International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

# Heterogeneous Database Semantic Integration Based on Ontology

**Heping Chen** [1,*], **Bin Chen** [1], **Lu He** [1], **Ping Liang** [2]

[1] *School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China*

[2] *School of Computer, Huazhong University of Science and Technology, Wuhan 430074, China*

**Abstract:** To enable accessing deep web information at semantic level, this paper develops a semantic query rewriting and planning mechanism on heterogeneous database enabled web information system with complex ontology mapping technology. It discusses the procedure of converting database schema and instances to ontologies at first, and then the patterns of complex ontology mappings , and then the ontology-based query planning in Mediator-Wrapper based environment with GAV style querying request. The relational algebra rewriting and planning algorithm is discussed in detail.

**Keywords:** SQL Query Rewriting, Ontology Fusion, Complex Semantic Mapping, Semantic Similarity

## 1. Introduction

We witness a rapid increase in the number of web information sources that are available online. the World-Wide Web (WWW), in particular, is a popular medium for interacting with such sources[1]. While the surface Web has linked billions of static HTML pages, a far more significant amount of information is believed to be "hidden" in the deep Web, behind the query forms of searchable databases[2], which means that the integrated querying of distributed database systems still plays a key role in web information age. Problems that might arise due to heterogeneity of the data are already well known within the distributed database systems community: *structural heterogeneity* and *semantic heterogeneity*. Structural heterogeneity means that different information systems store their data in different structures. Semantic heterogeneity considers the content of an information item and its intended meaning[3]. How to accessing distributed information with a consistent semantic environment and how to make the structural query mechanism with semantic enabled are the main problems that should be discussed.

The use of ontologies for the explication of implicit and hidden knowledge is a possible approach to overcome the problem of semantic heterogeneity. Ontologies can be used to describe the semantics of the sources and to make the content explicit. With respect to the data sources, they can be used for the identification and association of semantically corresponding information concepts.

This paper focuses on how to use ontology technology to enable semantic level querying on heterogeneous database systems. The procedure can be described as follows:(1) Building local Ontology from the database schema and instances. (2) Using ontology mapping technology to get a consistent semantic environment, and (3) extending the querying technologies to enable semantic querying on distributed database systems. The remainder of this paper is structured as follows. Section 2 gives the general discussion about ontology enabled heterogeneous database systems. Section 3 discusses the procedure of converting local database schema and instances to local Ontology. Section 4 defines the patterns of ontology mapping, and discusses semantic fusion mechanism. Section 5 discusses ontology enabled querying rewriting and planning. Section 6 introduce a prototype system named OBSA[4] which employed the technology discussed in this paper. Section 7 summarizes the

* Corresponding author.
*Email address: chp@wust.edu.cn.*

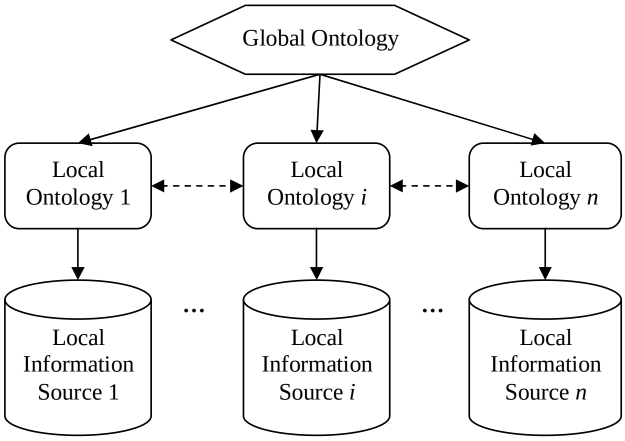


Figure 1. A mediator-based approach for Ontology Integration.

whole paper.

## 2. Semantic enabled Distributed Database Systems

In a distributed environment, every local site contains a structured or semi-structured ontology-based information source. From the view of information integration, all the local information sites can be expressed as the collections of database enabled information systems (DISs), which means it can support structural query interface. By an database enabled information system we mean $S = (\{I_d\}, W)$, where $\{I_d\}$ is a finite set of database instances, $W$ is the ontology based wrapper or mediator[4]. In this paper, we employ ontology to define semantic wrappers or mediators, which are used for the explicit description of the information source semantics. By an integrated database enabled information systems we mean $PS = (\{S_i\}_{i \in I}, M)$, where $I$ is a set of sites, $S_i$ is an DIS for any $i \in I$, $M$ is the mapping relation on the set $I$ which can be expressed as $M : (S_1, S_2, \ldots, S_n) \rightarrow S_0$ while $S_i$ $(1 \leq i \leq n)$ denotes the local DIS sites, $S_0$ denotes the global DIS site acted as the mediator site, $M$ denotes an integrated procedure.

This paper employs a mediator based approach to use ontology in information integration system, which can be expressed as figure 1. In this approach, the semantics of each source is described by its own ontology, a global shared ontology is built to make the local ontologies comparable with each other. The advantage of the mediator-based approach is that new sources can easily be added without modifying the mediator site. It also supports the acquisition and evolution of ontologies. The use of a shared global ontology makes the source ontologies comparable and avoids the disadvantages of multiple ontology or single ontology approaches'[3]. In this paper we will focus on the integration of local ontology to the global ontology, and discusses the semantic querying with the global and integrated ontologies. Mapping discovery between different ontologies is not the main topic we need to discuss in this paper, how to find similarities between them, how to determine which concepts and properties represent similar notions, and so on are the motivations for the researchers who focusing on this topic, a survey about this topic is discussed in paper [5].

## 3. Converting Relational Database to Local Ontology

### 3.1. *Extraction database logic model*

Design of database logic model is a part of database design. Extraction is the revering process of design, so this paper adopts the reverse engineering technology to get logic model from database.

The basic principle of reverse-engineering is extracting software system's branch, otherwise hiding the detail. Then the entities extracted are used to describe the software system on the high level. Rational Rose is one kind of object-oriented visualization modeling tool with formidable reverse-engineering function[6]. It can connection database such as DB2, SQL Server, Oracle .etc, induct "Schema" and then produce the database logic model. The process is easy and quick, as soon as the database is inducted in Ration Rose, the transformation operation will automatically run, which adds database module in the module view, and produces the database logic model in the logic view with the tree structure. By unfolding the tree structure, the logic model graph is clearly showed in time. From the logic tree structure and graph, the logic relationships among various tables and views are extracted.

### 3.2. *Converting Mechanism*

Through analyzing, the formal corresponding relationships between relational databases and OWL[7] ontology are as follows: a relational database contains several tables, a table contains several fields and records are the collection of fields' value; on the other hand, OWL ontology contains several classes, a class contains several properties and instances are the collection of property value. The formal corresponding relationships between tables, fields and records in relational databases and classes, properties and instances in OWL ontology make it possible to convert one kind of model to another[8][9]. According to the two aspects of representation format[10] based on OWL, the conversion mechanism includes the schema conversion and data conversion.

3.2.1. *Schema conversion*

There are ten conversion rules or so from database schema (database logic model) to OWL ontology schema, for saving space, author only gives part of the rules (the identifiers and functions involved are shown in Table 1:

(i) Each table $T$ in a relational database is converted to an OWL class. The class is named after the table; the table description is correspondingly converted to the class comment. That is:
$\forall T \in RDB \rightarrow Class(ID(T), Cmt(T))$.

(ii) When two tables $T$ and $T_{sub}$ in a relational database take part in FCR, the class converted from $T_{sub}$ is declared as a subclass, and the one converted from T is declared as super class in OWL. That is:
$\forall T, T_{sub} \in RDB \wedge Sub(T_{sub}, T) \rightarrow$
$SubClassOf(ID(T_{sub}), ID(T))$.

(iii) Given a table T in a relational database with a column F, a non-foreign key column is converted to data type property in OWL, the property is named after the column ,and column description is converted to property comment .Property domain is specified as the class converted from T, range is specified as the data type of F. That is:
$\forall T \in RDB \wedge \forall F \in Field(T_f)$
$\wedge \neg IsFKey(F, T) \rightarrow$
$DatatypeProperty(ID(F), domain(ID(T)),$
$range(datatype(F)), Cmt(F))$.

(iv) Given two tables $T_f$ and $T_p$ , $T_f$ is associated with $T_p$ by its foreign key $F$ which is converted to the object property of the same name in OWL schema. Property domain is specified as the class converted from $T$, range is specified as the class converted from $T_p$. That is:
$\forall T_f, T_p \in RDB \wedge \forall F \in Field(T_F)$
$\wedge IsFKey(F, T_f) \wedge Relation(F, T_f, T_p) \rightarrow$
$ObjectProperty(ID(F), domain(ID(T_f)),$
$range(ID(T_p), Cmt(F)))$.

3.2.2. *Data conversion*

On the basis of schema conversion, the database data can be converted to OWL ontology instances easily. Data conversion is much more direct compared to schema conversion. The process of data conversion includes the follow three steps:

Table 1
Identifiers and functions involved in the conversion rules

| **Name** | **Description** |
|---|---|
| RDB | collection of tables that belong to relational database |
| Field(T) | collection of columns that belong to table T |
| IsFKey(F,T) | Whether column F is the foreign key of table T or not |
| Sub(T,T') | Table T' and T take part in FCR |
| Relation(T,T',F) | Table T is associated with T' by its foreign key F |
| ID(T/F/V) | Class/Property/Instance Name |
| Cmt(T/F) | Class/Property/Instance Comment |
| Class(ID,Cmt) | Class definition |
| SubClassOf(C,C') | Class C' is defined as subclass of class C |
| DatatypeProperty (ID,D,R,Cmt) | Data type property definition |
| domain(C) | Domain with the value of class C |
| range(C) | Range with the value of class C |

(i) Mapping each database record to an OWL ontology instance and assign a unique identifier to each instance. Author takes primary key of each record as the unique identifier of corresponding instance considering that primary key is unique.

(ii) Mapping the non-foreign key property value of database to DatatypeProperty value of OWL ontology instance.

(iii) Foreign keys correlate two database tables, so foreign keys can be used to correlate two instances, mapping a foreign key property value to an instance, whose ObjectPorperty value is exactly the instance from the correlating primary key.

## 4. Semantic Fusion based on Ontology

In this section, we first discuss semantic integration in general, and then discuss the semantic fusion based on complex similarity mapping and mapping patterns.

### 4.1. *A general discussion about semantic fusion*

Different approach uses different method to map and integrate ontology, IF-Map[11] and FCA-Merge[12] are the most mature methods which have been accepted in knowledge management community widely, this paper focuses on a mediator-wrapper based distributed environment, just like FCA-Merge approach, it uses the bottom-up mapping method to integrate different local ontologies. Most of the approaches do not define the ontology mapping patterns, and they can not define the relationship between the local ontologies and the integrated global ontologies with formal method, and this is the reason why they can not apply ontology integration to support semantic level query rewriting, MBL[13] approach defines the mapping patterns, but it uses logic mapping method, and does not discuss how to combine the ontology reasoning with structured or semi-structured information query rewriting. HOME[14] and TOSS[15] are the most similar approach with our proposed approach, but they only support one-one mapping and the query rewriting algorithm is simple. Another problem is the definition of ontology mapping, it is not flexible, and can not express the relationship between mapping or integrated ontologies formally, which makes users can not use the global ontology to enable semantic level query information over integrated systems.

### 4.2. *Semantic Fusion based on the patterns*

The patterns of semantic fusion can be categorized into four kinds of ontology mapping[5]: direct mapping, subsumption mapping, composition mapping and decomposition mapping[13], a mapping can be defined as:

**Definition 1** *A **Ontology mapping** is a structure $\mathcal{M} = (\mathcal{S}, \mathcal{D}, \mathcal{R}, v)$, where $\mathcal{S}$ denotes the concepts of source ontology, $\mathcal{D}$ denotes the concepts of target ontology, $\mathcal{R}$ denotes the relation of the mapping and $v$ denotes the confidence value of the mapping, $0 \leq v \leq 1$.*

A direct mapping relates ontology concepts in distributed environment directly, and the cardinality of direct mapping could be one-to-one. A **subsumption mapping** is a 6-tuple $\mathcal{S}_\mathcal{M} = (\mathcal{D}_m, \mathcal{R}_m, \mathcal{B}_m, \preceq_m, \mathcal{I}_m, v)$, where $\mathcal{D}_m$ is a direct mapping expression; $\mathcal{R}_m$ is the first target concept, which is the most specialized ontology concept. The mapping between the source ontology and $\mathcal{R}_m$ is denoted as **Root ontology concept mapping**; $\mathcal{B}_m$ is the last target concept, which is the most generalized ontology concept. The mapping between the source ontology and $\mathcal{B}_m$ is denoted as **Bottom ontology concept mapping**; $\preceq_m$ is inclusion relation between target ontology concepts; $\mathcal{I}_m$ is the inverse mapping. Subsumption mapping is used to denote concept inclusion relation especially in the multiple IS-A inclusion hierarchy. The **composition mapping** is a 4-tuple $\mathcal{C}_\mathcal{M} = (\mathcal{F}_m, \mathcal{A}_m, \mathcal{B}_m, v)$, where $\mathcal{F}_m$ is a direct mapping expression; $\mathcal{A}_m$ is chaining of role(s) between target ontology concepts; $\mathcal{B}_m$ is the last target symbol, which is the node of chaining target role(s), and composition mapping is used to map one concept to combined concepts. For example, the mapping *address*=*contact*(*country, state, city, street, postcode*) is a composition mapping, in which the concept *address* is mapped to combined concept "*contact, country, state, street, and postcode*" of local schema elements. The **decomposition mapping** is a 4-tuple $\mathcal{C}_\mathcal{M} = (\mathcal{A}_m, \mathcal{B}_m, \mathcal{L}_m, v)$, where $\mathcal{A}_m$ is chaining of role(s) between source ontology concepts; $\mathcal{B}_m$ is the last target symbol, which is the node of chaining source role(s); $\mathcal{L}_m$ is a direct mapping expression. Decomposition mapping is used to map a combined concept to one local concept, and the example for the decomposition mapping is the reverse of the composition. These four mapping patterns was discussed in paper [4]

This paper defines some properties of semantic mapping which are useful in the task of semantic query planning. The first property is transitivity, for the mapping $\mathcal{M}_{i-1,i} = (C_{i-1}, C_i, \mathcal{R}, v_{i-1,i})$ and $\mathcal{M}_{i,i+1} = (C_i, C_{i+1}, \mathcal{R}, v_{i,i+1})$, a new mapping $\mathcal{M}_{i-1,i+1} = (C_{i-1}, C_{i+1}, \mathcal{R}, v_{i-1,i+1})$ can be created to satisfy the mapping relation $R$. The second property is symmetric, which means that the mapping $\mathcal{M} = (\mathcal{S}, \mathcal{D}, \mathcal{R}, v)$ is equal to the mapping $\mathcal{M}' = (\mathcal{D}, \mathcal{S}, \mathcal{R}, v)$. The third property is strong mapping property, it can be described as follows.

**Definition 2** *A set of mappings $\mathcal{M}_i$ $(0 \leq i \leq n)$are* ***strong*** *if they can satisfy the following conditions:*

*i). They share the same mapping relation R, and the mapping relation is transitivity;*

*ii). For $\forall(i,j,k)$,$v_i, v_j, v_k$ are the confidence value of mapping $\mathcal{M}_i, \mathcal{M}_j, \mathcal{M}_k$, then $v_i \leq v_j + v_k$.*

## 5. Semantic Query Rewriting and Planning

The semantic query in a mediator-based DIS can be express as figure 2. Each data source uses its local wrapper to describe its semantics and its mapping relationship with other nodes. The semantic information is described with the language based on its ontology, and constructs the global semantics in the mediator environment based on ontology via ontology fusion mechanism. The user's request is rewritten and modified accordingly based on the global semantics, and is due processed optimally. Corresponding operation plan is made and passed by the wrapper to each data source node for operation. From above description, we know that this paper employs the GAV(Global as View) method to process the user's query[1]. The knowledge stored at mediator supply a global seman-

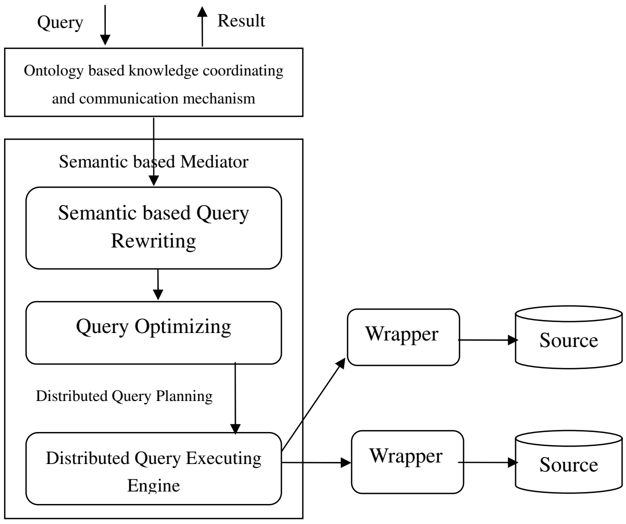


Figure 2. Semantic Querying in a Mediator-based WIS

tic view of the wrappers, which can be described as the ontology fusion connections list, and it can be described as[4]:

**Definition 3** ***Fusion Connection*** *is a structure* $\mathcal{F}_c(O_1 : C_1, O_2 : C_2, \ldots, O_n : C_n, \mathcal{M})$*, where* $C_1$ *denotes a concept or concept set of ontology* $O_1$*,* $C_2$ *denotes a concept or concept set of Ontology* $O_2$*,* $\mathcal{M}$ *denotes the mapping relationship between* $C_1$ *,* $C_2$ *, … and* $C_n$*.*

As has been mentioned above, the mapping patterns are direct mapping, subsumption mapping and composition mapping, the fusion connection can be described as $\mathcal{F}_{cd}$, $\mathcal{F}_{cs}$ and $\mathcal{F}_{cc}$ respectively. The query can be described as a structural query with semantic enhanced, which can be described as an extension of relational algebra.

In order to simplify the discussion, this paper just pays attention to the query planning mechanism of the selection operation. Briefly, a selection operation can be expressed as $\sigma(X : S, Y)$ $\{X \subseteq P_i \cup P_o, Y \subseteq PE\}$, where $P_i$ is the input query pattern, $P_o$ is output query pattern, $PE$ is predication list, $S$ denotes the site in which the query will be executed. We define two operators $\cup$ and $\bowtie$ to represent $Union$ and $Join$ operation separately, and define the operator $\Rightarrow$ to represent the query rewriting operation, and we use $\sigma(X : S_0, Y)$ or $\sigma(X, Y)$ to denote the user's query from the mediator site.

Firstly, we propose how to rewrite pattern tree (which is the $X$ element of expression $\sigma(X, Y)$), there maybe several cases as follows:

(i) X is one of the elements of input query pattern or output query pattern, and it is also a concept in the global ontology hierarchy. $X_i (1 \le i \le n)$ are the concepts for different local ontologies. $X$ and $X_i$ were combined into one concept in

**Algorithm 1**: SQLRWPlan($\sigma(X, Y), FL$)

**Input**: $\sigma(X, Y)$ is the query needed to be processed, $FL$ is the fusion connection list.
**Output**: $P$ is the query planning sequence

$P \leftarrow \emptyset$, $S_q \leftarrow \emptyset$;
**foreach** $x \in X$ **do**
  **switch** *Mappings of* $X$ *node in fusion list* $FL$ **do**
    **case** $directfusion$
      $P \leftarrow P + (\sigma(x, Y), \{\sigma(x, Y), \sigma(x_1 : S_1, Y), \sigma(x_2 : S_2, Y), \ldots, \sigma(x_n : S_n, Y)\}, \cup)$;
    **case** *subsumption or composition*
      $P \leftarrow P + (\sigma(x, Y), \{\sigma(x_1 : S_1, Y), \sigma(x_2 : S_2, Y), \ldots, \sigma(x_n : S_n, Y)\}, \cup)$;
    **end**
  **end**
  $S_q \leftarrow S_q + \sigma(x_1 : S_1, Y) + \sigma(x_2 : S_2, Y) + \ldots + \sigma(x_n : S_n, Y)$;
**end**
**foreach** $\sigma(x, Y) \in S_q$ **do**
  **foreach** $y \in Y$ **do**
    **switch** *Mappings of* $Y$ *concept in fusion list* $FL$ **do**
      **case** $directfunsion$
        $P \leftarrow P + (\sigma(x, y), \{\sigma(x, y), \sigma(x : S_1, y_1), \sigma(x : S_2, y_2), \ldots, \sigma(x : S_n, y_n)\}, \cup)$;
      **case** *subsumption*
        $P \leftarrow P + (\sigma(x, y), \{\sigma(x : S_1, y_1), \sigma(x : S_2, y_2), \ldots, \sigma(x : S_n, y_n)\}, \cup)$;
      **case** *decomposition*
        $P \leftarrow P + (\sigma(x, y), \{\sigma(x, y_1 \wedge F), \sigma(x, y_2 \wedge F), \ldots, \sigma(x, y_n \wedge F)\}, \bowtie, F)$;
      **end**
    **end**
  **end**
**end**
return $P$;

the integrated global ontology with strong direct mappings, which means that $X$ and $X_i$ can match each other, then we can rewrite $X$ as $X \cup \bigcup_{1 \le i \le n} X_i$. The responding selection rewriting can be expressed as:

$$\sigma(X, Y) \Rightarrow \sigma(X, Y) \cup \sigma(X_1 : S_1, Y) \cup \sigma(X_2 : S_2, Y) \ldots \cup \sigma(X_n : S_n, Y) \quad (1)$$

(ii) The concept of $X$ is generated by the subsumption mapping or composition mapping of $X_i (1 \le i \le n)$, then we can rewrite $X$ as $\bigcup_{1 \le i \le n} X_i$. The responding selection rewriting can be expressed as:

$$\sigma(X,Y) \Rightarrow \sigma(X_1 : S_1, Y) \cup \sigma(X_2 : S_2, Y) \\ \ldots \cup \sigma(X_n : S_n, Y) \quad (2)$$

And then, we propose how to rewrite the predication expressions (which is the Y element of the expression $\sigma(X,Y)$, there are also several cases, which can be described as follows:

(i) If there are lots of concept $Y_i (1 \le i \le n)$ combined in the concept $Y$ of global Ontology, we can rewrite $Y$ as $Y \cup \bigcup_{1\le i\le n} Y_i$. The corresponding selection rewriting can be described as:

$$\sigma(X,Y) \Rightarrow \sigma(X,Y) \cup \sigma(X : S_1, Y_1) \\ \cup\sigma(X : S_2, Y_2) \ldots \cup \sigma(X : S_n, Y_n) \quad (3)$$

(ii) If the concept $Y$ is generated by the subsumption mapping of $Y_i (1 \le i \le n)$, we can rewrite $Y$ as $\bigcup_{1\le i\le n} Y_i$. The corresponding selection rewriting can be described as:

$$\sigma(X,Y) \Rightarrow \sigma(X : S_1, Y_1) \cup \sigma(X : S_2, Y_2) \\ \ldots \cup \sigma(X : S_n, Y_n) \quad (4)$$

(iii) If the concept $Y$ is generated by the composition mapping of $Y_i (1 \le i \le n)$, suppose the composition condition is $F$, we can rewrite $Y$ as $(Y_1 + Y_2 + \ldots Y_n) \cap F$. The corresponding selection rewriting can be described as:

$$\sigma(X,Y) \Rightarrow \sigma(X : S_1, Y_1 \wedge F) \bowtie \\ \sigma(X : S_2, Y_2 \wedge F) \ldots \bowtie \sigma(X : S_n, Y_n \wedge F) \quad (5)$$

It is worth to point out that rewriting process may require a recursion in the transitivity property of semantic mapping.

The query planning is a sequence, each node of the sequence can be denoted as $P_n = (Q_n, S_n, C_n, F_n)$, where $Q_n$ is the query which is needed to rewrite, $S_n$ is a set of sub query executed on different sites, $C_n$ denotes the connection operator, in most time, it is $\cup$ or $\bowtie$ operator, $F_n$ is the predication which denotes the connection conditions. $P_n$ represents the query rewriting procedure of query $Q_n$. The query planning procedure of user's query $\sigma(X,Y)$ can be expressed in algorithm 1.

## 6. Evaluation

We develop a mediator based information integration system named OBSA, the logical architecture of OBSA is illuminated in figure 3. It is divided into five layers:

1. Information Source Layer (ISL). This layer is the autonomic systems, such as Enterprise Information System, File System Servers, Relation or Object-oriented Database System, Workflow System and so on. This paper focuses on the topic of converting relation database to Ontology.

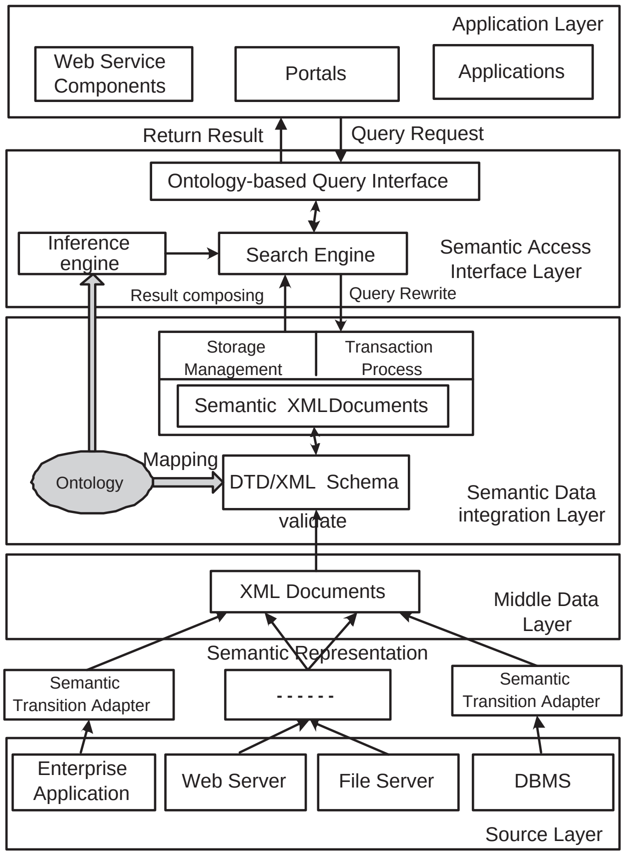


Figure 3. The general architecture of OBSA.

2. Middle Data Layer (MDL). The semantic transition adapter generates the middle data layer under a specified mechanism.

3. Semantic Integration Layer (SIL). This layer manages the storage of the data from middle data layer, and supplies the basic semantic service for semantic access interface layer. It is also the core semantic platform for the application layer.

4. Semantic Access Layer (SAL). Provided the access method and interface for the application systems. This paper focuses on relational query, an XML query has been discussed in paper [16].

5. Application Layer (APL). Applications that use the access layer interface to process the data from the distributed data source.

Unstructured and semi-structured information have different types and operation methods, they are incompatible with each other. It is very important to provide a global and consistent semantic view to all the information to overcome the diversity of them. OBSA uses a top-down semantic method to collect unstructured and semi-structured information, and stores them at

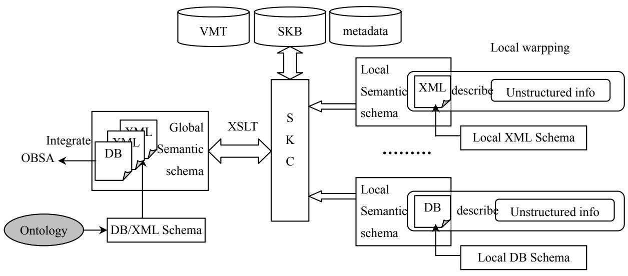


Figure 4. OBSA Adapter Architecture

the Semantic Integration Layer. The procedure can be described as follows:

OBSA employs web service based components named **Semantic Adapter** to perform the task of ontology integration. The semantic adapter acts as the wrapper of local site information, different local site has different semantic adapter, the global ontology site acts as the mediator of OBSA, the structure of the semantic adapter based information integration system can be illustrated by figure 4. The function of the semantic adapter can be described as follows:

– **Ontology Establishing**. With the help of domain expert, semantic adapter creates the local ontology to supply a local semantic view to express the semantic of local information source;
– **Semantic Mapping**. The semantic adapter maintains a mapping table, and mapping the local semantics to global semantics using method introduced in this paper;
– **Query Processing**. The semantic adapter accepted the query request from the global site, transfer it to the form which the local information source can accepted, the local information source execute the query and semantic adapter transform the result to the form which the global information site needed with XSLT technology;
– Some other functions will be added in the future.

The main components of semantic adapter can be described as following:

(i) **SKC** (semantic knowledge construction). SKC constructs semantic mapping knowledge between schemata, it uses the results of schema extraction and concept matching to establish mapping between local and global semantics schemata. The mapping knowledge is saved in VMT;

(ii) **MDD** (Meta data dictionary). It could include some description of information source, such as schema, storage path, type and provider etc.;

(iii) **SKB** (semantic knowledge base). It includes the knowledge needed to understand Ontology concept and their attributes, they are synonymous words, comparison of Chinese and English etc. these knowledge is crucial to concept matching. SKB can expand automatically in the process of matching;

(iv) **VMT** (vocabulary mapping table).The VMT contains the mapping list of local ontology and its instances, one item of the list can be described as figure 5:

```
<TItem>
  <STerm>Ontology.Term</STerm>
  <Description>
      Description of the Concept or Ontology
  </Description>
  <MappingList>
    <MapItem Type="M"> %Direct Mapping
      %IP Address, Port or Semantic
      %Adapter Service Descrption
      <Source>Source Description</Source>
      <MTerm>Source1.Term</MTerm>
      <Relation>Map1.Relation</Relation>
      <CValue>Confidence Value></CValue>
    </MapItem>
    <MapItem Type="S"> %Subsumption Mapping
      <Source>Source Description></Source>
      <MTerm>Source2.Term1</MTerm>
      ....
      <MTerm>Source2.Termn</MTerm>
      <Relation>Map2.Relation</Relation>
      <CValue>Confidence Value></CValue>
    </MapItem>
    <MapItem Type="C"> %Composition Mapping
      <Source>Source Description></Source>
      <MTerm>Source3.Term1</MTerm>
      <MConcatenate>
         Term1.Concatenate
      </MConcatenate>
      ....
      <MTerm>Source3.Termn</MTerm>
      <MConcatenate>Termn.Concatenate</MConcatenate>
      <Relation>Map3.Relation</Relation>
      <CValue>Confidence Value></CValue>
    </MapItem>
  </MappingList>
</TItem>
```

Figure 5. Vocabulary Mapping Table

(v) **MQW** (Mapping and Querying Wrapper), act as the wrapper for ontology mapping and information retrieval.

Semantic accessing interface layer of OBSA accepts the request from the user of application layer, validates and optimizes it, and makes plan with rewriting technology for the requests and sends the executing plan to appropriate local sites, and gets results from these sites and returns the results to the users.

## 7. Discussion and Conclusion

The paper mainly discusses the extension of query planning on distributed database enabled information systems with wrapped ontologies. It discusses the converting from relation database to ontology and the complex ontology mapping patterns and fusion, it also discusses the semantic planning mechanism, which primarily extends structural query algebra wrapped with local ontologies. However, query optimizing in distributed web sites was not considered in the query planning mechanism discussed in this paper, future research will be focused on this topic.

## Acknowledgment

This work was partially supported by a grant from the NSF (Natural Science Fundation) of China under grant number 60803160 and NSF of Hubei Prov. under grant number 2007ABA296, NSF of educational agency of Hubei Prov. under grant number Q200711004, B20071106 and T200801. It was partially supported by China Postdoctoral Science Foundation under grant number 20060400275 and Jiangsu Postdoctoral Science Foundation under grant number 0601009B.

## References

[1] A. R. Alon Y. Levy and J. J. Ordille, "Query heterogeneous information sources using source descriptions," in *Proceedings of the 22nd VLDB Conference*. Mumbai, India: Morgan Kaufmann Publishers Inc, 1996, pp. 251–262.

[2] B. He, M. Patel, Z. Zhang, and K. C.-C. Chang, "Accessing the deep web," *Communications of the ACM*, vol. 50, no. 5, pp. 94–101, 2007.

[3] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schüster, H. Neumann, and S. Hubner, "Ontology-based integration of information - a survey of existing approaches," in *Proceedings of IJCAI-01 Workshop: Ontologies and Information Sharing*. Seattle, WA: Springer, 2001, pp. 108–117.

[4] J. Gu, B. Xu, and X. Chen, "An XML query rewriting mechanism with multiple ontologies integration based on complex semantic mapping," *Information Fusion*, 2008.

[5] Y. Kalfoglou and M. Schorlemmer, "Ontology Mapping: The State of the Art," *The Knowledge Engineering Review*, vol. 18, no. 1, pp. 1–31, 2003.

[6] W. Boggs and M. Boggs, *Mastering UML with Rational Rose*. Sybex, 1999.

[7] D. L. McGuinness and F. van Harmelen. (2004) Owl web ontology language overview. World Wide Web Consortium(W3C). [Online]. Available: http://www.w3.org/TR/owl-features/

[8] Y. An, A. Borgida, and J. Mylopoulos, "Inferring complex semantic mappings between relational tables and ontologies from simple correspondences," in *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, ser. Lecture Notes in Computer Science, no. 3761. Springer Verlag, 2005, pp. 1152–1169.

[9] J. Barrasa and A. G.-P. Òscar Corcho, "$R_2O$, an extensible and semantically based database-to-ontology mapping language," in *Second Workshop on Semantic Web and Databases (SWDB2004).*, 2004, pp. 92–119.

[10] C. P. de Laborda and S. Conrad, "Relational.OWL - A Data and. Schema Representation Format Based on OWL," in *Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005)*, ser. CRPIT, no. 43. ACS, 2005, pp. 89–96.

[11] Y. Kalfoglou and M. Schorlemmer, "IF-Map: An Ontology-Mapping Method Based on Information-Flow Theory," *Journal on Data Semantics, LNCS*, vol. 2800, pp. 98–127, 2003.

[12] G. Stumme and A. Maedche, "FCA-MERGE: Bottom-up merging of ontologies," in *Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, WA, 2001, pp. 225–234.

[13] J. KWON, D. JEONG, L.-S. LEE, and D.-K. BAIK, "Intelligent semantic concept mapping for semantic query rewriting/optimization in ontology-based information integration system," *International Journal of Software Engineering and Knowledge Engineering*, vol. 14, no. 05, pp. 519–542, 2004.

[14] P. Bonatti, Y. Deng, and V.S.Subrahmanian, "An ontology-extended relational algebra," in *IEEE International Conference on Information Reuse and Integration (IRI 2003)*. IEEE, 2003, pp. 192 – 199.

[15] E. Hung, Y. Deng, and V.S.Subrahmanian, "TOSS: An Extension of TAX with Ontologies and Similarity Queries," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, G.Weikum, Ed. Paris, France: ACM Press, June 2004, pp. 719–730.

[16] H. Chen, J. Gu, X. Li, and H. Fang, "An XML Query Mechanism with Ontology Integration," in *Proceedings of the WISA05 in conjunction with ISPA05*. Springer Verlag, 2005.