web service semantics.

4.2 Mapping Web Service Views to Ontologies

As described above, any Web service can be wrapped as an XML collection that supports some more or less general XML queries. This provides a simple way to map Web site to structural views defined as XML collections with associated path set. In general, for each site, local logical views can be built that resembles the structure of the information source possibly encapsulated by Web services. It provides a local XML schema to represent the source defined in local terms. If it helps to extract data from the source using XQuery, it does not help much to integrate the information of various sources with more or less different semantics. Thus, further mappings are required to provide an integrated view of the information.

Several working groups are currently defining ontologies in various domains, as wine, travel, tourism, health, sciences, and manufacturing.

Definition: Global Ontology
Definition of classes with properties developed by specialists of a given domain, specifying a vocabulary of terms with their relationships.

Unfortunately, existing Web sites does not follow these global ontologies and have their specific models. However, to be able to integrate information coming from various sources of related topics, it is necessary to use a common ontology at some point. To make the source ontologies comparable to each other, global shared vocabularies have to be built. A shared vocabulary contains basic terms (the primitives) of a domain. Each site should implement its local view of the common ontology, that we call a **local ontology**.

Definition: Local Ontology
Projection on a local site of a global ontology, defining the global classes and properties understood at a local site.

Local ontologies are defined for example in OWL and used to generate RDF descriptions of the associated source. From a local ontology, an XML path set can be derived describing the source as a collection of XML documents expressed in terms of the ontology. This collection is called the **ontological view** of the source.

Definition: Local Ontological View
Representation of a local source as a set of partially queryable collections of documents whose path set is derived from the local ontology.

In general, an ontological view is built using an ontology vocabulary. Its path set is composed of elements defined in the ontology. Its schema should be derived from the ontology. This is also true for the integrated view, defined as follows:

Definition: Global Ontological View
Representation of the integrated sources as a set of partially queryable collections of documents whose path set is derived from the global ontology.

As local ontologies are derived from the global one by projection, it is natural to use a local as view approach to derive the local ontological view from the global one. Thus, we propose to define each local ontological view as one or more XQuery queries on the global ontological view. In general, we use XQuery to map the XML logical collections to a domain-related ontology, locally using a GAV approach, and globally using a LAV approach. This makes the mapping more flexible as clearly divided in two steps, logical to ontological (mostly corresponding to semantic integration) and global to local (mostly corresponding to localization integration). Views and ontologies are summarized in Figure 6.

4.3 Query Processing and Schema Design Tool

The XLive architecture is currently being extended to meet ontologies. XLive receives XQuery requests expressed on a simple integrated schema composed as the union of local views. XLive decomposes queries according to the source path sets. Local sub-queries on local views are then shipped to local systems with respect to source capabilities. To fully integrate the proposed ontology and view architecture, XLive has first to be extended with view support on the mediator. Second, each local wrapper should be able to translate XQuery expressed on local ontological views to queries on local logical views, which are finally translated to Web service calls. Figure 7 shows query processing in the extended XLive

architecture. Several optimization features are currently proposed to support efficiently the multilevel mappings.
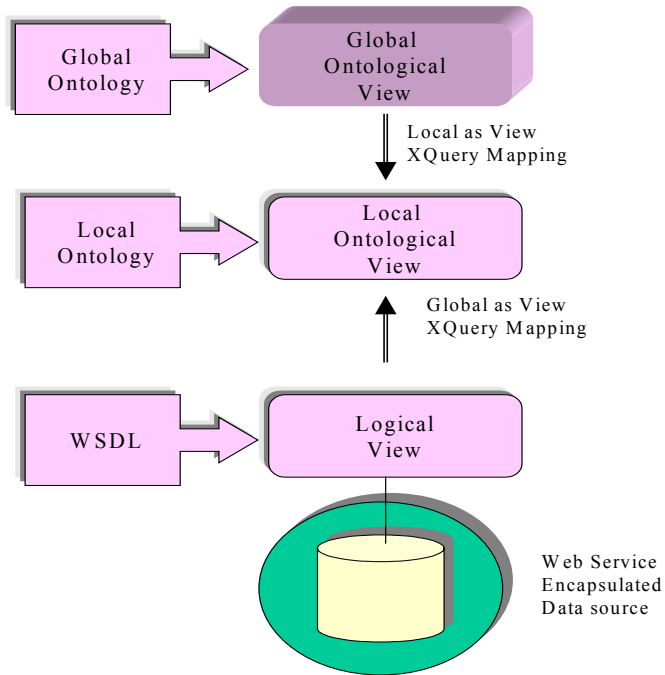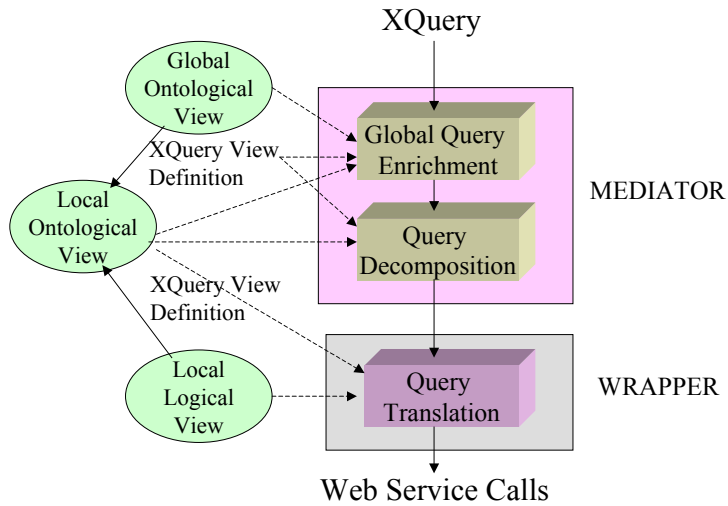

FIG. 6 – Ontologies and Views

FIG. 7 – Query Processing in Extended XLive.

A design tool should help in designing the views and specifying the mappings. The role of the design tool is to generate:
- Local logical view schemas.
- Local ontological view schemas.
- Global ontological view schemas.
- Mappings from global ontological views to local ontological views, then to local logical view.

The inputs of the design tool are the global ontology, the WSDL description of Web services, and semantic correspondence given by some administrator. Correspondence may be defined by simple identity mappings (using drag and drop), but more complex mappings are possible using libraries of operations developed in XQuery for the application or generic. Semantic annotations of local sources can be added to help in mapping development. Figure 8 gives an overview of the envisioned design tool. Notice that local and global designer components should collaborate in order to design the local ontological view as a compromise between the global ontological view and the logical view. We are currently designing more precisely our design tool.
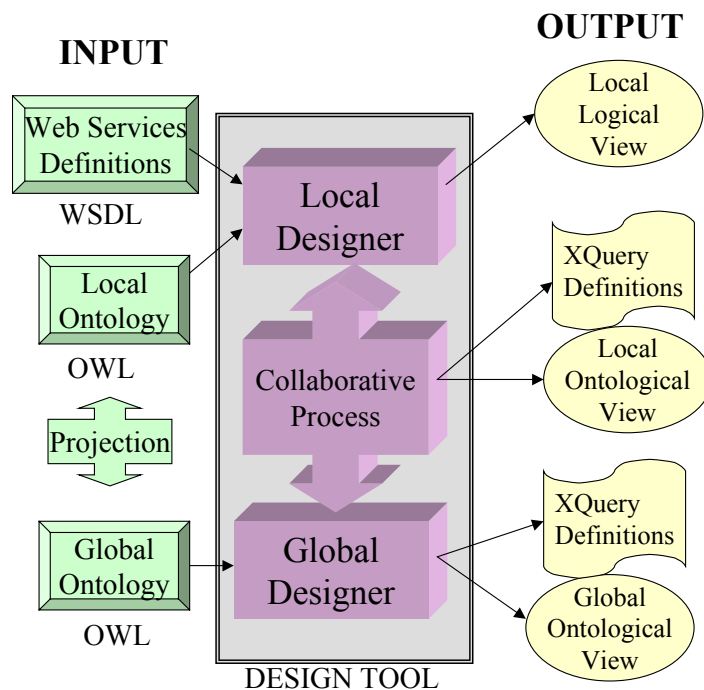
INPUT

OUTPUT

Web Services Definitions

WSDL

Local Ontology

OWL

Projection

Global Ontology

OWL

Local Designer

Collaborative Process

Global Designer

DESIGN TOOL

Local Logical View

XQuery Definitions

Local Ontological View

XQuery Definitions

Global Ontological View

FIG. 8 – Design Tool Overview

5. Conclusion

In this paper, we presented XLive, a mediation project developed at PRiSM in the context of the WebSI project. The mediator is currently operational to support relational or XML sources coupled through specific wrappers. We consider extending XLive to query the semantic Web using XQuery. We summarize the contribution of the semantic Web. Ontologies and description of Web site through annotations based on ontology have to be considered. Web services must also be integrated.

Extending XLive with ontology and Web service is not an easy task. It leads us to a hybrid architecture, mixing top down and bottom up schema and mapping design. More precisely, we introduced three levels of views of Web services as collections of documents, namely the local logical view, the local ontological view, and the global ontological views. We defined clearly these three layers and discuss query processing and mapping design tool.

Further research is necessary to be able to use properly ontology in mediator-based data integration. We believe that XQuery is an excellent language both for expressing mappings and formulating queries. Optimization techniques have to be developed to make the semantic Web mediator process efficiently all that mappings; fortunately, some mappings can be resolved at compile time. More optimization techniques are discussed in [Dang-Ngoc, 2003]. Finally, we would like to point out that more than two levels of mapping could be supported to meet a global ontology. This is possible thanks to the recursive architecture of XLive that makes possible to see a mediator as a local source. Intermediate mediator can progress towards the vocabulary of a given ontology, leaving upper layer mediator additional mappings to reach the desired ontology.

Références

[Abiteboul, 1999] Abiteboul S., Cluet S., Ferran G., Rousset M.C. The Xyleme project. Computer Networks 39(3): 225-238 (2002)Abiteboul S. On Views and XML. PODS 1999: 1-9

[Adali, 1996] Adali S., Selçuk K., Candan,, Papakonstantinou Y., Subrahmanian V. S. Query Caching and Optimization in Distributed Mediator Systems. SIGMOD Conference 1996: 137-148.

[Castano, 2000] S. Castano, V. De Antonellis, S. De Capitani di Vimercati, M. Melchiori: An XML-based Framework for Information Integration over the Web, Proc. of the 2nd International Workshop on Information and Web-based Application and Services (IIWAS 2000), Yogyakarta, Indonesia, 2000

[Chauduri, 1995] Chaudhuri S., Krishnamurthy R., Potamianos S., Shim K. Optimizing queries with materialized views. Proc. of IEEE Conf. on Data Engineering, 1995.

[Chawathe, 1994] Chawathe S., Garcia-Molina H., Hammer J., Ireland K., Papakonstantinou Y., Ullman J., and Widom J. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In Proc. of IPSJ Conference, pp.7-18, 1994.

[Dang-Ngoc, 2003] Dang-Ngoc T.-T. and Gardarin G. Federating heterogeneous data sources with XML. In Proc. of IASTED IKS Conference, 2003..

[El-Sayed, 2002] El-Sayed M., Wang L., Ding L., Rundensteiner E.A. An algebraic approach for incremental maintenance of materialized XQuery views. WIDM 2002: 88-91.

[Garcia-Molina, 1997] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajamaran, Y. Sagiv, J. Ullman, V. Vassalos and J. Widom, The TSIMMIS Approach to Mediation: Data Models and Languages, Journal of Intelligent Information Systems, 8 (1997) 117-132.

[Gardarin, 2002] Gardarin G., Mensch A., Tomasic A.: An Introduction to the e-XML Data Integration Suite. EDBT 2002: 297-306

[Gardarin, 2003] Gardarin G, Kou H, Zeitouni K., Meng X. and Wang H., SEWISE: An Ontology-based Web Information Search Engine, 8th NLDB, Germany, June, 2003.

[Haas, 1999] Haas, L.M., R.J. Miller, B. Niswonger, M. Tork Roth, P.M. Schwarz, E.L. Wimmers: Transforming Heterogeneous Data with Database Middleware: Beyond Integration. IEEE Tech. Bull. Data Engineering 22 (1), 1999:31-36.

[Ives, 1999] Z. G. Ives, D. Florescu, M. Friedman, A. Levy, D. S. Weld. An Adaptive Query Execution System for Data Integration. ACM SIGMOD Conference on Management of Data, Philadelphia, PA, June 1-3, 1999

[Manolescu, 2002] Manolescu I., Florescu D., Kossmann D.: "Answering XML Queries over Heterogeneous Data Sources", Proceeding of 27th VLDB, pp. 241-250, Roma, Italy, Sept. 2001.

[OWL, 2003] http://www.w3.org/TR/owl-features/

[Papakonstantinou, 2003] Papakonstantinou Y., Borkar et. al., "Queries and algebra in the Enosys Integration Platform", Data & Knowledge Engineering, Volume 44 , Issue 3 (March 2003), Special issue: Data integration over the Web, pp. 299 - 322

[RDF, 2002] http://www.w3.org/RDF/

[Wache, 2001] Wache H., Vogele T., Visser U., Stuchenschmidt H., Schuster G., Neumann H. and Hubner S. Ontology Based Integration Approaches, IJCAI Workshop: Ontologies and Information Sharing, pp.108-11, 2001.

[Wiederhold, 1992] Wiederhold G., Mediators in the Architecture of Future Information Systems, IEEE Computer, vol.25,N.3,pp.38-49, 1992.

[XQuery, 2002] http://www.w3.org/TR/2002/WD-xquery-20021115/

Summary

This paper introduces an extension of a mediator-wrapper architecture to integrate the semantic Web. The extensions are implemented in XLive, an XML mediator developed at PRiSM in the context of the WebSI European project. XLive is decomposing global XQuery to local ones according to source capabilities. After a short description of XLive and some recalls on the Semantic Web, an architecture encompassing three levels of ontology-based schemas is proposed. It is a hybrid architecture integrating both the local as view (LAV) and global as view (GAC) approaches. It allows the development of semantic Web wrappers and the progressive integration of sources according to global domain ontology. Finally, an extended XQuery mediator with view enrichment, a mapping design tool, and Web service wrappers allows full support of the three layered architecture.