行政院國家科學委員會專題研究計畫　期中進度報告

本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(1/2)

計畫類別：個別型計畫
計畫編號：NSC94-2416-H-004-018-
執行期間：94 年 08 月 01 日至 95 年 07 月 31 日
執行單位：國立政治大學資訊管理研究所

計畫主持人：諶家蘭

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 95 年 5 月 24 日

行政院國家科學委員會補助專題研究計畫 □ 成 果 報 告
■期中進度報告

本體論和資料模式輔助之資訊整合與績效評估工作量模型研究(1/2)

計 畫 類 別 ： ■ 個 別 型 計 畫 □ 整 合 型 計 畫
計 畫 編 號 ： NSC 94－ 2416－ H－ 004－ 018－
執 行 期 間 ： 2005 年 8 月 1 日至 2006 年 7 月 31 日

計畫主持人：謝家蘭
共同主持人：
計畫參與人員：

成果報告類型(依經費核定清單規定繳交)：□精簡報告 ■完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立政治大學會計學系所

中 華 民 國 2006 年 05 月 22 日

Abstract

The research issues of heterogeneous information integration have become ubiquitous and critically important in e-business (EB) with the increasing dependence on Internet/Intranet and information technology (IT). Accessing the heterogeneous information sources separately without integration may lead to the chaos of information requested. It is also not cost-effective in EB settings. A common general way to deal with heterogeneity problems in traditional heterogeneous information integration (HII) is to create a common data model. The eXtensible Markup Language (XML) has been the standard data document format for exchanging information on the Web. XML only deals with the structural heterogeneity; it can barely handle the semantic heterogeneity. Ontologies are regarded as an important and natural means to represent the implicit semantics and relationships in the real world. And they are used to assist to reach semantic interoperability in HII in this research.

In this research, we provide a generic construct orientation no ad hoc method to generate the global schema to enable the web-based alternative to traditional HII. We provide a wiser query method over multiple heterogeneous information sources by applying global-as-view (GAV) approach with the use of ontology to enhance both structural and semantic interoperability of the underlying heterogeneous information sources. We construct a prototype implementing the method to provide a proof on the validity and feasibility.

Keywords: Heterogeneous Information Integration, XML, Ontology, Syntactic and Semantic Interoperability

摘要

由於對資訊科技以及網際網路/和企業內網路的依賴持續加深，異質資訊整合在電子化企業中已經成為一個普遍存在且相當重要的議題。因為在缺乏整合的情形下，個別地存取異質資訊來源可能會造成資訊的混亂，而且在電子化企業的環境中，這麼做也不符合成本效益。在傳統異質資訊整合的研究中，通常會創造一個共同資料模式來處理異質性的問題，而可延伸性標記語言(XML)已經成為網路上交換資訊時的標準文件格式，使得 XML 成為整合工作中共同資料模式標準的一個很好的選擇；然而，XML 僅能夠處理結構異質性，無法處理語意異質性，而本體論被視為是一個重要而且自然的工具，用來表現真實世界中模糊不清的語意和關係，因此，在本研究中也加入了本體論以期達到異質資訊整合中的語意互動性。

在本研究中，我們提出一個非特殊隨機式對應方法的一般性概念導向來產生全區域綱要方法（Global Schema），以達成相對於傳統，以網路為基礎的異質資訊整合。我們也提出一個較具智慧性的多異質資訊來源的查詢方法，該查詢方法應用了 global-as-view （GAV）全區域景觀導向方法加上本體論觀念運用，可以同時提高對底層異質資訊來源的結構互動性和語意互動性。我們透過雛型系統的實作來驗證本研究所提供的異質資訊整合方法的可行性。

關鍵字：異質資訊整合、延伸性標記語言、本體論、結構互動性和語意互動性

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Research Motivation

The research issues for heterogeneous information integration (HII) have become ubiquitous and critically important with the increasing dependence on Internet/Intranet and information technology (IT). In a contemporary firm, information is distributed company-wide due to competition, evolving technology, mergers, acquisitions, and geographic distribution. The popularity and dynamics of the Internet/Intranet is another source of this widespread distribution, with information represented and stored in different forms including structured data, semi-structured data, and unstructured data. Heterogeneity is here to stay for these very reasons and it is up to users and managers in terms of how and why to tackle the integration and distribution research issues.

Although Information technology marks a new era in business management and enables any firm to achieve complicated e-business, it may face difficulty in dealing with the distributed and heterogeneous information sources. One of the main concerns is that the information obtained may be inconsistent and contradictory. The other is how to access the different information sources effectively and efficiently. Therefore, heterogeneous information integration has been at the top of list for IT investment and strategy (Jhingran, Mattos, & Pirahesh, 2002). Companies must make a concerted effort to tackle information integration. In light of this, the objective of this research is to address the research problems of interoperability, scalability and portability between multiple heterogeneous information sources. However, so far the results are neither adequate nor complete.

1.2. Research Issue

There are some roadblocks in the way of achieving effective information integration. For example, information sources are distributed and the amounts of data are large. So the first target is to efficiently access multiple information sources and to decrease large amount of information transformation. Moreover, information sources are heterogeneous in system, syntax, structure and semantics (Sheth, 1998). A general way to deal with heterogeneity problems is to create a common data model. It plays the role of giving a common representation for the different information sources handled by it, and it offers users a global view of the information sources that can be accessed.

In the previous works, they adopted expert-dependent method to create the common data model for the interoperability between the underlying heterogeneous information sources. It does not seem to fit in with syntactic as well as semantic web-based heterogeneous information integration in e-business (EB). So in this research, we try to find out a solution to solve this shortcoming in the previous works.

The eXtensible Markup Language (XML) has been the W3C standard document format for exchanging information on the Web. It is a good candidate to be the lowest common denominator for integration tasking. Some of its numerous advantages are that it is simple and self-describing.

Furthermore, some related technologies such as the query language utilized by it (for example, XQuery) have also been standardized recently. Due to the above advantages of XML, there are more and more research adopted XML as the common data model. However, while XML can indeed establish interoperability between different information sources on the Web, its main limitation is that it copes only with structural heterogeneity; it can barely handle semantic heterogeneity. Hence, we deem that it can hardly reach the better interoperability of structure and semantics between heterogeneous information sources over EB settings.

The clarification of implicit and hidden knowledge depends on ontologies, which can be regarded as an important and natural means to represent real world knowledge XML can hardly catch. For example, XML cannot catch the relationships like intersection, union, complement and so on. In contrast, ontologies are fit to represent such relationships. Hence, we try to combine this two different data models in order to reach better interoperability not only about structures but also semantics of the heterogeneous information sources. As such, ontology is added in this research to assist to reach not only structure but also semantic interoperability in HII.

1.3. Research Objective

This research dealing with HII alternative relies on XML with ontology assisted. The main objectives of this research are:

A. To enable the web-based alternative to traditional HII in EB settings.
B. To enhance both structural and semantic interoperability of the underlying heterogeneous information sources by extending global schema with ontology.
C. To enhance the query processing capability by taking the advantage of global schema's easy and efficient query reformulation characteristics and working with ontology.

2. Research Methodology

In this section we further describe the integration problems and present our research method and research structure. We propose the approach to tackle the research problems addressed in prior section. We focus on the resolution of the heterogeneity problems among information integration over the Internet. This research approach hopes to provide systematic and methodological information integration.

2.1. Research Method

Liang (1997) summarized the MIS research methods. He stated that MIS scholars held a series of conferences on research methods in 1989, and identified the five primary research methods including (1) case study, (2) survey, (3) experiment, (4) model driven, and (5) prototyping. Taking the five methods into consideration, prototyping is suitable and fit to be applied in this research.

2.2. Research Structure

There have been many works focusing on heterogeneous information integration. Typical

information integration systems have adopted mediator/wrapper architecture (Wiederhold, 1993). Under such architecture, the mediator provides an integrated and global view of different heterogeneous information sources. With this view, queries can be formulated by the clients. Besides, wrappers provide local views of information sources in a uniform data model. The local views can be queried in a limited way according to wrapper capabilities.

TSIMMIS, DISCO, Garlic, Information Manifold and so on were the methods which have adopted mediator/wrapper architecture. They focused on providing an integrated data model that is an object model. However, beginning in the 21st century, XML has taken the place of object model as the pivot model. XML has become an emerging standard of data exchange and has many advantages to become the best candidate to be the common data model when performing heterogeneous information integration.

However, the information integration studies which adopt mediator/wrapper architecture and use XML as the common data model to capture heterogeneous sources have met with semantic problems, but only syntactical and structure ones. Ontology from the field of artificial intelligence describes the knowledge representation that provides definitions of vocabulary in certain domain. The use of ontology to explicate and explore the implicit and hidden knowledge seems to be a promising approach to tackle the problems of semantic heterogeneity. Therefore, we add ontology and develop an information integration model and method that is based on mediator/wrapper architecture to solve the heterogeneity problems over the heterogeneous information sources. Users can thus access heterogeneous information sources via one uniform and seamless platform.

The research structure is illustrated in Figure 2-1.

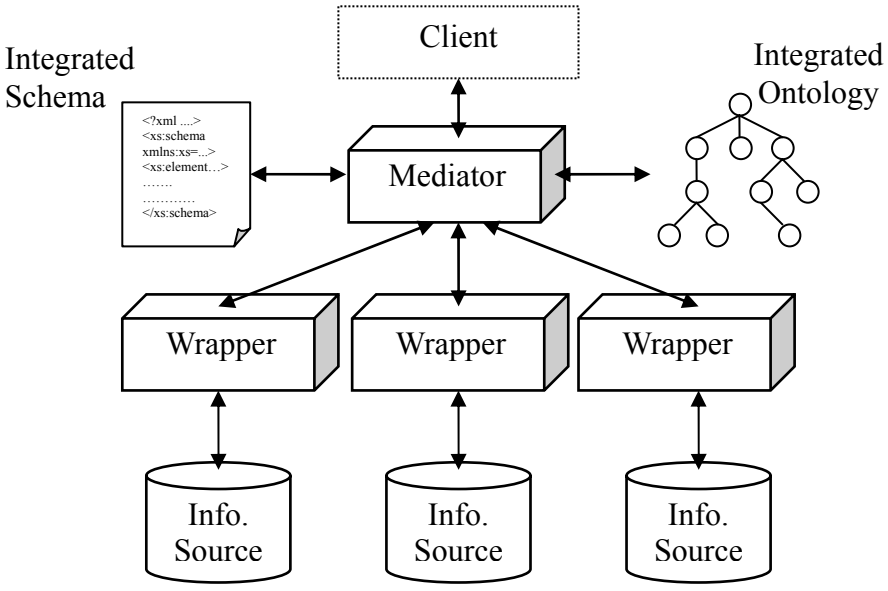


Figure 2-1: Research Structure

At the bottom of Figure 2-1 there are a number of information sources which contain diverse information that needs integration. Different information sources present their own data in a different data model so the client has to use different access interfaces to get the data, and at the same time, take the following details into consideration, such as the location of data,

effectiveness and efficiency of accessing different information sources, data quality, and consistency if an update is performed. To overcome the above difficulties, we construct corresponding wrappers for different types of information sources. The wrapper is used to translate data access and manipulation requests between mediator and information sources. Above each wrapper in the figure is a mediator, in charge of query processing in the research structure. In addition, the mediator provides the client with the integrated view of the underlying heterogeneous information sources and processes clients' queries against the information sources.

In the following, we describe the research structure in detail. The components in the research structure are depicted in Figure 2-2.

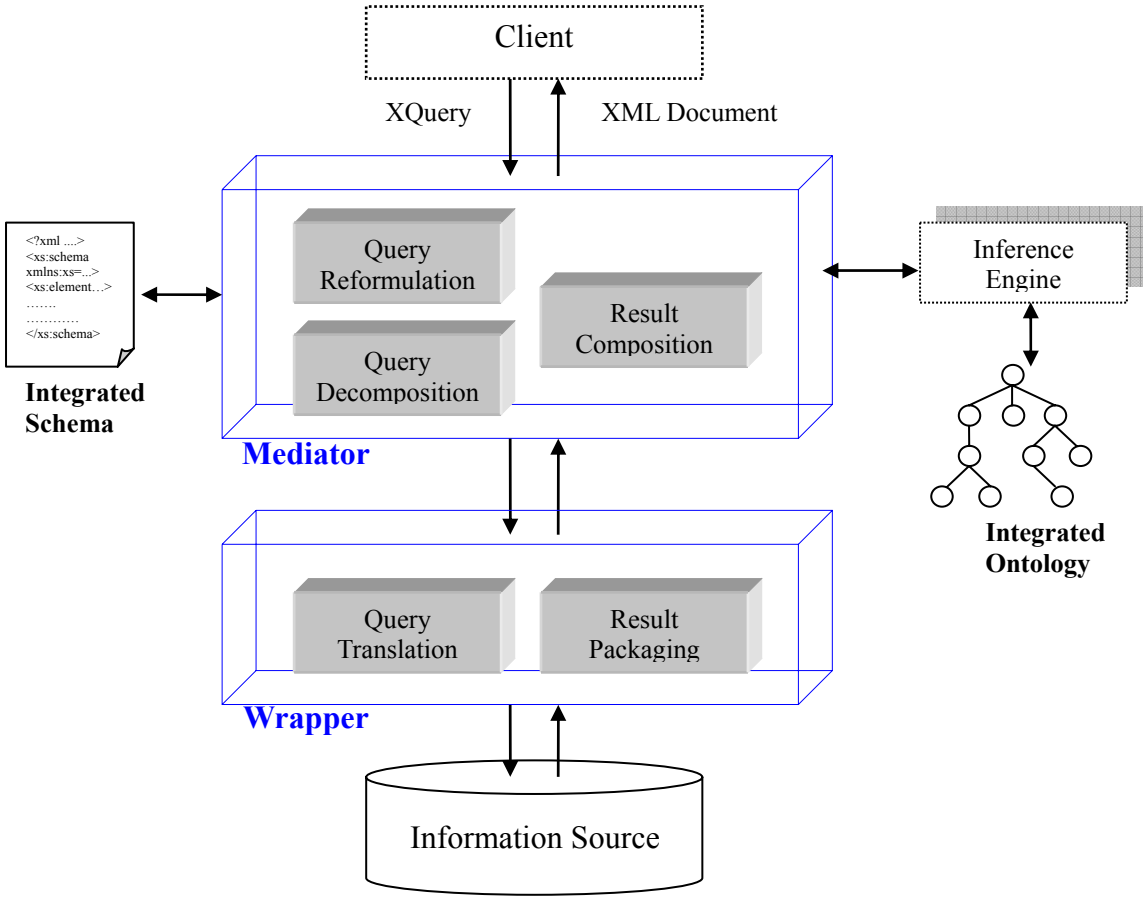


Figure 2-2: Components in Research Structure

As the Figure 2-2 shows, there are two major parts in our research structure: (1) Mediator (2) Wrapper. We describe the functions of the individual component of each part as follows.

First, the components in the mediator:

A. Query reformulation is used to receive the query from system query interface and then send out a reasoning request according to the query from the interface to the inference engine in order to find out the implicit knowledge and relationships in that query. The inference engine then gets the reasoning result and passes it back to the mediator, in which query reformulation component receives this result. The component, according to the result, reformulates the query to represent the facts in an explicit form. Afterwards, it passes the reformulated queries to the query decomposition component for further process.

B. Query decomposition receives the reformulated query from query reformulation component.

After receiving the query, it decomposes the query into several sub-queries according to the integrated schema and the specified mapping between global schema and local schemas. Then it passes those sub-queries to the corresponding wrappers.

C. Result composition receives the packaged results from wrappers and recombines the results into an XML document according to the user request. It may also require the assistance of the integrated schema while composing the results.

Second, the components in the wrapper:

A. Query translation is used to receive the sub-query of the target source and then translate it into native query of that information source. After that, it sends the native query into the underlying information source for finding out the data demanded.

B. Result packaging gathers the native results and packages them in a form that is known by the mediator. Then, it sends the packaged result to the mediator for further process.

2.3. Information Integration Method in Research Structure

In this section, we detail our methods of information integration in our research structure. Our goal is to provide a convenient and effective way for users to access a number of heterogeneous information sources simultaneously and get an integrated result just like accessing only one information source. Users who interact with the information integration structure do not have to consider the details of the information sources they face. To achieve this goal, we must integrate the underlying sources and provide users with a unified view of the structure and content of these sources. Providing the unified view depends on the integration of different data models of the underlying information sources. Hence, integrating different data models of the underlying sources is significant and helpful.

But before performing the data model integration, we must identify problems that we will meet in the information integration. Problems coming from heterogeneity of the data are already well known within the distributed database systems community: (Cui, Jones, & O'Brien, 2001; Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann, & Hübner, 2001).

A. The system level of heterogeneity includes incompatible hardware and software systems, which results in a variety of different access mechanisms and protocols.

B. The syntactic level of heterogeneity refers to different languages and data representations;

C. The structural level includes different data models;

D. The semantic level considers the contents of an information item and its intended meaning.

XML is widely predicted to improve the degree of interoperation on the Internet. Yet XML does not address ontology and provides only a syntactic and structure representation of knowledge. For this reason, we use XML as the uniform data model for performing HII with ontology assisted for the dimension of the semantics. We would like to present the details of our methods of HII as follows. And we use an example which is about the domain of university to explain our method of information integration.

2.4. The Creation of Global Schema

When performing heterogeneous information integration, we first encounter the representation problem for the structure of different data models. Parent et al. 1998 formalized the database integration process in order to develop an integrated schema (see Figure 2-3). To establish the integrated schema as a unified view of existing information sources, the heterogeneous schemas of the corresponding underlying information sources are usually transformed to make them as homogeneous as possible. Researchers in database integration generally assume that input schemas are all expressed using the same data model, the so-called "common" data model.

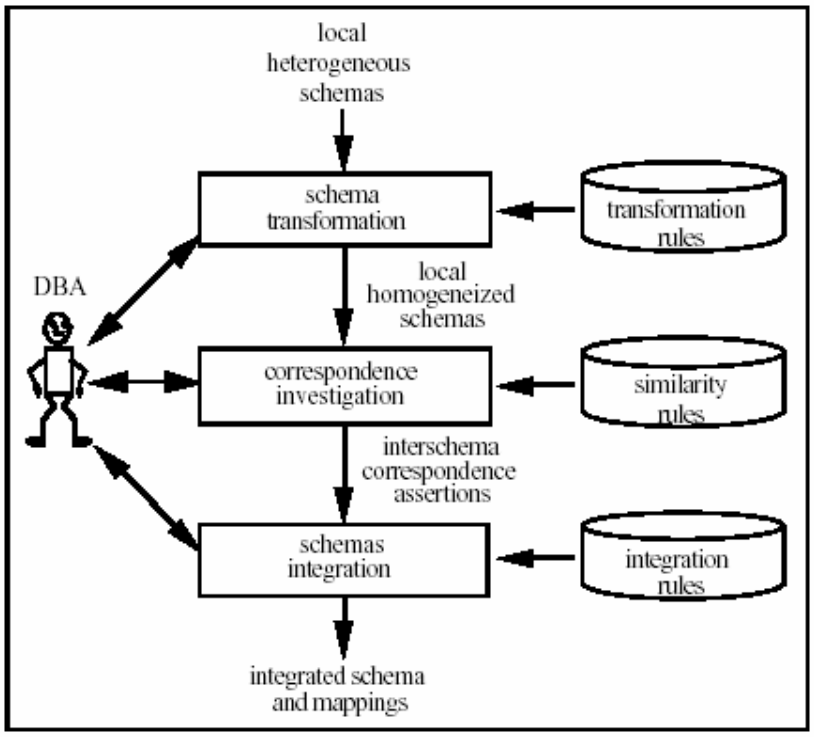


Figure 2-3: The Global Integration Process
(Data Source: Parent, & Spaccapietra, 1998)

In this subsection, we extend the published database integration process to our information integration method to create the global view of the underlying information sources. In contrast to the traditional HII, they use an expert-dependent method to create their global schema. However, in this research, we try to provide a more general method to handle this issue. We use XML as the common data model to enable HII and propose two steps for the creation of global schema in our method, which are: (1) Generic Construct Oriented Schema Rewriting, and (2) Schema Integration. Performing the schema transformation by using the method of generic construct oriented schema rewriting is a more general and convenient way to apply to most kinds of information sources in contrast to the traditional method. That is our emphatic point.

2.5. Generic Construct Oriented Schema Rewriting

In order to homogenize the representations of the data models using in heterogeneous information sources, we have to create rules for rewriting between XML and the native data

models. Since our information integration model is regarded as a generic model, it is expected to tackle any kinds of information sources. The heterogeneous information sources that we most often encounter can be roughly classified into three categories, which are: structured information sources, semi-structured information sources, and unstructured information sources. Structured information sources include Relational Database Management System (RDBMS) and Enterprise Information System (EIS, such as ERP, SCM, and CRM) files, among others. One example of semi-structured information sources may be Object Database Management System (ODBMS) or XML data files. Unstructured information sources may include HTML pages, multimedia files, office flies, and legacy files, and so on.

We deign to apply the generic construct oriented schema rewriting process to the structured and semi-structured information sources. The unstructured information source here is hard pressed to receive this type of HII pre-processes because it is lack of the structure definition, schema. As such, in our research structure we treat the unstructured information sources as special cases and they need an additional process described in the later sections.

To transform the data models of the structured and semi-structured information sources into XML, we have to specify one-to-one rewriting rules for every native data model. Before specifying the rewriting rules, we have to identify the correspondences between the constructs of XML and other native data models. Here we provide the correspondences between XML and two representative data models of structured and semi-structured information sources, which are a relational model and an object model as explained. Table 2-1 shows the correspondences of relational schema constructs and XML Schema constructs. According to the specified correspondences, the relational schema can be rewritten into a W3C XML Schema just as the example shown in Figure 2-4 describes.

Table 2-1: Correspondences between Relational Schema Constructs and W3C XML Schema Constructs

Relational Schema Constructs	W3C XML Schema Constructs
Relation	element (with xs:complexType)
Attribute	element
Date type	date type (primitive type / xs:simpleType)
Cardinality	multiplicity (minOccurs / maxOccurs)
primary key (PK)	key (xs:key)
foreign key (FK)	keyref (xs:keyref)

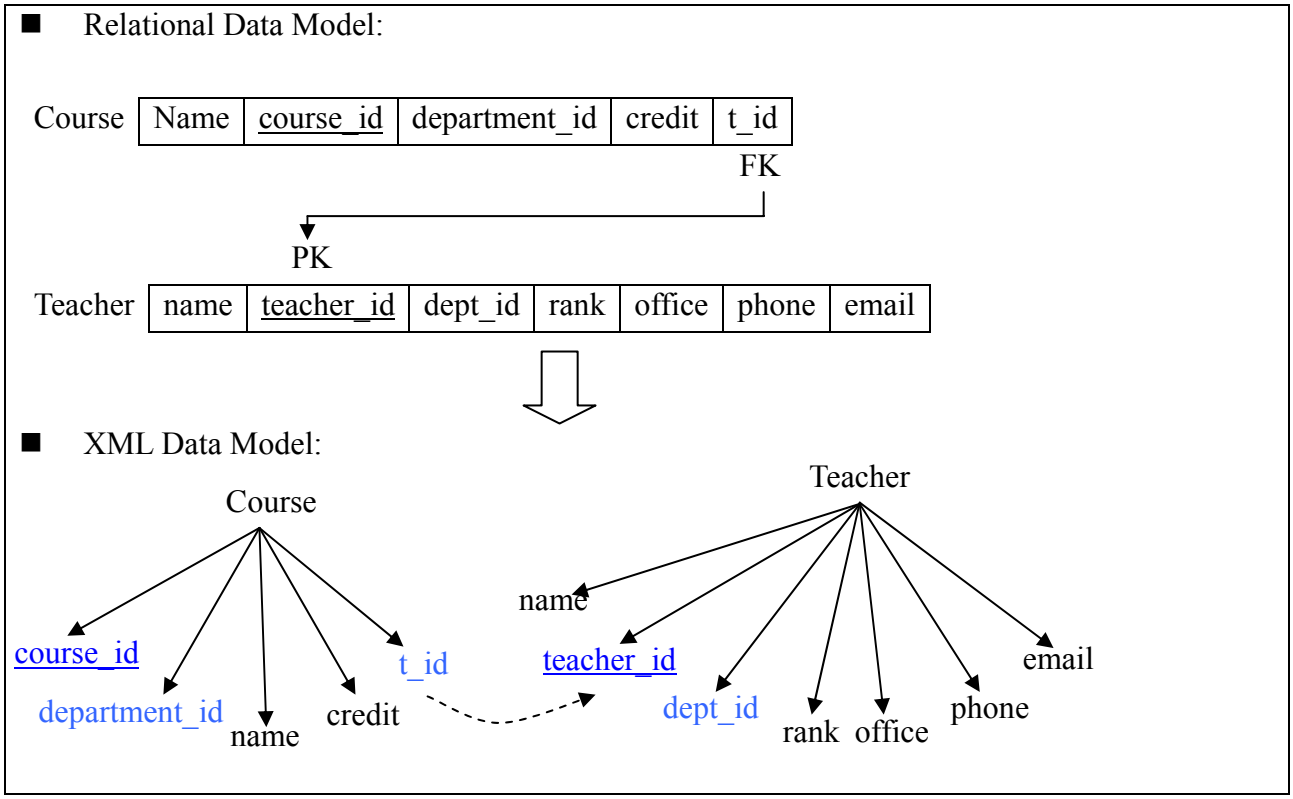


Figure 2-4: Transform Relational Data Model into XML Data Model

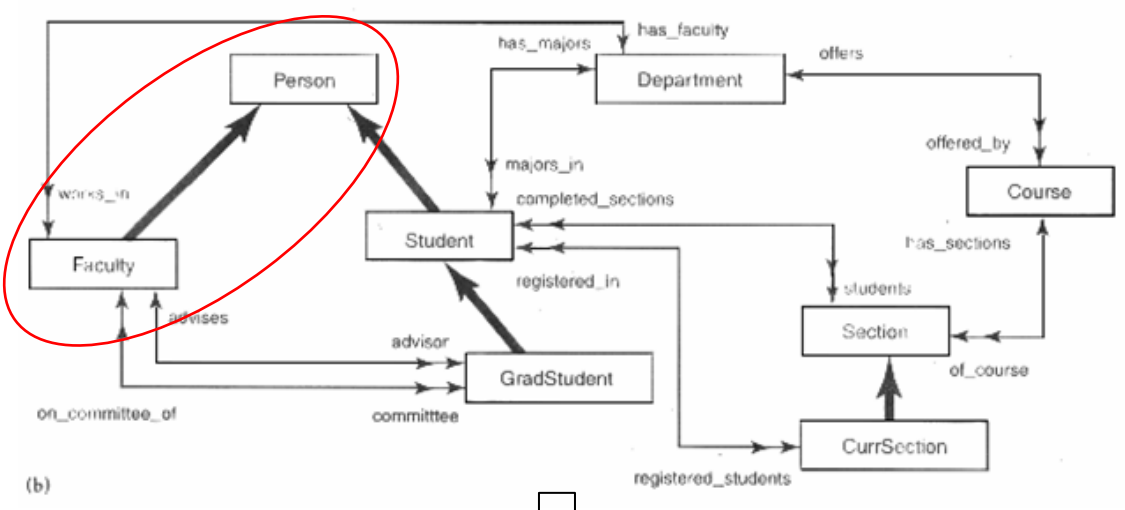
Table 2-2 shows the correspondences between object database schema constructs and XML Schema constructs. Similar to rewriting relational schema into W3C XML Schema, we provide a simple example in Figure 2-5 to illustrate the transformation between object database schema and XML Schema according to the specified correspondences in Table 2-2.

Table 2-2: Correspondences between Object Database Schema Constructs and W3C XML Schema Constructs

Object Database Schema Constructs	W3C XML Schema Constructs
Class	element (with xs:complexType)
attribute (simple)	element
primitive type	data type (primitive type)
struct (user-defined type)	data type (xs:simpleType / xs:complexType)
Key	key (xs:key)
extend (inheritance)	only single inheritance supported (xs:extension / xs:restriction)
relationship/inverse	Not Supported
Extent	
Method	

We use an object data model that is illustrated in (Elmasri, & Navathe, 2000) to be an example. We use just two classes, “Person” and “Faculty”, of the entire data model in order to show how to rewrite an ODL schema for object database into W3C XML Schema.

■ Object Data Model: (Data Source: Elmasri, & Navathe, 2004)



■ XML Data Model:

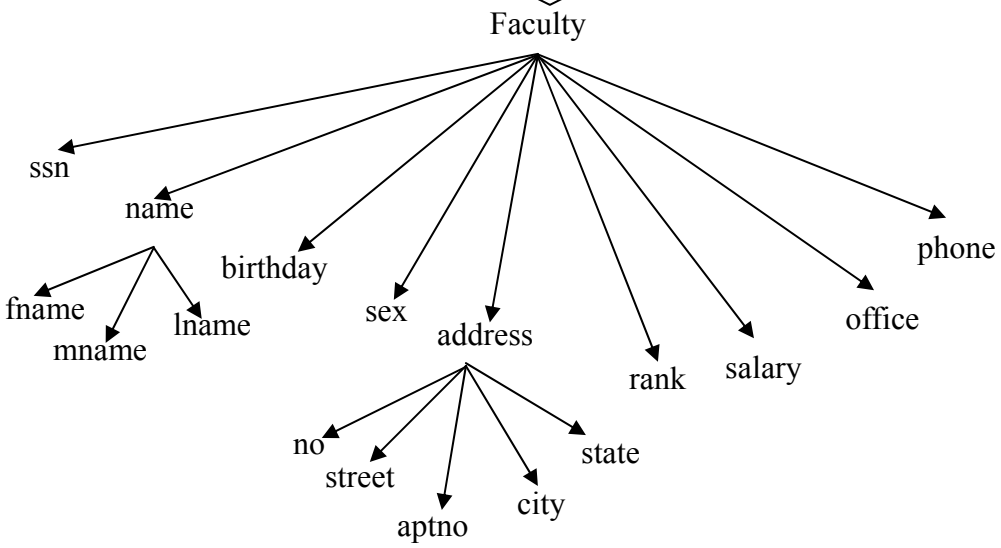


Figure 2-5: An Example of Transforming Object Data Model to XML Data Model

We just show the generic construct correspondences between XML and two structured information sources, RDBMS and ODBMS, as explained. According to the correspondences, we can rewrite the native data model using in local information source into XML. Different information sources use different data model to describe their own data. To enable heterogeneous information integration, the one-to-one generic construct correspondences to rewrite the local data model into the common data model, XML, by using our integration structure is necessary.

2.6. Schema Integration

Before we integrate the schemas, we have to identify the commonalities between different schemas and characterize the inter-schema relationships. Schema integration uses the correspondences to find similar structures in heterogeneous schemas, which are then used as integration points.

However, in order to find out the correspondences between a set of independently developed schemas, we must recognize the causes for the structural heterogeneity between them in advance.

We must gain the interoperability among the underlying sources by solving the heterogeneity problems between them so that we will achieve information integration. But the causes of the heterogeneity must be clarified first, and then we can deal with the heterogeneity problems by pointing out the correspondences between different schemas. Kashyap et al. (1996) and Visser et al. (2003) have categorized the causes for structural heterogeneity. We summarize the reasons for them in Table 2-3.

Table 2-3: Causes for Structural Heterogeneity

Causes	Explanations
Naming Conflict	These are of two types. Synonyms are the one which means that two attributes (or entities) that are semantically alike might have different names. Homonyms are the other one which means that two attributes (or entities) that are semantically unrelated might have the same names.
Domain Conflict	Two attributes that are semantically similar might have different domains or data types.
Default Value Conflict	This one depends on the definition of the domain of the concerned attributes. For example, the default value for age of an adult might be defined as 18 in one data source and as 21 in another.
Identifier Conflict	The primary keys of two entities in two sources are incompatible, because they use identifier records that semantically different. For example, the key of student entity might be defined as ID# in one source and as NAME in another.
Integrity Constraint Conflict	Two semantically similar attributes might be restricted by constraints which might not be consistent with each other. For example, the age of adult is defined to over 18 in one source and as to over 21 in another.
Missing Data Item Conflict	This conflict arises when, of the entity descriptors modeling semantically similar entities, one has a missing attribute.
Aggregation Conflict	These conflicts arise when an aggregation is used in one source to identify a set of entities (or attributes) in another source.
Attribute – Entity Conflict	This one arises when the same thing is being modeled as an attribute in one source and an entity in another source.
Data Value – Entity Conflict	It arises when the value of an attribute in one source corresponds to an entity in another source.
Data Value – Attribute Conflict	This conflict arises when the value of an attribute in one source correspond to an attribute in another source.

Those mentioned above are possible structural heterogeneity problems likely encountered while performing the schema integration to construct a global, unified schema to be the foundation of the information integration. It deserves consideration to construct a global schema with correspondences or mappings between the different source schemas to solve the structural

conflicts between them and then gain the interoperability.

Therefore, we can analyze the structural heterogeneity problems between the schemas that are rewritten in XML we want to integrate according to the listed causes of recognition. In the following, we continue to use the example addressed in the previous subsection to explain the process of correspondences identification.

A. Naming Conflict: element "Teacher" using in schema S1 and element "Faculty" using in schema S3 are semantically the same but have different names.
   Correspondence: S1.Teacher = S2.Faculty
B. Aggregation Conflict: the aggregation of element "fname", "mname", and "lname" using in schema S2 is semantically the same.
   Correspondence:
   S1.Teacher.name = S2.Person.name (S2.Person.name.fname +
   S2.Person.name.mname + S2.Person.name.lname)
C. Identifier Conflict: the primary key of entity "Teacher" in source schema S1 is "teacher_id", but the primary key of class "Faculty" in source schema S2 is not specified explicitly, that is "ssn" which is inherited from its parent class "Person".

Afterward, we can specify the integration rules according to the identified correspondences to integrate the independent schemas into the global schema. Continuing the previous example, we can specify the following integration rules:

A. Correspondence: S1.Teacher = S2.Faculty
   Integration rule: G.Faculty
B. Correspondence: S1.Teacher.name = S2.Person.name (S2.Person.name.fname + S2.Person.name.mname + S2.Person.name.lname)
   Integration rule:
   G.Faculty.name (G.Faculty.name.fname + G.Faculty.name.mname + G.Faculty.name.lname)
C. Identifier Conflicts:
   Integration rule:
   Because the semantics of these two identifiers is a little different, we keep them separately in the integrated schema and use "teacher_id" to be the identifier of the integrated element "Faculty".

However, identifying the structural conflicts and correspondences between the independent schemas and specifying the integration rules for our method still needs the intervention of human experts. There are still some research efforts for automatic schema matching (Rahm, & Bernstein, 2001) for producing correspondences between different schemas. Once they are identified, matching elements can be unified under a coherent, integrated schema or viewed by using techniques like schema merge.

2.7. Special Process for the Unstructured Information Sources

Unstructured information sources such as static web pages, multimedia files, etc. do not have "schema", so we must treat such information sources as a special case and provide special

process for them. We create indexes of those sources and do not perform transformation on them. We simply wait until the global schema is created and specify the mapping between them, which will be described in a later section.

2.8. The Creation of Ontology

XML is a representation language for specifying the structure of the underlying information sources and thus their structure dimension. The structural representation can represent some semantic properties but it is not clear how this can be deployed outside of a special purpose application. To allow for a real semantic interpretation for HII, the common data model, XML, must be complemented by a conceptual model that adequately describes the domain we want to perform the information integration. This role cannot be filled by just XML data model (Erdmann, & Decker, 2000).

Using an ontology containing facts and relationships about the application domain of interest as the conceptual model to capture real world knowledge may be a promising approach. However, most ontology creation is carried out on a manual basis. There are a number of publications about ontological development that have been published. Uschold & Grüniger 1996 proposed four main phases when developing ontologies, which are: (1) identifying a purpose and scope, (2) building the ontology: this includes three sub-phases, which are: (a) ontology capturing, (b) ontology coding, (c) integrating existing ontologies. The later two phases are (3) evaluation and (4) guidelines for each phase. Furthermore, Sugumaran & Storey 2002 provided a heuristics-based ontology creation methodology to create the domain ontology.

For our research, we follow the proposed principles to create the needed ontology on a manual basis. We create ontology in order to allow for real semantic interpretation for HII to complement the shortcoming of just using XML in the task of information integration. Besides defining the terms and relationships for the domain in which we perform HII on the ontology, the semantic heterogeneity should also be considered when creating the needed ontology. We recognize the reasons for the semantic heterogeneity problems that the ontology in our research structure wants to handle. Visser et al. (2003) have also categorized the reasons for semantic heterogeneity. We list and explain the reasons for semantic heterogeneities in Table 2-4.

Table 2-4: Causes for Semantic Heterogeneity

Causes	Explanations
Conflicts with Scale and Currency	Two attributes that semantically similar might be represented using different units and measures.
Representation Conflicts	Two attributes are semantically similar, but they might be represented in different formats, for example, school grade: {1, 2, 3, 4, 5} vs. {A, B, C, D, E}
Subjective Mapping Conflicts	The subjective of two attributes is the same, but they are represented in their own styles. For example, German grades: {15, 14, …, 0} vs. American grades: {A, B, C, D, E}
Subsumption Conflicts	The content of an attribute is subsumed by the other one. For

	example, “hotels” includes “congress-hotels”, but the latter, with smaller scope of concept, is only part of the former.
Overlapping Conflicts	Parts of the content of two attributes are the same, but they are not equal to each other. For example, hostels and hotels vs. hotels and camp-sites.
Incompatibilities	The concepts of two attributes are the same, but actual meanings of them are still a little different. For example, hostels and hotels all mean the places for accommodation when traveling, but hostels are cheaper, some are only for youth. In contrast, hotels are more expensive.
Aggregation Conflicts	The concept of two attribute are different, but the concept of one of them is the aggregative concept of the other one. For example, hotel company vs. hotel. Hotel company means a company that operates hotels, but hotel means the place for accommodation when traveling.

We take the above conflicts into consideration when performing the conceptual modeling for the underlying information sources for the creation of the needed ontology. Afterward, we must create a connection between the global schema and the created ontology for the use in our research structure in the following.

Because the ontology defines terms and relationships with axioms of a domain, we view the elements in the global schema as the instance of the resources defined in the ontology. In other words, we use the ontology to define the relationships between the elements in the global schema.

The fragment of the above ontology defines a resource “Faculty” and represents the relationship between resource “Faculty” and resource “Employee” which means faculty must be an employee. We use such definition to define the element “Faculty” in the global schema.

Under such kind of connection between the two different data model, semantics defined in the ontology itself can be appended to the elements in the global schema which is represented in XML format.

We can hardly describe the relationship that a teaching assistant “must be” a graduate student in XML data model. However, such kind of relationship is common in the real world. To complement such a shortcoming of XML, we catch the relationship and define it in the ontology. And then we connect the two data model by defining the element about teaching assistant in the global schema as the instance of the resource in the ontology.

Afterwards, we can obtain the knowledge that a teaching assistant must be a graduate student by inferential against the ontology. Obtaining more knowledge about the real world can help enhance the accuracy and precision of the interoperation between the underlying heterogeneous information sources.

Although an XML data model could represent certain kinds of semantics, for instance inheritance, there are still some relationships that cannot be represented by the XML data model, for instance intersection, union and so on. We use ontology to define the complete relationships

of the domain and then use it to define the relationships between the elements in the global schema by viewing the global schema as an instance of the ontology.

Only while creating the connection between global schema and ontology, we can enable reasoning over the ontology to assist the query against the global schema in the research structure and reach the interoperability of structure and semantics.

2.9. Mapping Global Schema to Local Data Sources

After we create the integrated schema, we still have to consider the mapping between global schema and the local data source schema. Since we view the integration structure as an independent system from the local data source, we must build some bridges between the schemas of the integration system and those local data sources. The mapping is the bridge of the global schema and the local data source schema. However, there have been several proposed approaches to specify the mapping between global schema and local schema, which are global-as-view (GAV), local-as-view (LAV) (Levy, 2000; Manolescu, Florescu, & Kossmann, 2001). The first approach is to define the global schema as a view over the local schemas. In contrast, the second approach is to define the local sources as views over the global schema. The fundamental comparison between these three approaches is presented in Table 2-5.

Table 2-5: Comparison between GAV and LAV

	GAV	LAV
Query Reformulation	Translating the query on the global schema into queries on the local schemas is a simple process of view unfolding.	The query on the global schema needs to be reformulated in the terms of the local data sources' schemas; this process is traditionally known as "rewriting queries using views" and is a known hard problem.
Data Modification	To handle modifications in the local data sources set or in their schemas, the new global schema needs to be redesigned considering the whole modified set of sources.	A local change to a data source can be handled locally, by adding, removing or updating only the view definitions concerning this source.
Data Format	If the local data sources do not have the same data format (e.g. some are relational while others are XML), it would be difficult to define the global schema as a view over sources in different formats.	Each source can be described in isolation, by a view definition mechanism appropriate to its format.

We adopt the GAV approach to specify the mapping between global schema and local data source schema because its query reformulation process is easier than the LAV approaches. Although the evolution process of GAV approach is harder than LAV approach, the query reformulation process is our prior consideration of adopting which approach for specifying the

mapping. We also apply this approach to the unstructured information sources to specify the mapping between the global schema and the index created in the previous section.

Creating the mapping between global schema and local source schema gains the interoperability of different systems. That is also the goal of this research. In the following, we describe how to apply the integrated model in our research structure.

2.10. Query Resolution in Research Structure

In the beginning, the query interface was designed according to XQuery because we used XML for the common data model of our research structure. As a result, users have to formulate their query request in an XQuery form against the integrated schema. Figure 2-6 shows the steps of query processing in the research structure. We explain every step in more details as follow:

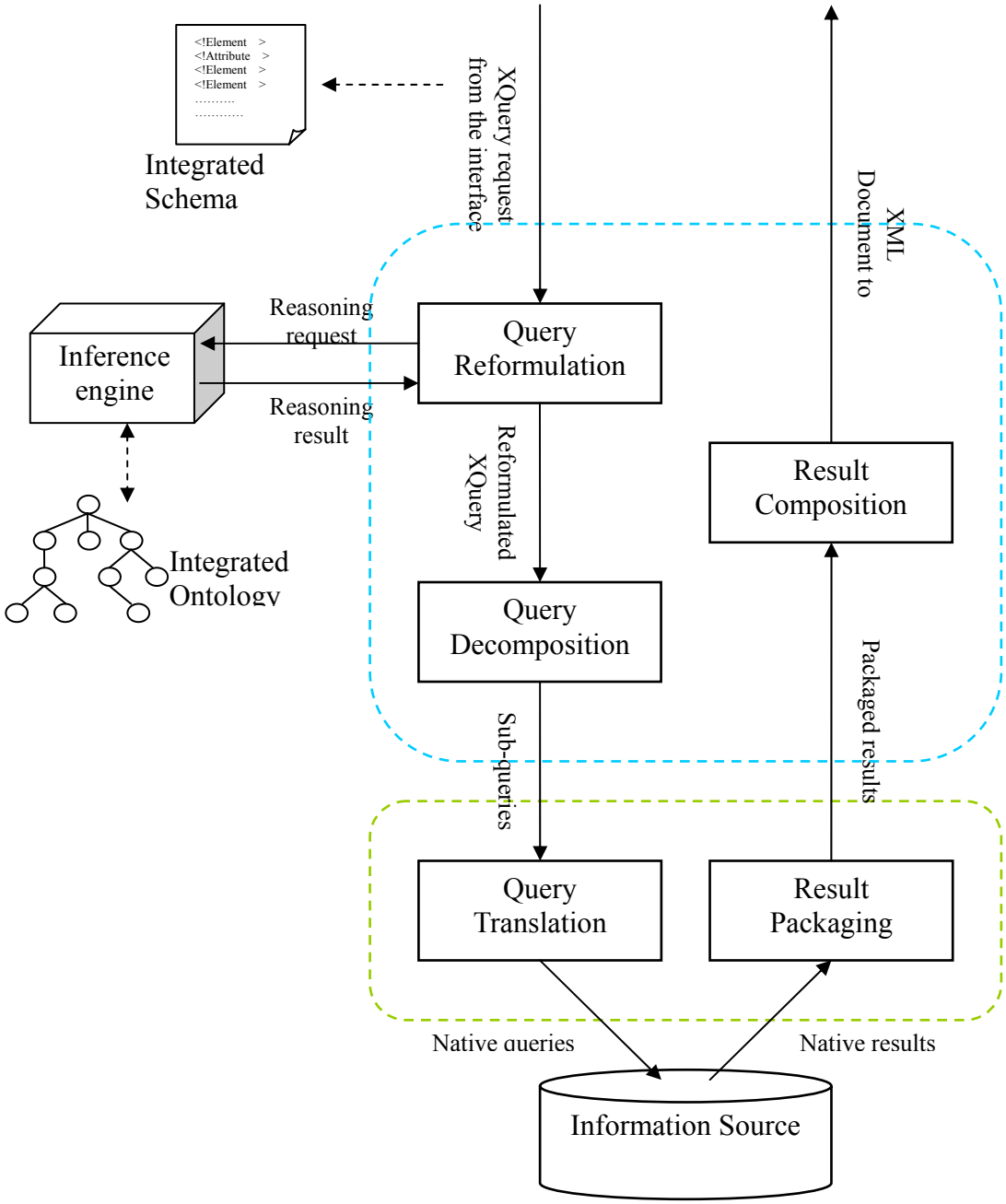


Figure 2-6: Query Processing in Research Structure

A. Query Reformulation:

After a user issues an XQuery request from the query interface against the unified view we provide, this query request would be sent into the mediator, where the query reformulation component in the mediator would receive it. The query reformulation component would pass a reasoning request according to the original XQuery request issued by the user to the inference engine and then inference engine would access the ontology in order to find out the relationships that are implicit in the user's original XQuery request.

Since we want to find out the implicit relationships in the user original XQuery request, we have to take the user query apart. We identify and extract entities in the user query for issuing the reasoning request to the ontology. According to the reasoning request, inference engine can have the reasoning results.

The reasoning results are sent back to the query reformulation component. Hence, the query reformulation component could reformulate the original XQuery request on the basis of the reasoning results. For example, the user might not discover the implicit relationships between the entities specified in the global schema because the global schema just gives the user the sketch of the structure of the underlying information sources. Therefore, the reasoning results can be used to complement the path expression that formulate by the user original that can clear the relationships specified in the user query. Besides, the reasoning results may help to find out much more and related answers of the query by adding new query expression.

Afterward, the query reformulation component would send the reformulated query to the query decomposition component.

B. Query Decomposition:

After the query decomposition component received the reformulated query, the reformulated query would be decomposed with the assistance of the mapping between integrated schema and local source schema. So query decomposition component could use such information to decompose the query into several sub-queries that are respectively applicable to their target information sources. Afterwards, query decomposition component would send these sub-queries to the corresponding wrappers.

C. Query Translation:

The wrapper would then receive the corresponding sub-queries. However, due to the limited capability of the underlying information source, the wrapper would have to translate the generic sub-queries (used in this framework) into native queries (e.g. SQL) according to the capability. Afterwards, such native queries would be issued to the corresponding source in order to find out the data really needed.

To enable query translation, we have to identify the correspondences between XQuery expression used in our research structure and the local source query expression. In this research, we also use the representative heterogeneous information sources that are RDBMS and ODBMS as the explanation. We identify the correspondences between the different query languages and list them in Table 2-6 and Table 2-7.

Table 2-6: The Correspondences between XQuery Expression and SQL Expression

XQuery Expression	SQL Expression
For	No Corresponding function
Let	
Where	Where
Order by	Order by
Return	Select
Aggregation function	Aggregation function
Comparison	Comparison
Path expression	No Corresponding function
included in XPath expression	From
No Corresponding function	Group by
	Having

Table 2-7: The Correspondences between XQuery Expression and OQL Expression

XQuery Expression	OQL Expression
For	No Corresponding function
Let	
Where	Where
Order by	Order by
Return	Select
Aggregation function	Aggregation function
Comparison	Comparison
Path expression	Path expression
included in XPath expression	From
No Corresponding function	Group by
	Having

According to the list of correspondences between the two different query languages, we can find out some kinds of the query expression cannot be completely mapped to another one. However, we just treat the exception of the mapping as a special case and markup it for the further process that might be the requirement for writing additional rules at the execution time.

D. Result Packaging:

After the information source processes those native queries, it would send the native results (e.g. record set) back to the wrapper. Because it is one-on-one between a wrapper and a type of information source, this makes sure that the results would be sent back to the corresponding wrapper. After receiving the results, the wrapper would package the results into a normal form and send it back to the mediator.

E. Result Composition:

It is the query composition component in the mediator that would collect several packaged

results sent back from several wrappers according to the previous decomposed result. And with the aids of the integrated schema, the individual results could be composed in a complete XML document. Finally, it would send the XML document to the interface for the user.

To construct an integration system, we must consider execution rate, completeness of the query result, and consistency of all the underlying data. And the optimization issue must be taken into consideration while performing the query process. However, because the optimization issue is out of our scope, we won't discuss it here in this research.

3. Research Prototype

According to the research method described in prior section, a schema and ontology-assisted heterogeneous information integration prototype system is implemented. This system shows that the integration method of this research is able to obtain the interoperability between multiple heterogeneous information sources. In this section, the platform, architecture, and development of our prototype system is described in the following sections.

3.1. Prototype System Architecture

The prototype system architecture is shown in Figure 3-1. In our implementation, we tackle three kinds of heterogeneous information sources including structured data source, semi-structured data source, and unstructured data source. They respectively are relational database management system, native XML database, and web pages. Microsoft SQL Server 2000 is choose as the structured data source as well as Tamino 4.1.4 native XML Database is choose as the semi-structured data sources. Besides, we also take a web page repository as the unstructured data source.

Users connect to the prototype system which is built on the web server through the Internet by the client side browser. The prototype system receives the request from the user and sends it to the underlying information sources. After the underlying information sources process the requests, the prototype system collects the results and shows them to the user on the browser.

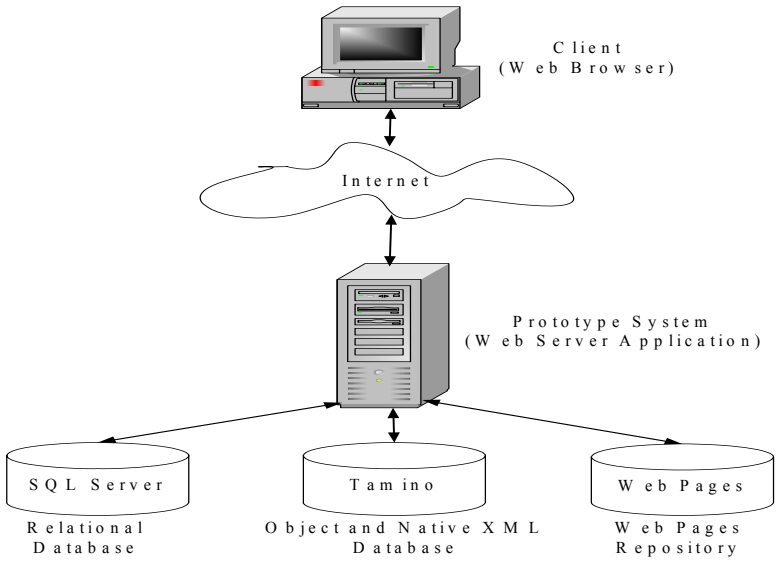


Figure 3-1: The Prototype System Architecture

3.2. Prototype System Platform

In our implementation, we choose Active Server Page (ASP) and Java Server Pages (JSP) as our programming language and Microsoft Windows XP Professional edition as the operating system. We use Microsoft Internet Information Server 6.0 and Tomcat 5.0 as the web server. Microsoft SQL Server 2000 and Tamino 4.1.4 native XML database are the underlying databases. This prototype system uses client/server architecture. Microsoft Internet Explorer 6.0 is chosen as the client side web browser.

We also use Protégé 2.0 with OWL plug-in to edit the required ontology. And we use Jena API from HP Labs as the ontology inference engine to perform the needed reasoning in the prototype system.

3.3. Prototype System Design

Because of the lack of real world cases, we choose an application scenario of a university to implement our prototype system. Under such a scenario, we simulate three kinds of information sources as the required implementation scenario, which are:

A. Structured information source: Relational Database
B. Semi-structured information source: Native XML Database
C. Unstructured information source: Web Pages Repository

After the simulation of the required scenario, we start to create the needed global schema. We first use the generic construct correspondences to transform the schema of the structured information source, the relational database, into the form of XML Schema. Since the form of the schema of the semi-structured information source, the native XML database, is XML Schema, we need not to do the transformation on it. Afterward, we create global schema manually by using the method.

We still must create the required ontology to catch the relationships that can not be hold in the global schema. We follow the research method to create the ontology in OWL by using the ontology editor – Protégé 2.0 with OWL plug-in from Stanford. Figure 3-2 demonstrates the creation of the ontology by means of Protégé 2.0.

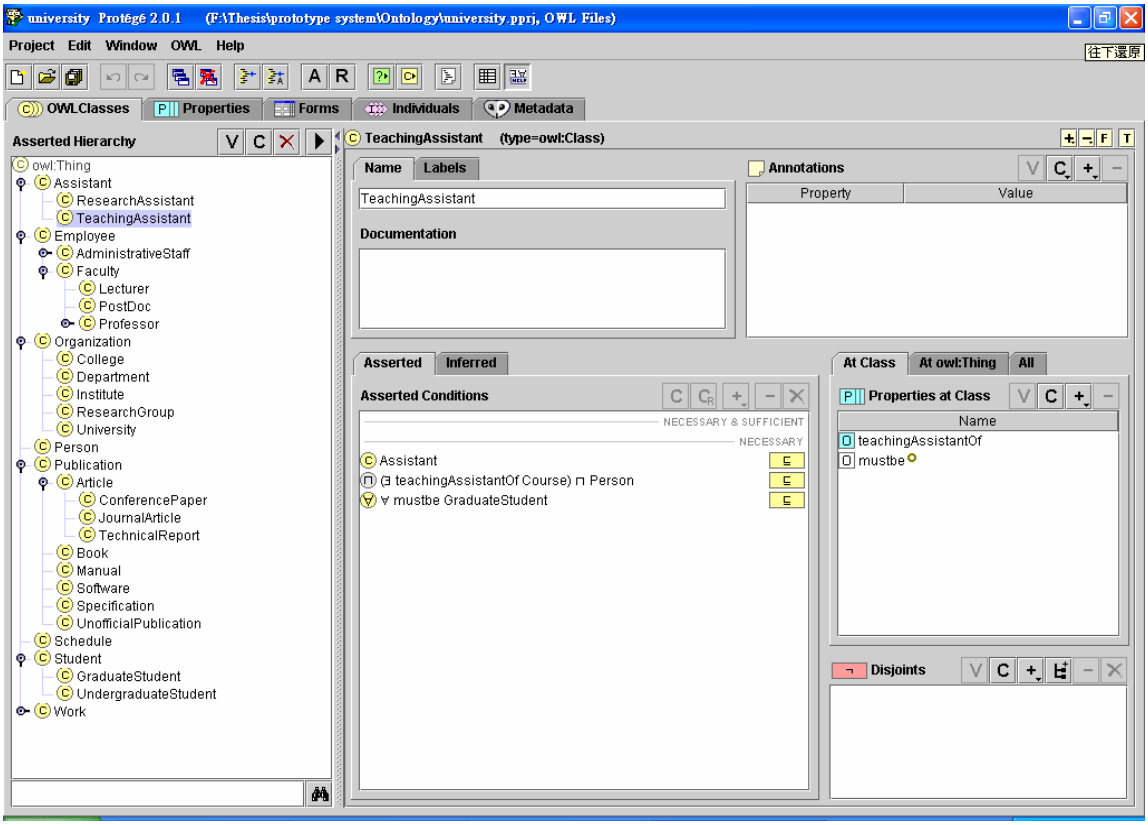


Figure 3-2: Demonstration of the creation of the ontology by means of Protégé 2.0

Figure 3-3 shows the prototype system functions design architecture. We provide an open query interface. Users formulate their query in the form of XQuery expression by referencing the structure described in the global schema. And the individual function of the other two primary query-processing components is described one by one as follows:

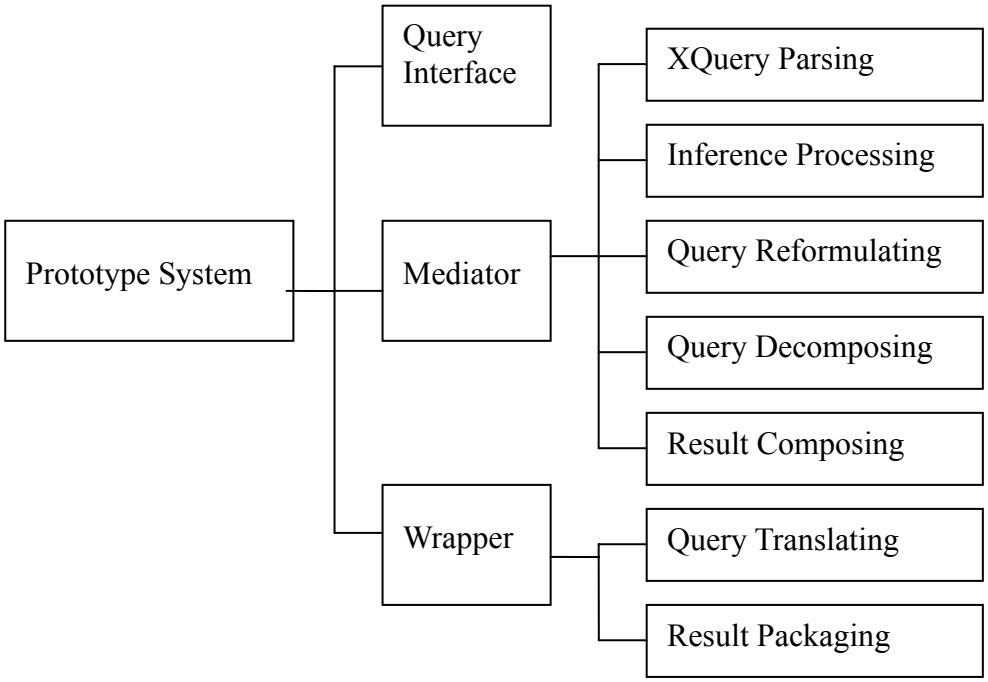


Figure 3-3: Prototype System Functions

A. Mediator:

I. XQuery Parsing:

Because the query interface is designed to accept open query in an XQuery expression, the system needs an XQuery parser to parse the query issued by the user from query interface.

II. Inference Processing:

After receiving the user query from interface, the mediator issues a reasoning request according to the user query to the ontology in order to find out the implicit relationships hidden in the user query.

III. Query Reformulating:

After receiving the reasoning results, the mediator reformulates the user query according to those results. Due to the reformulation process, the system can find out more answers with higher precision and accuracy to the query.

IV. Query Decomposing:

The mediator then arranges the query plan and decomposes the reformulated query into sub-queries according to the specified mapping which is specified according to the GAV approach.

V. Result Composing:

The mediator takes responsibility for composing the final results in XML document format from the temporal results sent by the wrapper. Finally, show the results to the user on the browser.

B. Wrapper:

VI. Query Translating:

The wrapper takes the responsibility of translating the sub-query passed by the mediator into the form of the native query. And then pass the native query into local sources for finding out the needed answers.

VII. Result Packaging:

The wrapper should also collect the query results sent by the local source and sent it back to the mediator to wait for further processing.

3.4. Prototype System Presentation

According to the functions description of the previous section, we demonstrate the implementation of the prototype system as follows:

Figure 3-4 is the query interface of this prototype system.

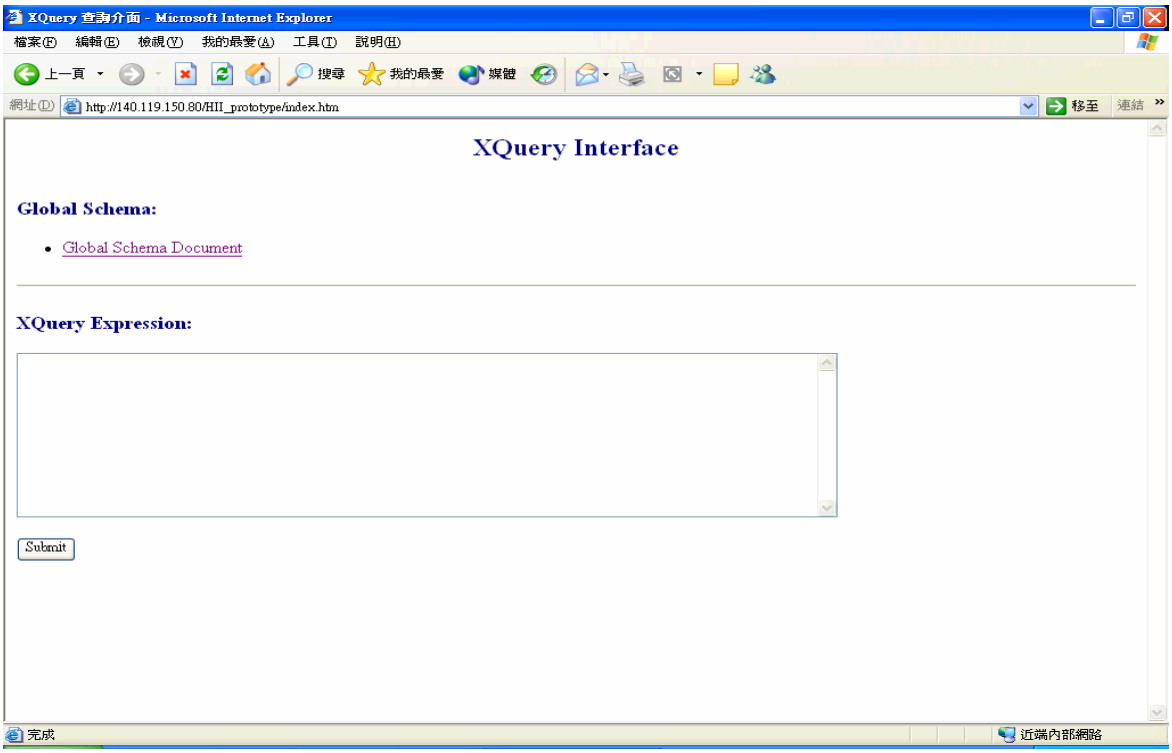


Figure 3-4: Query Interface of the Prototype System

Figure 3-5 illustrates that users formulate the query according to the global schema provided by the prototype system. We give a sample query here to illustrate the query process of the prototype system.

Sample Query: “find out advisors of teaching assistants of course ‘ADB’ ”

XQuery Expression:

```
FOR $a IN document(‘GS.xml’)/GSROOT
LET $b := FOR $t IN document(‘GS.xml’)/GSROOT/TeachingAssistant
          LET $d := FOR $c IN document(‘GS.xml’)/GSROOT/course
                    WHERE $c/cname = ‘ADB’
                    RETURN $c/ta_id
          WHERE $t/ta_id = $d
          RETURN $t/ta_name
WHERE $a//name = $b
RETURN $a//advisor, $b
```

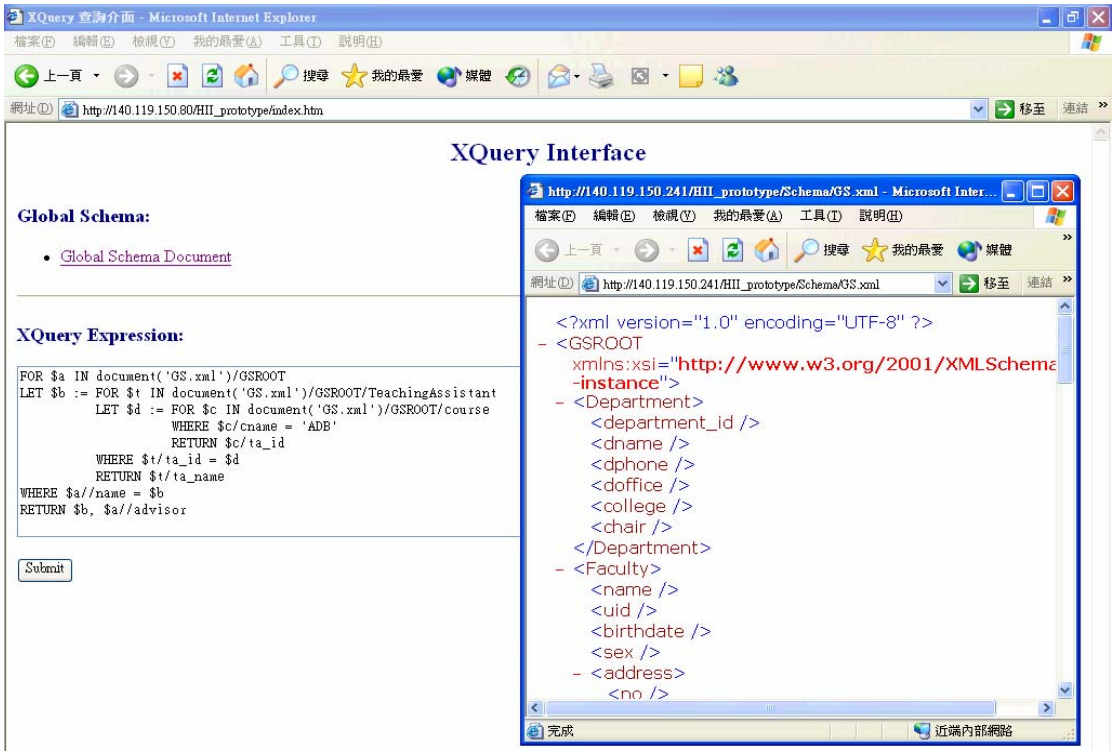


Figure 3-5: Users formulate the XQuery expression of their own queries according to the global schema

After receiving the user query, the system sends a reasoning request to ontology to find out if there have been additional relationships hidden in the user query. Then system reformulates the user query according to the reasoning result. Figure 3-6 shows the reformulated query generated by the system after the query reformulation process. The paths highlight in red color in the reformulated query is complemented by the system automatically according to the reasoning result.

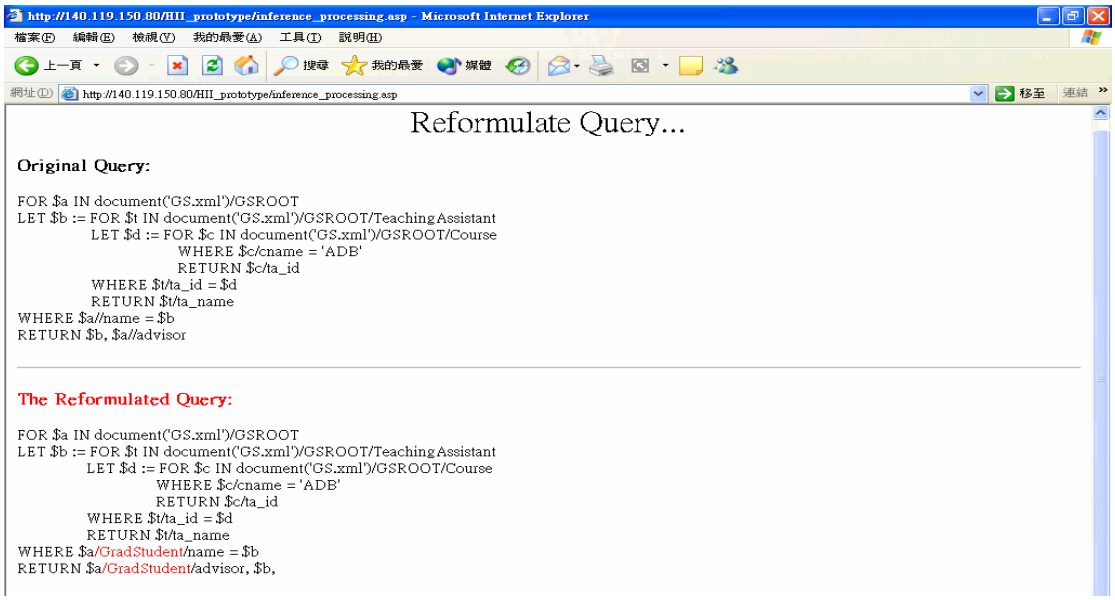


Figure 3-6: The reformulated query

The system decomposes the reformulated query and generates the query plan as the sequential reference of sending the sub-queries to the wrappers. Figure 3-7 shows the query plan

generated by the prototype system.

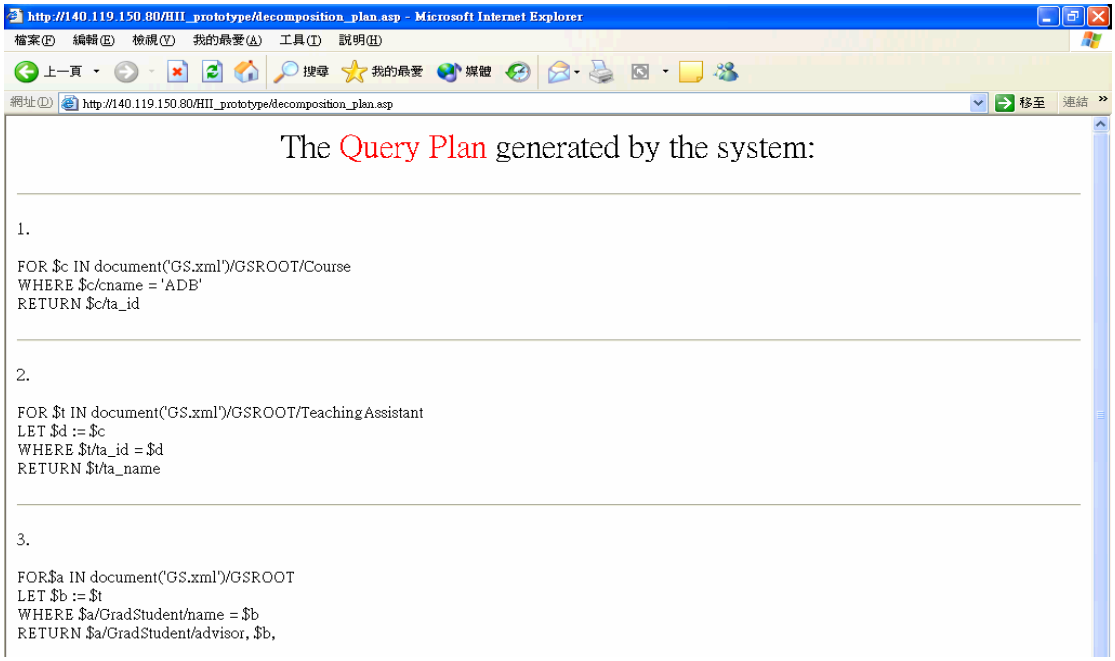


Figure 3-7: The query plan generated by the prototype system

According to the query plan, system passes the sub-queries to the corresponding wrapper. The wrapper takes responsible of translating the sub-query into the corresponding native query. Then the wrapper sends the native query to the underlying source to find out the answer of the query request. Figure 3-8 and 3-9 show the translated query generated by wrappers and the temporal result after querying the underlying source.

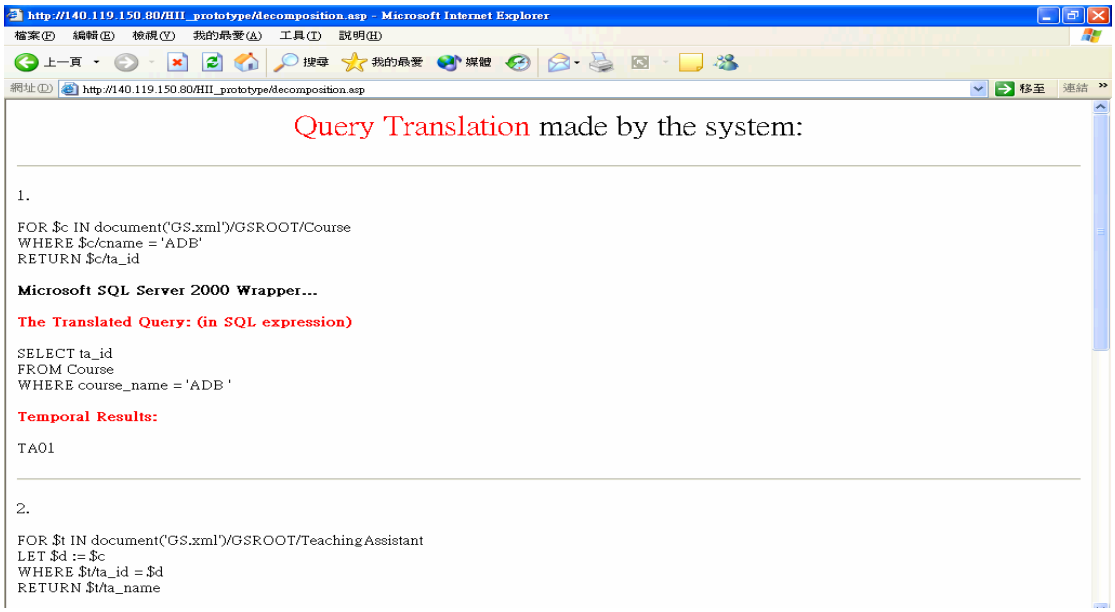


Figure 3-8: The decomposed sub-queries and the translated query generated by wrappers

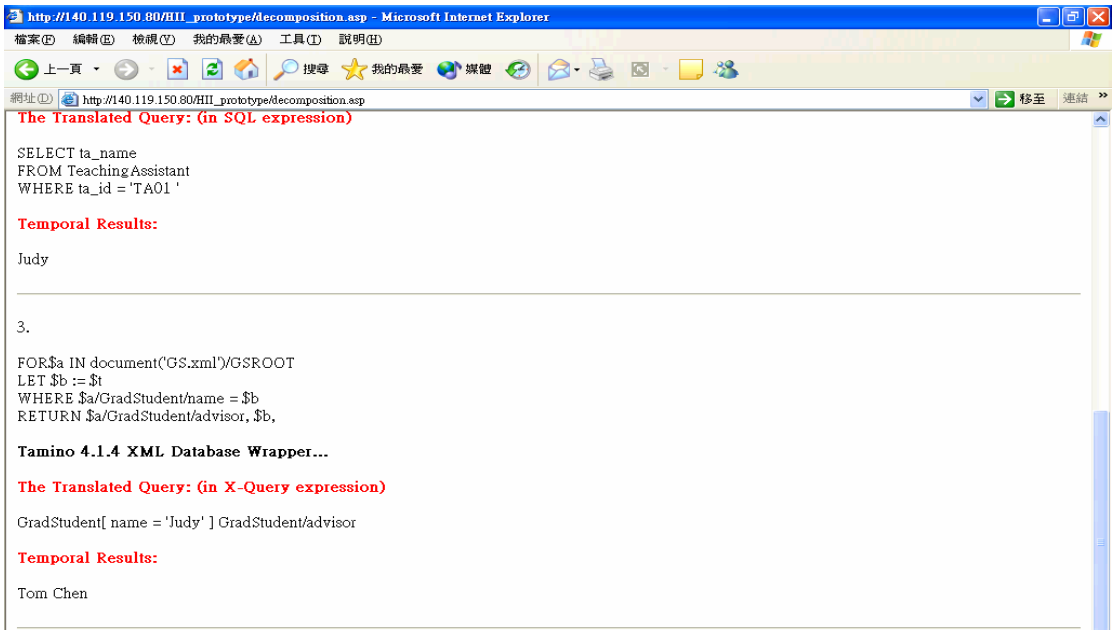


Figure 3-9: The decomposed sub-queries and the translated query generated by wrappers (continued)

After finishing the tasks of the wrappers, the system collects the temporal results and composes them into an XML document to complete the query-processing. Figure 3-10 shows the completeness of the query-processing.

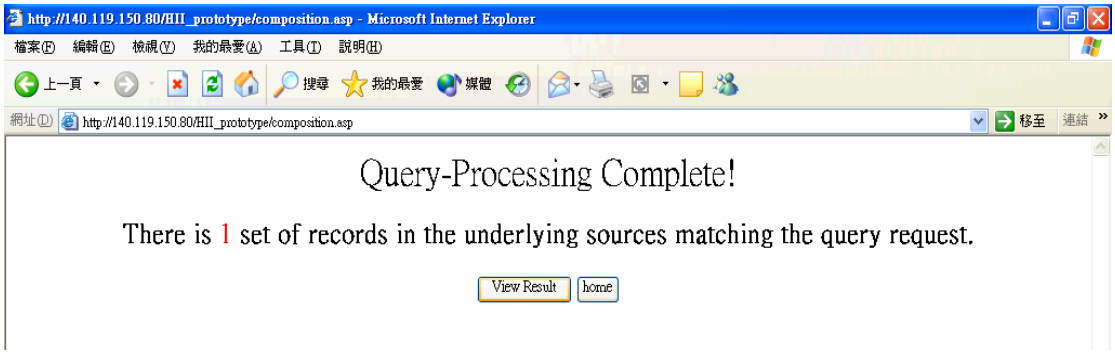


Figure 3-10: Query-processing complete

Figure 3-11 shows the final result in XML document. The structure of the final result is according to the global schema. We add an additional annotation attribute, source, for showing the answer that was retrieved from which information source.

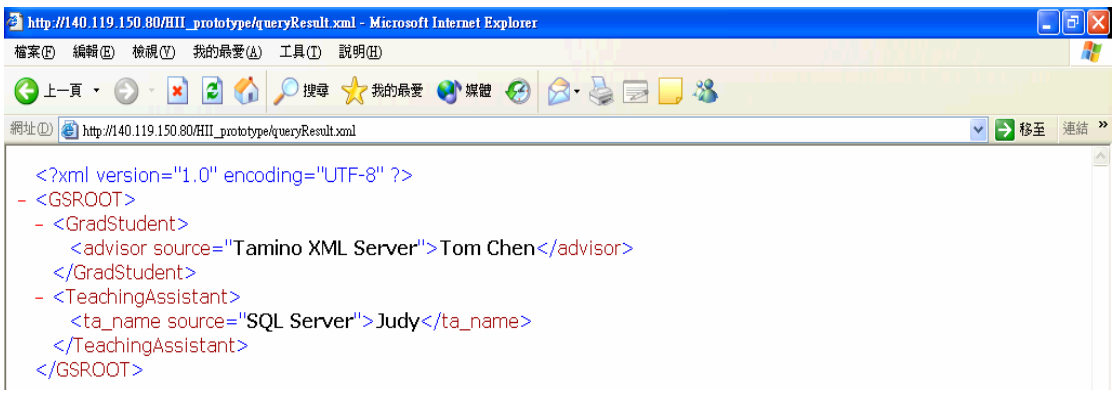


Figure 3-11: The query result in XML document

4. Conclusions and Future Research Directions

This section presents a summary of this research. At the end, a few possible future works are also described.

4.1. Summary

The research issues of heterogeneous information integration have become ubiquitous and critically important in e-business because of the speedy development of information technology and widespread of the Internet. Users in the contemporary company may face the problems of accessing multiple heterogeneous information sources simultaneously. Accessing the heterogeneous information sources separately without integration may lead to the chaos of information requested. It is also not cost-effective. Therefore, the issue of heterogeneous information integration in EB is really of great urgency. In this research, we go deep into such issue and provide our solution by implementing the prototype system as a support. To sum up the research results, the conclusions are described as follows:

A. Provide generic construct orientation method to generate the global schema of the underlying heterogeneous information sources.

The creation of the global schema in the preceding research almost needs human experts give ad hoc specification. We propose a generic construct orientation method to generate the global schema in this research. Under such an approach, it gives application dependence which is important to web-based HII in EB. Although it still requires experts to identify the conflicts between transformed local source schemas, this idea provides a systematic way to generate the needed global schema when performing heterogeneous information integration.

B. Provide a wiser query method over multiple heterogeneous information sources.

In this research, we add ontology to assist querying over multiple heterogeneous information sources. Ontology can catch the relationships that cannot be held in the global schema. By reasoning the relationship described in the ontology, the system finds out the additional implicit meaning of the user query. As such, it can discover more precise and necessary answers regarding the user query. We implement it successfully. Therefore, the query capability is enhanced.

C. Enhance the interoperability between the heterogeneous information sources.

The prototype system implemented in this research indeed enhances the interoperability between the heterogeneous information sources. Through the XQuery interface of the prototype system, users can get the needed and integrated information from different heterogeneous information sources without additional considerations of the complexity of accessing each information source. The contradictory nature of the requested information is reduced through the

information integration process. Interoperability enhancement is our emphatic point.

4.2. Future Research Directions

The prototype for HII with schema and ontology assisted implemented in this research are trying to prove the information integration method we provide and get a satisfactory result. Due to time and resource restrictions, this research still has a number of areas to improve in the future. For example, thoroughly design the evolution process of the information integration method. The automatic or semi-automatic correspondence and conflict identification of the creation process of the global schema may need further study.

In addition, our method is best used in a stable environment, which seems applicable for e-businesses and Intranets. However, the trend toward globalization stresses collaboration between enterprises. The need for information exchange makes the issue of cross-enterprise HII more and more important. So extending the scope of the heterogeneous information integration from the e-business to e-commerce or even collaborative-commerce could be an interesting research direction for further study in the future.

References

Baru, C. K., Gupta, A., Ludascher, B., Marciano, R., Papakonstantinou, Y., Velikhov, P., & Chu, V. (1999). XML-based information mediation with MIX. *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD1999),* 597-599.

Baru, C. K., Ludäscher, B., Papakonstantinou, Y., Velikhov, P., & Vianu, V. (1998). Features and requirements for an XML view definition language: Lessons from XML information mediation. *Position paper, W3C Query Language Workshop (QL'98).*

Carey, M., Hass, L. M., Schwarz, P. M., Arya, M., Cody, W. F., Fagin, R., Flickner, M., Luniewski, A. W., Niblack, W., Petkovic, D., Thomas, J., Williams, J. H., & Wimmers, E. L. (1995). Towards heterogeneous multimedia information systems: The Garlic approach. *5th International Workshop on Research Issues in Data Engineering-Distributed Object Management (RIDE-DOM'95),* 124-131.

Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J., & Widom, J. (1994). The TSIMMIS project: Integration of heterogeneous information sources. *Proceedings of the 10th Meeting of the Information Processing Society of Japan (IPSJ),* 7-18.

Chu, Yu-Chi. (2001). Integrating heterogeneous information sources through ontology-driven model and data quality analysis (Doctoral dissertation, National Taiwan University of Science and Technology, 2001). *Electronic Theses and Dissertations System, 089NTUST428107.*

Cluet, S., Delobel, C., Siméon, J., & Smaga, K. (1998). Your mediators need data conversion. *Proceedings of the ACM SIGMOD Conference of Management of Data.*

Cui, Z., Jones, D., & O'Brien, P. (2001). Issues in ontology-based information integration. *Paper in Joint Session with IJCAI-01 Workshop on Ontologies & Information Sharing*.

Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., & Horrocks, I. (2000). The semantic web: The roles of XML and RDF. *IEEE Internet Computing, 4(5),* 63-74.

Ding, Y., Fensel, D., Klein, M., & Omelayenko, B. (2002). The semantic web: Yet another hip. *Data & Knowledge Engineering, 41(2-3),* 205-227.

Elmasri, R., & Navathe, S. B. (2004). *Fundamentals of database systems.* (4th ed.). Addison-Wesley.

Erdmann, M., & Decker, S. (2000). Ontology-aware XML-queries. *Submission for WebDB 2000.*

Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Vassalos, V., & Widom, J. (1997). The TSIMMIS approach to mediation: Data models and languages. *Journal of Intelligent Information Systems, 8(2),* 117-132.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition, 5(2),* 199-220.

Hass, L. M., Miller, R. J., Niswonger, B., Roth, M. T., Schwarz, P. M., & Wimmers, E. L. (1997). Transforming heterogeneous data with database middleware: Beyond integration. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering.*

Jhingran, A. D., Mattos, N., & Pirahesh, H. (2002). Information integration: A research agenda. *IBM SYSTEMS JOURNAL, 41(4),* 555-562.

Josifovski, V., Schwarz, P., Haas, L., & Lin, E. (2002). Garlic: A new flavor of federated query processing for DB2. *Proceedings of the 2002 ACM SIGMOD international conference on Management of data,* 524-532.

Kashyap, V., & Sheth A. (1996). Semantic and schematic similarities between database objects: A context-based approach. *The VLDB Journal, 5,* 276-304.

Kirk, T., Levy, A., Sagiv, Y., & Srivastava, D. (1995). The information manifold. *Proceedings of the AAAI Spring Symposium on Information Gathering.*

Kuo, W. (2003). A Generic Construct based Transformation Model between UML Data Model and XML (Master Thesis, National Chengchi University, 2003). *Electronic Theses and Dissertations System, 091NCCU5396018.*

Levy, A. Y. (2000). Logic-based techniques in data integration. *Logic Based Artificial Intelligence.*

Levy, A. Y., Rajaraman, A., & Ordille, J. J. (1996). Querying heterogeneous information sources using source descriptions. *Proceedings of the Twenty-second International Conference on Very Large Databases*, 251-262.

Mena, E., Illarramendi, A., Kashyap, V., & Sheth, A. P. (2000). OBSERVER: An approach for query processing in Global Information Systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases, 8(2),* 223-271.

Manolescu, I., Florescu, D., & Kossmann, D. (2001). Answering XML queries over heterogeneous data sources. *Proceedings of the 27th VLDB Conference*.

Manolescu, I., Florescu, D., Kossmann, D., Xhumari, F., & Olteanu, D. (2000). Agora: Living with XML and relational. *Proceedings of the 26th VLDB Conference*.

Miller, R. J., Hernández, M. A., Haas, L. M., Yan, L., Ho, C. T. H., Fagin, R., & Popa, L. (2001). The Clio project: managing heterogeneity. *ACM SIGMOD Record, 30(1),* 78-83.

Parent, C., & Spaccapietra, S. (1998). Issues and approaches of database integration. *Communications of ACM, 41(5),* 166-178.

Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal, 10,* 334-350.

Roddick, J. F. (1995). A survey of schema versioning issues for database systems. *Information and Software Technology, 37(7),* 383-393.

Roth, M. T., Arya, M., Hass, L., Carey, M., Cody, W., Fagin, R., Schwarz, P., Thomas, J., & Wimmers, E. (1996). The Garlic project. *In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data,* 557.

Sugumaran, V., & Storey, V. C. (2002). Ontologies for conceptual modeling: their creation, use, and management. *Data & Knowledge Engineering, 42(3),* 251-271.

Tomasic, A., Amouroux, R., Bonnet, P., Kapitskaia, O., Naacke, H., & Raschid, L. (1997). The Distributed Information Search Component (Disco) and the World Wide Web. *ACM SIGMOD.*

Tomasic, A., Raschid, L., & Valduriez, P. (1998). Scaling access to distributed heterogeneous data sources with DISCO. *Proceedings of the IEEE Transactions on Knowledge and Data Engineering.*

Uschold, M., & Grüniger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review, 11(2),* 93-136.

Vdovjak, R., & Houben, G. (2001). RDF-based architecture for semantic integration of heterogeneous information sources. *Proceedings of the Workshop on Information Integration on the Web 2001,* 51-57.

Visser, U., Stuckenschmidt, H., & Wache, H. (2003). *Ontology-based information integration.* IJCAI-Tutorial SP5. http://www.cs.vu.nl/~heiner/IJCAI-03/Tutorial (Data Accessed: January 7, 2004)

Wache, H., Vögele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001). Ontology-based integration of information ─ A survey of existing approaches. *Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing.*

Wiederhold, G. (1993). Intelligent integration of information. *ACM SIGMOD Conference on Management of Data,* 434-437.

Internet References

TSIMMIS: http://www-db.stanford.edu/tsimmis/tsimmis.html
DISCO: http://www-caravel.inria.fr/Eprototype_Disco.html
Garlic: http://www.almaden.ibm.com/cs/garlic/
MIX: http://www.npaci.edu/DICE/mix-system.html
Agora: http://www-rocq.inria.fr/~manolesc/AGORA/index.html
OBSERVER: http://sol1.cps.unizar.es:5080/OBSERVER/
ONTOBROKER: http://ontobroker.aifb.uni-karlsruhe.de/index_ob.html
HERA: http://wwwis.win.tue.nl/~hera/
W3C: http://www.w3.org
XQuery: http://www.w3.org/XML/Query
XML Schema: http://www.w3.org/XML/Schema
RDF: http://www.w3.org/RDF/
OWL: http://www.w3.org/2001/sw/WebOnt/
Jena: http://jena.sourceforge.net/
Protégé: http://protege.stanford.edu
APA Style Essentials: http://www.vanguard.edu/faculty/ddegelman/index.cfm?doc_id=796#title
Other :http://www-ksl.stanford.edu/kst/what-is-an-ontology.html