A Design and Implementation of XML-based Mediation Framework(XMF) for Integration of Internet Information Resources[*]

Kangchan Lee, Jaehong Min, Kishik Park
Protocol Engineering Center
ETRI, Taejon, 305-350, Korea
{chan,jhmin,kipark}@etri.re.kr

Kyuchul Lee
Department of Computer Engineering
Chungnam National University, Taejon, 305-764, Korea
kclee@ce.cnu.ac.kr

Abstract

As the proliferation of the Internet, especially World Wide Web, numerous information resources have been constructed. The characteristics of information resources on the Internet are that the information resources are distributed, autonomous, and heterogeneous. Moreover each information resource has its own query method, data representation, and schema structure. The integration of information resources is one of the most important research issues in the Internet data management. The task of information resources integration system is to answer queries that require extracting and combining data from multiple information sources. In this paper, we propose an XML-based Mediation Framework(XMF) for integrating the Internet information resources.

1. Introduction

As the huge amount of information is accumulated on the Internet, the Internet emerges as the largest database. However, the Internet information resources are various according to the type of information resources that they maintain and the interface that they provide. Moreover, the only way to integrate data from the multiple sites is to build specialized applications by hand because the information of each resource is usually described in HTML on the Internet environment.

The main problem in integrating Internet resources, that is, is heterogeneity. Each site has the different semantic, structure, schema, and data model. In the field of database, Multi-database System(MDS)[1] that enables the utilization of a complete functionality of database integration was used in early stage. But the drawback of the MDS is that the MDS provides an integration method that only integrates data in databases. However recent integration requirements on the Internet are extended to integrate not only database but also data produced by Internet applications. A true sense of information integration is a seamless integration of database and Internet application data at the same time.

In this paper, we propose a new approach for integrating Internet information resources named XMF(XML-based Mediation Framework), which adopts the mediator-wrapper architecture[2][3] to provide the end user with an integrated view of the underlying information sources. To resolve the heterogeneity problem, XMF uses the Internet open standards. First, XMF describes the information resources and mapping rules in XML[4]. XMF solves the problem of schematic conflict with XMF mapping rules. XMF supports the integration of various kinds of information resources and dynamic management of global schema information in XML. Second, XMF's wrappers support well-known protocols such as HTTP and JDBC. Third, XMF uses XPath[5] as a query language because integrated result of XMF is an XML document.

In the remaining part of this paper, we introduce the features and architecture of XMF in Section 2. The mediation rule language and the query language of XMF are proposed in Section 3. In Section 4, we identify the comparison aspects and make a comparison between

[*] This works are partially supported by project of development for Industry technology, which is funded by MIC(Ministry of Information and Communication) of Korea

XMF and other mediators. Finally, we present the conclusion and future works in Section 5.

2. XMF Architecture

Figure 1 illustrates the layered architecture of XMF. XMF is composed of three layers, i.e. Application Layer, Mediation Layer, and Resource Layer. Various Internet information resources take their position in the Resource Layer. DBMSs, Search Engines, file systems, and applications on WWW are the examples of the Internet information resources.

XMF Wrapper is a module for extracting data from each Internet information resource and converting into XML instances. Each wrapper has one-to-one mapping relationship with each information resource in the resource layer. The Wrapper does not need to run on the same platform with the mediator.

Mediator in mediation layer is a module for information resource integration. The mediator controls each wrapper. The mediator provides own program library, which support API of XMF. Programmers can make the client program using the XMF Library with Java Language.

The uppermost layer is the application layer in XMF architecture. Web browsers and any kinds of user interfaces of XMF can be built on the XMF Library in the application layer. And XMF mapper™ that performs integration rule generation between global schema and local schema is also at the application layer.

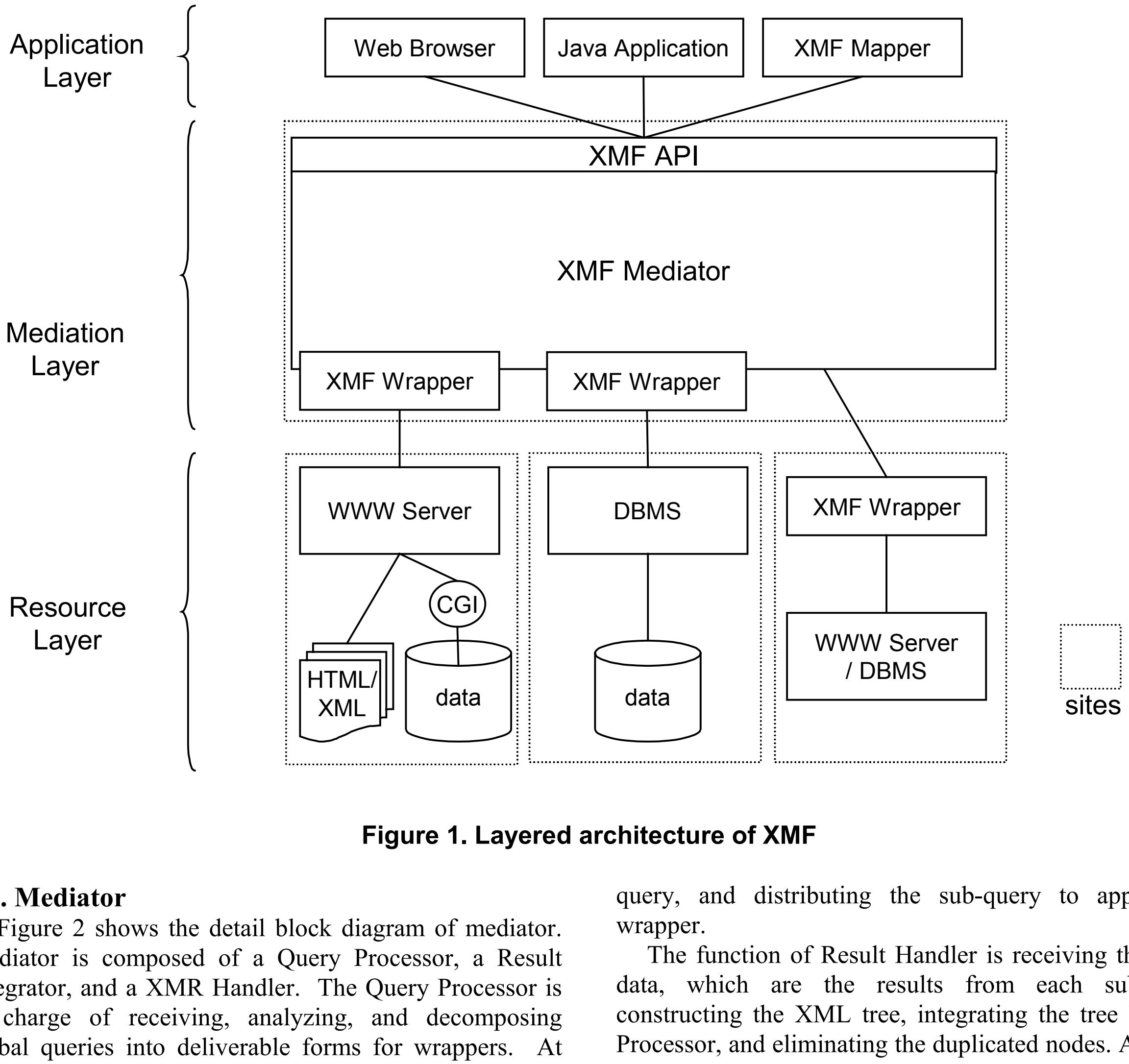

Figure 1. Layered architecture of XMF

2.1. Mediator

Figure 2 shows the detail block diagram of mediator. Mediator is composed of a Query Processor, a Result Integrator, and a XMR Handler. The Query Processor is in charge of receiving, analyzing, and decomposing global queries into deliverable forms for wrappers. At first, query processor tests the user query for query conformance test with stored global schema. After that, the main role of query processor is generating the sub-query, and distributing the sub-query to appropriate wrapper.

The function of Result Handler is receiving the XML data, which are the results from each sub-query, constructing the XML tree, integrating the tree by Rule Processor, and eliminating the duplicated nodes. And then, Result Handler checks the validity of the result with global schema, and returns the result to the application layer.

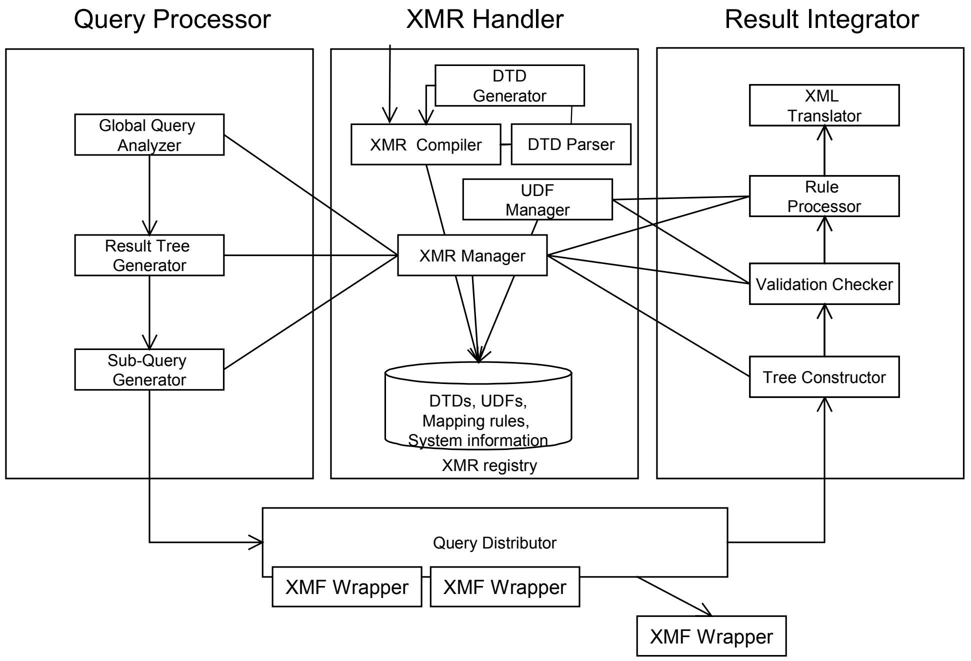

Figure 2. Mediator Architecture

The XMR Handler is the metadata management module. The XMR means XMF Mediation Rules include wrapper location information, user id, password, global and local schema, and the relationship between global schema and local schema. Resource Map Table, TypeID Table, Function Table is the main table for managing the metadata. If user wants to use XMF, user constructs the map information using XMF mapper, which is the visualization tool for generating the map information. After generating the mapping rule information, XMR handler manages the map information for query processor, result integrator, and user application.

2.2. Wrapper

The main function of Wrapper is to translate the sub-query into local query according to its information resource management system and to transform the local result into XML data format. The Wrapper is composed of Query Translator, Result Translator, and XMR Handler.

The role of Query Translator is receiving the query from the mediator, generating the local query for local information resource management system, and transmitting the query to local resource management systems. The local information resource management system returns the query result with its proprietary format. Result Translator in Wrapper harmonizes the result as XML format. After generating the result with XML format, wrapper checks results against the XML syntax.

XMF supports the five types of wrappers as shown in Table 1. Type 1, 2, 5 are wrappers for integrating of WWW information resources, and Type 3, 4 are wrappers for the integrating of DBMS information resources. The reason of why XMF supports various wrapper types is that there are many types of information systems on the Internet and each information system has the characteristics in querying usage, result format, and connection method.

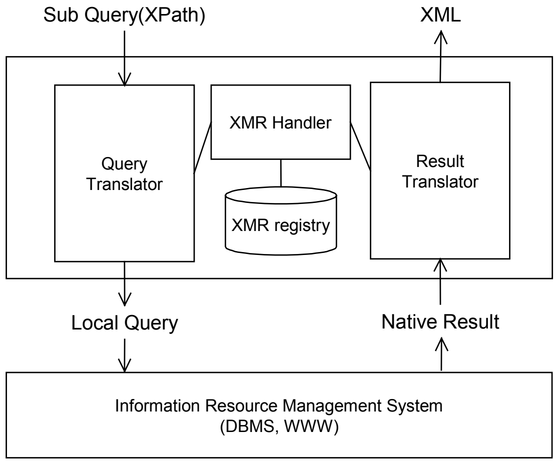

Figure 3. Wrapper Architecture

Table 1. Wrapper types of XMF

	Type 1	Type 2	Type 3	Type 4	Type 5
Information Resources	WWW	WWW	DBMS	DBMS	WWW
Protocol	HTTP	HTTP	JDBC	JDBC	HTTP
Query Type	Query String	Query String	SQL	SQL	Xpath
Result Format	HTML	XML	Database Result	XML	XML

3. XMF Mediation Rules

XMF overcomes heterogeneity of data models, DBMS types and platforms by using XML on the Internet. The reason why we use XML is that it provides the common data-modeling framework of heterogeneous information integration, and interoperability between applications. Furthermore, XML supports the separation of content, structure, and presentation. The mediator based on XML in XMF can easily integrate the data, and users of XMF can access information transparently without having to know what software and hardware platform are resided and without having to know where it is located.

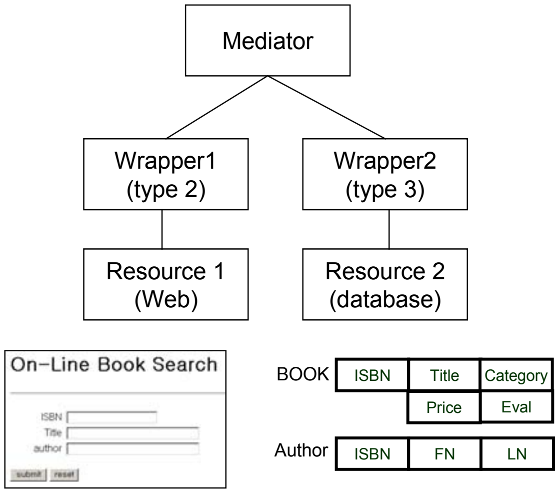

Figure 4. XMF Usage Environment

In this section, we show an example of integrating distributed, heterogeneous, and autonomous on-line bookstore on the Internet. Let suppose that there are two on-line bookstores. For convenience, we illustrate the XMF environment in Figure 4 and name each bookstore R1(Resource1) and R2(Resource2). Although R1 and R2 contain the book information, the data structures and units are different. Let's assume that the author information of R1 is represented by the book author's full name, and the author information of R2 is divided into first name and last name, and the price unit of R1 is Korean Won(₩) and R2 is Dollar($). And more, R2 has the information of category, but R1 has not. Because all of books information in R1 is related to computer and default category of R1 is the 'Computer'.

In XMF, wrapper makes it possible that each on-line bookstore can be considered as a XML repository. Each wrapper of bookstore understands the XMF queries and returns the results in XML format as shown in Figure 5, which shows the query results of each wrapper and the results are similar but the structures are different.

Wrapper1 (Resource1)

```
<Books>
<Book ISBN="1-800-32-31345">
  <title>Database</title>
  <Author>Kangchan Lee</Author>
  <Price unit="₩">15000</Price>
  <eval>2</eval>
  <Category>Computer</Category>

</Book>
...
</Books>
```

Wrapper2 (Resource2)

```
<products>
<Book title="XML Bible">
  <isbn>1-800-47-12121</isbn>
  <authors>
    <author><fn>Bob</fn>
            <ln>Roy</ln></author>
  </authors>
  <price>25</price>
  <evaluation>A</evaluation>
<Book>
...
</products>
```

Figure 5. The Result of Wrapper

For the integration of the result of the wrapper, XMF utilize XMR(XMF Mediation Rule), which defines mapping information between global schema and local schema. XMR consists of three consecutive blocks: (i) data registry block contains access method (resource name, location information, ID, password) description for local information system, user-defined function for resolving the semantic conflict, (ii) schema definition block has the information of global/local schema represented in DTD, XML schema, and (iii) relationship information between global schema and local schemas are in mapping rule block.

Figure 6 is an example of XMF mediation rule for integrating the Internet information resources in Figure 4. The XMR consists of a main element, xmr, and the subelements registry, schemas, and maps, etc. These subelements in turn contain other subelements such as integrate, source, dest, schema, and map. Integrating resources are defined using the integrate element and such definitions typically contain a set of integrating resources. Remarkable point of integrate element is that it has the operation attribute. According to the value of operation attribute, XMF mediator integrates the Internet information resources. Available operations are *merge*, *union*, and *join*. The name, connection, and type attribute in source element indicates the wrapper name, location, and type of integrating information resources, respectively, and UDF(user defined function) for resolving semantic conflict is registered in function element.

```
<?xml version="1.0"?>
<!DOCTYPE xmr SYSTEM "xmf.dtd">
<xmr>
<registry>
    <integrate operation="merge">
      <source name="ls1" method="GET"  type="2">
         <url>http://foo.com/search.asp?</url>
         <query name="ISBN" />
         <query name="Title" />
         <query name="Author" />
      </source>
      <source name="ls2" connection="LOCAL" type="3" />
    </integrate>
    <dest name="gs" />
    <function name="arithmetic"  sclass="division.class"     dclass="multiplication.class"
/>
    <function name="concatenate" sclass="SpilitString.class" dclass="MergeString.class" />
    <function name="convert"     sclass="ToNumber.class"     dclass="ToChar.class" />
</registry>
<maps>
    <map dest="gs:/catalog"/>
    <map dest="gs:/catalog/book"/>
    <map dest="gs:/catalog/book/@ISBN" source="ls1:/Books/Book/@ISBN"/>
    <map dest="gs:/catalog/book/@ISBN" source="ls2:/products/Book/isbn"/>
    <map dest="gs:/catalog/book/title" source="ls1:/Books/book/title"/>
    <map dest="gs:/catalog/book/title" source="ls2:/products/Book/@title"/>
    <map dest="gs:/catalog/book/authors"/>
    <map dest="gs:/catalog/book/authors/author" source="ls1:/Books/book/Author"
         capability="selection"/>
    <map dest="gs:/catalog/book/authors/author"
         source="concatenate(ls2:/products/Book/authors/author/fn,
                             ls2:/products/Book/authors/author/ln)" />
    <map dest="gs:/catalog/book/price" source="ls1:/Books/book/Price"
         capability="selection"/>
    <map dest="gs:/catalog/book/price"
         source="arithmetic(ls2:/products/Book/price, 1200)" />
    <map dest="gs:/catalog/book/eval" source="convert(ls1:/Books/Book/eval)" />
    <map dest="gs:/catalog/book/eval" source="ls2:/products/Book/evaluation" />
    <map dest="gs:/catalog/book/category" source="ls1:/Books/Book/Category"/
         capability="selection"/>
    <map dest="gs:/catalog/book/category" source="Computer"/>
</maps>
</xmr>
```

Figure 6. The XMR Example

The global and local schema is defined in schema element in schemas element. The name attribute is correspondence with name attribute in source and dest element, and the value of ref attribute is the global and local schema filename.

XMF can analyze both DTD and XML Schema[6] for schema description of each information resources. And map elements in mapping rule division are the relationship between local schema and global schema. The map element has dest and source attributes. Each attribute has the value, which consists of global/local schema name and XPath like "gs:/catalog/book/@ISBN". If the value of source attribute has the form of function call, it means that the value of source attribute is the result of UDF, which is defined in function element. And more, setting the default value is available like "Computer".

XMF administrator should understand the whole of information resources and write out the XMR. Because XMR is the instance of XML, any one who knows the syntax of XML can writes the XMR files easily with a text editor. In addition to, we support XMF mapper, which is the XMR generation tool written in Java.

After the construction of XMR, user can use the XMF using XMF query. XMF support the query language, which is the subset of XPath. The noble aspect of XMF query is that mediator and wrapper has the same form of query language, execution model, and the result format. So, if there are several mediator and wrappers, user executes the query using hierarchical XML architecture.

```
<catalog>
   <book ISBN="1-800-32-31345">
     <title>Database</title>
     <authors>
       <author>Kangchan Lee</author>
     </authors>
     <price>15000</price>
     <eval>B</eval>
     <Category>Computer</Category>
   </book>
   <book ISBN="1-800-47-12121">
     <title>XML Bible</title>
     <authors>
       <author>Bob Roy</author>
     </authors>
     <price>30000</price>
     <eval>A</eval>
     <Category>Computer</Category>
  </book>
  ...
</catalog>
```

Figure 7. The Final Result of Mediator

4. Related Works

Various mediator systems, such as TSIMMIS, IM, YAT, IDIMS, HERMES, have been implemented for the integration of heterogeneous sources. The key aspects, which distinguish XMF from other systems, are that XMF is easy to implement and that users can have single view because XMF is only based on XML environment.

TSIMMIS[7] follows the mediator architecture, allowing us to create a hierarchy of wrappers and mediators that talk to one another. TSIMMIS components communicate among themselves using a data model named OEM(Object Exchange Model), which is an object-oriented model that uses object labels to represent both class information and attributes(instance variables) of object. The query language LOREL for OEM objects provides partial-match semantics matches the flexible structure of OEM objects. And MSL(Mediator Specification Language) is used to describe mediators and wrappers at a high level, and these components can be generated automatically from the MSL specification.

Information Manifold(IM)[8] is a system for browsing and querying of multiple networked information sources. IM's representation language enables describing the semantic content of structured sources in a way that can be used to answer queries that may involve accessing data in multiple sources. The architecture of IM is based on a knowledge base containing a rich domain model that enables describing properties of the information sources. In particular, IM's domain model includes the representation of topics of information sources, as well as properties having to do with the physical characteristics of the sources.

IDIMS(INEEL Data Integration Mediation System)[9] is a mediator system for database integration aimed for data integration from multiple heterogeneous sources. ODL of ODMS was extended to provide a highly dynamic method for domain definitions to IDIMS. IDIMS supports system scalability using common service interface, which is shared by both wrapper and mediator. Also, a newly query representation, QEM (Query Exchange Model) is suggested in IDIMS for common query represent method. QEM can present the structured query and support structural information of data. In order for mediators and wrappers to exchange data, IDIMS use OEM (Object Exchange Model) as a common data representation.

YAT[10] is a mediator system for the specification and the implementation of data conversions among heterogeneous data sources. The model of YAT is based on named trees with ordered and labeled modes. Like semi-structured data models, it is simple enough to facilitate the representation of any data and it can easily map anything into a tree/graph. The YAT conversion language is declarative, rule-based and features enhanced pattern matching facilities and powerful restructuring primitives.

Another system related to ours is MIX[11], which focus on wrapper-mediator systems which employ XML as a means for information modeling, as well as interchange, across heterogeneous information sources. The wrapper associated with each source exports an XML view of the information at that source. The mediator is responsible for selecting, restructuring, and merging information from autonomous sources and for providing an integrated XML view of the information

Table 2 summarizes features the mediator systems mentioned above. The comparison aspects are data model, query language, the form of result, global schema definition(source representation), target resources, and communication protocol among the mediator component.

Table 2. Comparison of mediator systems

	TSIMMIS	IM	YAT	IDIMS	MIX	XMF
Common Data Model	OEM	Logic	Tree	ODL	XML	XML
Query Language	Lorel	Logic	YATL (Rule based)	QEM	XMAS	Xpath
Result Format	OEM	Logic Result	Rule Result	OEM	XML	XML
Global schema definition	MSL	Predicate	Rule	MODL (Mediator ODL)	XMAS	XMR (XML)
Resource	Sybase, file, WWW	WWW, other information sources	Internet resources	Oracle, Foxpro	WWW	Internet Information Resources
Protocol	CORBA + Internet	WWW	WWW	Proprietary	HTTP(GET)	HTTP,JDBC

5. Conclusion

With the recent advances in information technology such as digital libraries, WWW, data warehouse, and CALS, structured and unstructured data have been widely recognized as important information resources. Moreover, information resources on the Internet are often maintained in heterogeneous, distributed, and autonomous information repositories. Thus, the integration of Internet information resources is one of the significant issues.

In this paper, we propose a new integration framework, XMF, which provides uniform views over large number of Internet information resources by using only XML and Internet. XML provides self-describing modeling method for capturing semantic of heterogeneous information resources, and the Internet protocol supports the common data communication mechanism. The features of XMF are integrating various kinds of information sources and its application on the Internet, supporting common data model and run-time integration of information resources by using its mediation mechanism and query language. In consequence, XMF supports common architecture and query language for integrating the Internet information resources and user can easily access XMF with uniform method. Furthermore, XMF can be easily implemented with current Internet technology and XML-related software.

We anticipate that flexible, efficient, and general-purpose heterogeneous and distributed information resource integration methodology is needed as huge amount of information is accumulated on the Internet. XMF is the one of the solutions of seamless integration of Internet information resources.

The future works include the following research items:

- Expending the query processor in order to support more complex and more colorful user queries
- Refining and verifying the DTD of mediation language for various application, and improving the Rule Processor
- Devising automatic generation mechanism of wrappers from rule descriptions
- Supporting the full-grown XML query language, which is being developed by W3C
- Extending the various kinds of XMF wrapper, refining the query processing and result integrating mechanism.

References

[1] Won Kim, Injun Choi, Sunit Gala, and Mark Scheevel, “On Resolving Schematic Heterogeneity in Multidatabase Systems,” *Modern Database Systems; The Object Model, Interoperability, and Beyond*, Addison-Wesley Publishing Company, pp. 521-550, 1995.

[2] Gio Wiederhold, “ Mediators in the Architecture of Future Information Systems,” *The IEEE Computer Magazine*, 25(3):38-49, March 1992.

[3] Paul Benjamin Lowry, “XML Data Mediation and Collaboration: A Proposed Comprehensive Architecture and Query Requirements for using to Mediate Heterogeneous Data Sources and Targets,” *Proceedings of Hawaii International Conference on System Sciences*, 2001.

[4] Tim Bray and C.M. Sperberg-McQueen, "Extensible Markup Language (XML): Part I. Syntax", World Wide Web Consortium Recommendations, February 1998, Available at http://www.w3.org/TR/REC-xml.

[5] James Clark, Steve DeRose, "XML Path Language (XPath) Version 1.0", World Wide Web Consortium Recommendations, Nov. 1999, Available at http://www.w3.org/TR/xpath.

[6] David C. Fallside, "XML Schema Part 0: Primer", World Wide Web Consortium Candidate Recommendation, October 2000, Available at http://www .w3.org/TR/xmlschema-0/.

[7] Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey D. Ullman, Vasilis Vassalos, Jennifer Widom, “The TSIMMIS Approach to Mediation: Data Models and Languages,” *Journal of Intelligent Information Systems*, Vol. 8, No. 2, pp. 117-132, 1997.

[8] Alon Y. Levy, Anand Rajaraman, Joann J. Ordille, “The World Wide Web as a Collection of Views: Query Processing in the Information Manifold”, *Proceedings of Workshop on Materialized Views: Techniques and Applications*, pp. 43-55, 1996.

[9] B. Panchapagesan, J. Hui, G. Wiederhold, S. Erickson, L. Dean, “The INEEL Data Integration Mediation System White Paper”, Available at http://id.inel.gov/idim/paper.html.

[10] Sophie Cluet, Claude Delobel, Jérôme Siméon, Katarzyna Smaga, “Your Mediators Need Data Conversion!,” *Proceeding of ACM SIGMOD Conference,* pp. 177-188, 1998.

[11] Chaitanya K. Baru, Amarnath Gupta, Bertram Ludäscher, Richard Marciano, Yannis Papakonstantinou, Pavel Velikhov, Vincent Chu, “XML-Based Information Mediation with MIX,” *Proceeding of International Conference on ACM SIGMOD*, pp. 597-599, 1999.