CHAPTER 9
LITERATURE REVIEW AND COMPARISON

The development of complete information integration systems is challenging. Recent research endeavors furnish numerous approaches and methodologies with special focus on resolving semantic heterogeneity. This chapter investigates works of the existing systems on information integration with special focus on solving semantic heterogeneity. The selected works to be reviewed are classified into two main systems, namely, the mediator-based system and the description logic-based system. These systems provide declarative querying over heterogeneous sources, but differ in supporting techniques and quality of integration obtained. A range of systems are reviewed according to their frameworks that encompass the scope, architecture, and functionality. The advantages and disadvantages of these systems are explored to compare their characteristics with the proposed SIGA from various aspects. The selected systems to be reviewed are the TSIMMIS project based on the mediator-based systems, the Information Manifold, and the OBSERVER based on description logic-based system. However, these systems differ in architectures and techniques in fulfilling their objectives. A survey and comparison of these systems in more details can be found in (Busse, Kutsche, Leser, and Weber, 1999; Jakobovits, 1997; Paton, Goble and Bechhofer, 2000; Wache, Vögele, Visser, Stuckenschmidt, Schuster, Neumann, and Hübner, 2001).

9.1 The Mediator-based Systems (Busse, Kutsche, Leser and Weber, 1999)

The term mediator was introduced by (Wiederhold, 1992) and used in many data integration researches. In general, a mediator is a flexible and re-usable software component that mediates between the user and physical information sources. Some mediators may be designed to use other mediators as components.

A mediator-based information system is illustrated in Figure 9.1 which consists of four layers, namely, presentation layer, mediation layer, wrapper layer, and foundation layer. The presentation layer provides the global applications for users in accessing data at the lower layer. The mediator layer contains a number of mediators having their own

federated schema. The schema can be shared with other mediators schemas. User queries are systematically coordinated through the federated schemas. The wrapper layer abstracts out technical and data model heterogeneity transparency. Data and queries are converted to canonical format via the wrapper components. An example of such systems designed based on this architecture is the TSIMMIS. The foundation layer consists of various types of data sources to be accessed by the upper layer.

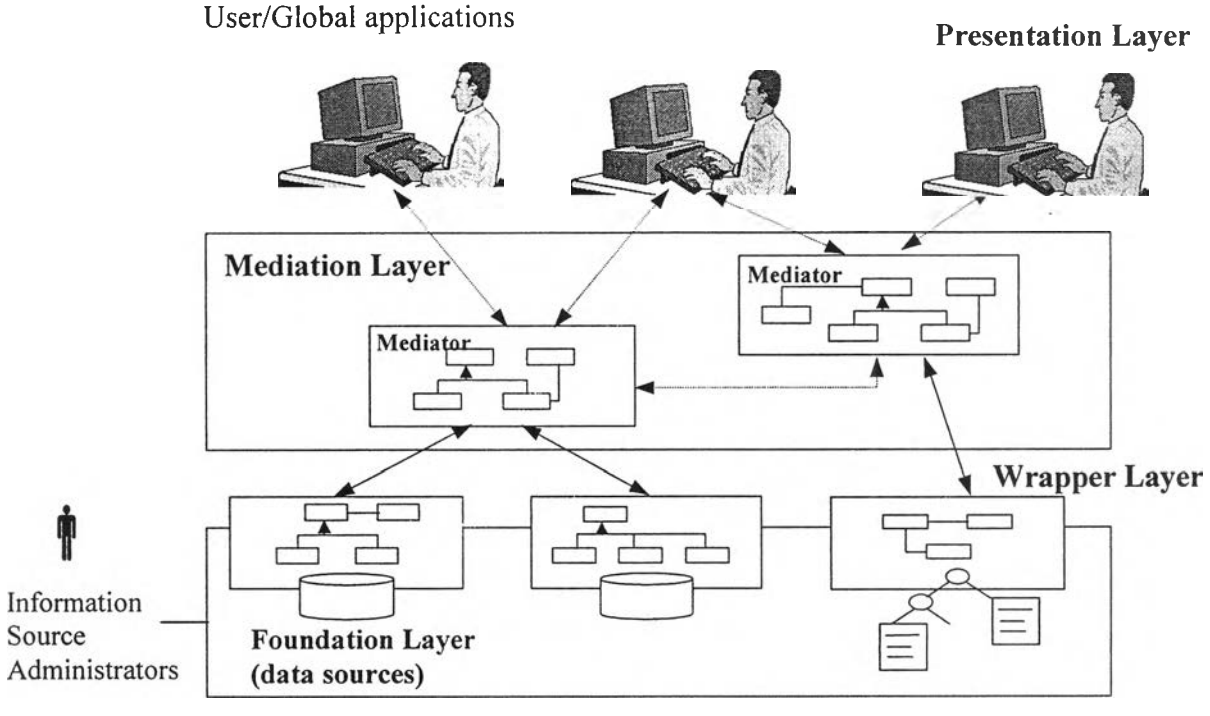

Figure 9.1 The mediator-based information systems architecture (Busse, Kutsche, Leser and Weber, 1999).

9.1.1 TSIMMIS (Garcia-molina, Papakonstantinou, Quass, Rajaraman, Sagiv, Ullman, Vassalos and Wisom, 1997)

The TSIMMIS ("The Stanford-IBM Manager of Multiple Information Sources") is a project originated by Stanford University that aims to support the integration of heterogeneous data sources. The goal is not to perform fully automated information integration, but to develop a framework and a collection of tools that assist users in their integration activities and facilitate rapid integration. The architecture consists of a collection of simple mediators, wrappers/translators, and mediators and wrappers generators as illustrated in Figure 9.2. A mediator possesses embedded knowledge that is necessary for processing a specific type of information and forwarding the query to the target sources, as well as processing and

merging the answers before forwarding them to the users. TSIMMIS does not provide an integrated schema, but propagates all schemas of the wrapper component to the users. To express the query and communicate between mediators and wrapper, including resolving data model heterogeneity, TSIMMIS adopted a simple self-describing (or tagged) object model called the Object Exchange Model (OEM) (Papakonstantinou, Garcia-Molina and Widom, 1995) as illustrated in Figure 9.3.

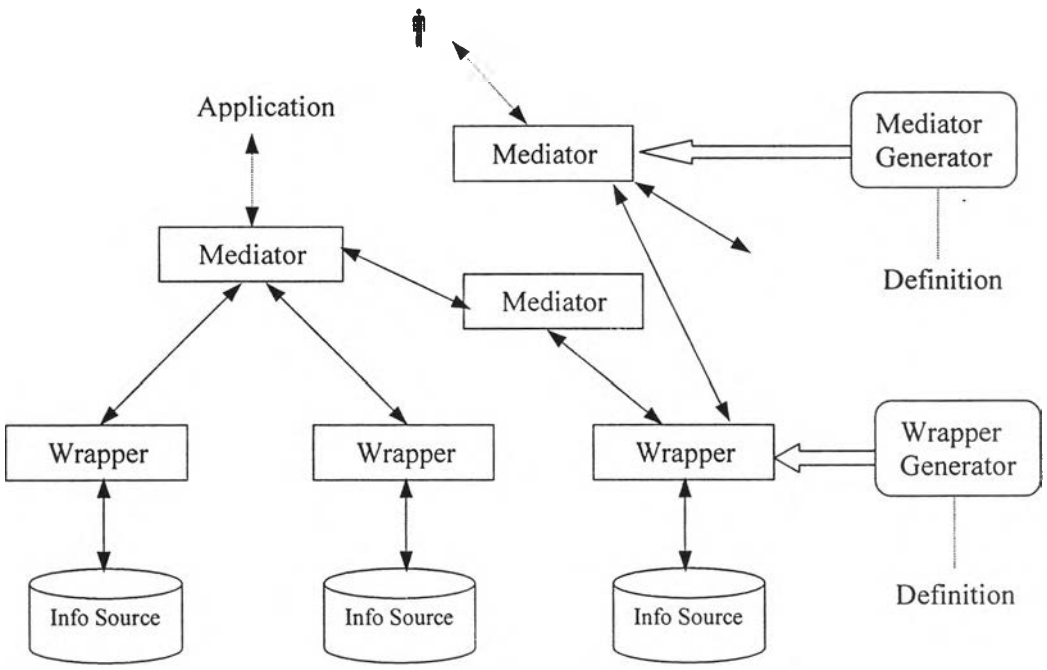

Figure 9.2 The TSIMMIS architecture (Garcia-molina, Papakonstantinou, Quass, Rajaraman, Sagiv, Ullman, Vassalos and Wisom, 1997).

To request OEM objects from an information source, a client issues queries in a language called OEM-QL adapting from SQL-like languages for object-oriented models. A wrapper accepts OEM-QL queries, decides whether its sources can directly support the query, and if it can do so, converts the expression into a local executable form. The results are exported as OEM objects.

Advantages:

(1) TSIMMIS aims to provide tools that facilitate the rapid integration of heterogeneous information sources to ensure that the information so obtained is consistent.

(2) TSIMMIS adopted a light weight model called OEM as a common model for solving the data model heterogeneity. This common model can be used to provide a simple and "client-friendly" front-end.

Disadvantages:

(1) TSIMMIS places high value on rapid and flexible "dynamic" integration by ignoring semantic or structural heterogeneity.

(2) TSIMMIS is a loosely coupled information system that does not offer global schema, hence, no transparency of location and schema to their users. Consequently, it is not possible to guarantee that every user or application sees consistent data every time it interacts with the system.

(3) The integration process requires extensive user participation. Integration may be automated by a mediator, but only after the user studies sample data and determines the procedures to follow. In the extreme case, the integration is performed manually by the end user.

(4) The wrappers are thick in that many tasks that are usually supposed to be the mediator's responsibility are assigned to the wrapper. Typical examples are query decomposition.

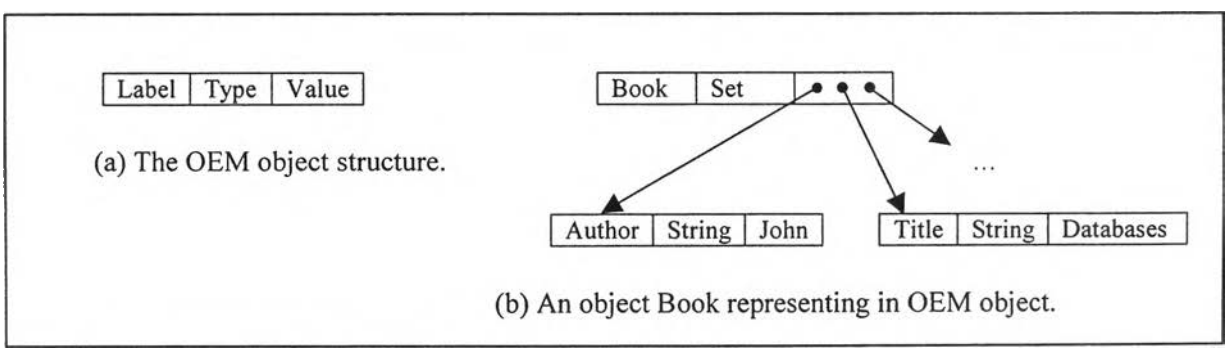

Figure 9.3 Examples of the OEM objects.

9.2 The Description Logic-based Systems

The description logic-based systems offer a different approach to elaborate source description by means of description logic (Borgida, 1995) for solving queries over multiple sources. Unlike the mediator approach which provides multiple views that can be layered

and tailored to specific user groups, the description logic approach abstracts the heterogeneous sources through a global view. Examples of such systems are the IM and OBSERVER.

9.2.1 Information Manifold (Levy, Rajaraman and Ordille, 1996)

The IM ("Information Manifold") is a project primarily based at AT&T and Bell Laboratories. IM provides a mechanism to describe declaratively the contents of information sources and query capability. The main objectives are query processing on heterogeneous information sources via source descriptions. The IM uses a relational data model, augmented with class hierarchies for describing and reasoning about the contents of information sources. The user can pose queries in terms of a single global view called the *World View*, which is a collection of virtual relations and classes that describes the contents of the information sources. The world view seems to be based on a knowledge representation language that extends the description logic CLASSIC (Borgida, Brachman, Mcguinness and Resnick, 1989). Since the principal language for expressing query is a conjunctive logic language over the set of relations in the world view (i.e., select-project-join queries), they are able to prune the source relevant to a given query. Examples of world view and source descriptions are illustrated in Table 9.1 and Figure 9.4, respectively.

Table 9.1: Examples of a class hierarchy representing the world view.

Class	Subclass of	Attributes	Disjoint from
Product		Model	Person
Automobile	Product	Model, Year, Category	Stereo
Car	Automobile	Model, Year, Category	Motorcycle
NewCar	Car	Model, Year, Category	UsedCar
UsedCar	Car	Model, Year, Category	NewCar
CarForSale	Car	Model, Year, Category, Price, SellerContact	

Source 1: Used cars for sale.
Contents: $V_1(c) \subseteq$ CarForSale(c), UsedCar(c)

Source 2: Luxury cars for sale. All cars in this database are priced above $20,000.
Contents: $V_2(c, p) \subseteq$ CarForSale(c), Price(c, p), $p \geq 20000$

Figure 9.4 Examples of source description related to the world view in Table 9.1.

An example of a user's query defined in terms of the world view is illustrated below:

$Q(c) \leftarrow CarForSale(c), Price(c, p), p > 40000$

The query processor can rewrite the user's query into a query defined in terms of the sources as illustrated below:

$Q(c) \leftarrow V_2(c, p), p > 40000$

The project also presents several algorithms that use source descriptions to generate executable query plans for subsequent inquiry on dispersed information sources, whereby obtaining the desired combined results. The IM architecture is shown in Figure 9.5.

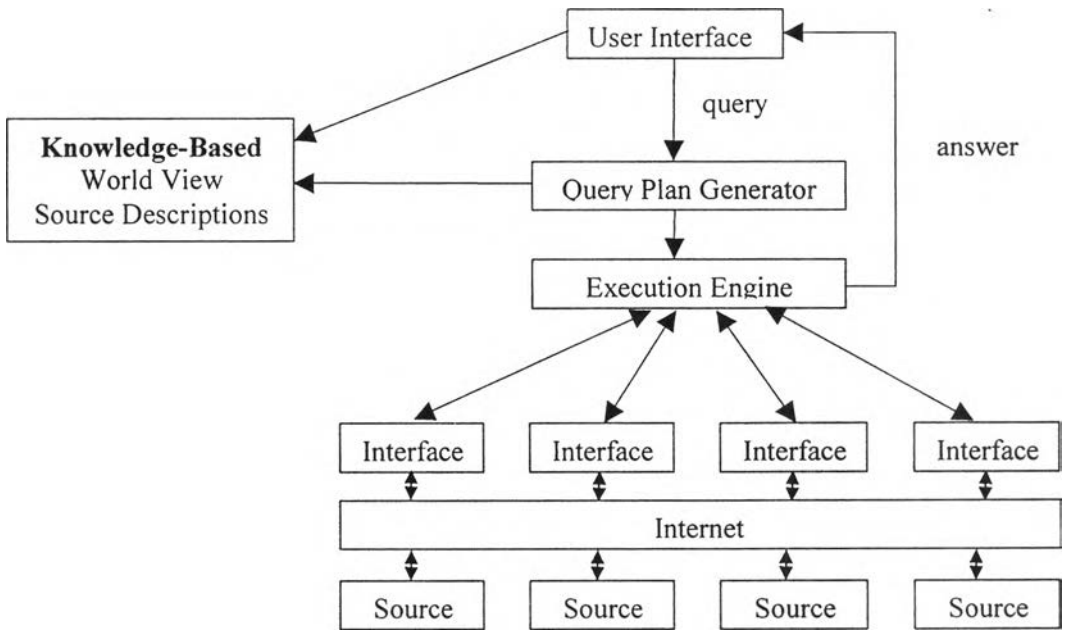

Figure 9.5. The information manifold architecture (Levy, Rajaraman and Ordille, 1996).

Advantages:

(1) The attempt to provide access to collections of information sources by focusing on describing declaratively the contents of an information source enables IM to express fine-grained distinctions between the contents of different information sources, thereby enabling to prune the sources that are irrelevant to a given query.

(2) Using the source descriptions to generate plans to answer the query is not restricted by which queries can be answered by the system, thus enables the system to add or

delete sources because they do not have to modify the query-specific procedures to accommodate the changes.

Disadvantages:

(1) Since IM is designed for use principally in conjunction with sources that provide declarative query facilities, the system may not be straightforward to extend any provision for more powerful querying facilities.

(2) Although user queries are formulated in terms of the world-view relations by freeing the users from having to interact with source schema individually, the conjunctive logic language of the queries is rather complex and not user-friendly in practice.

9.2.2 OBSERVER (Mena, Kashyap, Sheth, and Illarramendi, 1996)

The architecture of OBSERVER is designed to handle query processing over existing information sources in global information systems. The approach is based-on multiple ontology constructs where each information source associates with an ontology called component ontology that describes its contents. Interoperation across ontologies is achieved via terminological relationships by traversing semantic relationships defined between terms across ontologies. Their prototype system uses pre-existing real-world ontologies to describe real-world repositories from the same domain and to provide different conceptual views of the same data. The OBSERVER uses *intensional metadata* represented in Description Logic (DL) CLASSIC (Borgida, Brachman, Mcguinness and Resnick, 1989) to capture the information content of the repositories. For the first step of query processing, the user must choose and connect to one component ontology (refered to as the user ontology), where a user's query expressed in DLs is posing on. The basic elements of the architecture of OBSERVER are illustrated in Figure 9.6 and described in brief as follows:

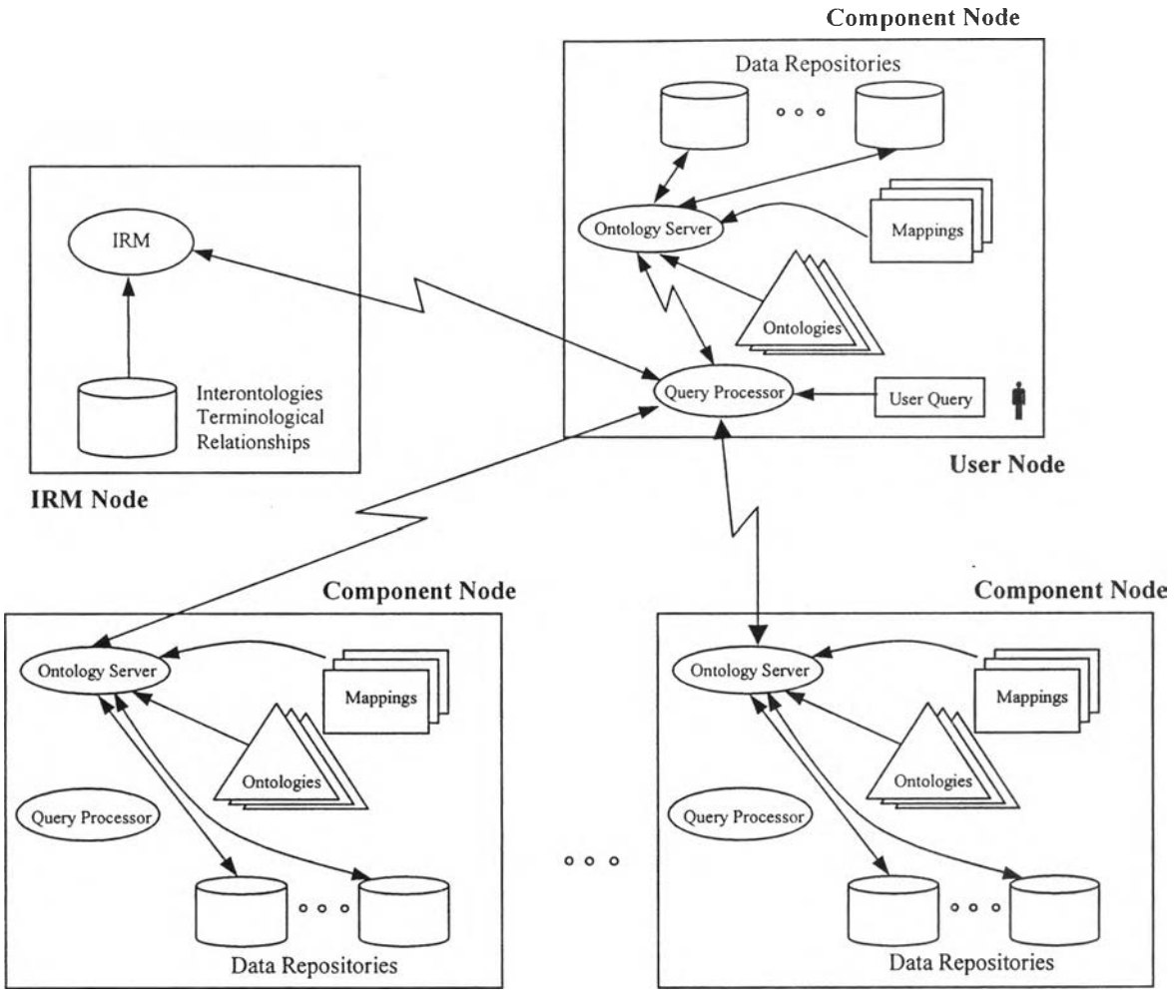

Figure 9.6. The OBSERVER: An architecture to support query processing (Mena, Kashyap, Sheth and Illarramendi, 1996).

- **Query Processor:** The Query Processor takes a user query expressed in DLs and translates query terms into the component ontology terms using synonym relationships and term definitions. The query processor can traverses other component ontologies, if the translation is unsatisfied for a given term. The data retrieved from the relevant ontologies are appropriately combined to yield the final answer.
- **Ontology Server**: The Ontology Server addresses the structure/format heterogeneity problem by submitting the definitions of terms in the ontology to the Query Processor and maps each term in the ontology with the data in the repositories.

- **Inter-ontology Relationships Manager (IRM)**: The IRM addresses the vocabulary problem by representing synonym relationships that relate the terms in various ontologies declaratively of an independent repository.
- **Ontologies**. Each ontology is defined as a set of terms of interest in a particular information domain expressed in DLs. This provides a solution to the querying information problem.

Advantages:

(1) The terminological relationships across the ontologies reduce the problem of learning the structure and semantics of data deposited over a number of repositories.

(2) Their approach provides an algorithm for partially translating the intensional query expression into proper ontologies, and a means to combine the partial translation from different ontologies.

Disadvantages:

(1) The rewriting of a CLASSIC query over one ontology to another ontology may lead to some loss of information in the query.

(2) Traversal over component ontologies in case the user is not satisfied with the answer may lead to considerable delay in the correlating process of information access and retrieval.

(3) The use of DLs to represent intensional queries and contents of the ontologies is not user-friendly and difficult to manipulate the ontologies.

9.3 Comparative Characterization with SIGA

The characteristics of the other systems and SIGA are summarized in various aspects concerning with query processing and integration of the HIS in Table 9.2.

Table 9.2: Comparison of various ontology systems and SIGA characteristics.

	TSIMMIS	IM	OBSERVER	SIGA
Architecture	Mediator-based	Description Logic-based	Description Logic-based	Layered structure of Mediator and Agent-based
Autonomy	High	High	High	High
Heterogeneity	Only data model	Structural, Semantic	Semantic	Semantic
Wrapper	Thick wrapper	Thin wrapper	Thin wrapper	Thin wrapper
Transparency	No Language, Location, and Schema transparency	Complete	Complete	Complete
Bottom-up vs. Top-down	Top-down	Top-down	Bottom-up	Bottom-up
Virtual vs. materialized	Virtual	Virtual	Virtual	Virtual
Read-only	Yes	Yes	Yes	Yes
Tightly coupled vs. loosely coupled	Loosely coupled	Tightly coupled	Loosely coupled	Tightly coupled
Data model of the FIS	OEM	Relational and class extension	Object-oriented	Object-oriented and XML data model
Mapping translation	Wrapper	Knowledge-based or World view	Ontology Server	Metadata Dictionary
Flexibility	Yes	Yes	Yes	Yes
Scalability	Yes	Yes	No	Yes
Interoperability	No	No	Yes	Yes
Robustness	No	No	No	Yes

SIGA also differs from other approaches in various standpoints, namely,

(1) SIGA provides a metadata dictionary as a knowledge repository that is flexible for agents to acquire knowledge dynamically. In other words, the agents are able to obtain knowledge from the metadata dictionary instead of operating on predefined static sources. This provision offers update flexibility of the knowledge in the metadata dictionary without affecting the normal operation of the agent;

(2) A user's query posed over virtual schema is mapped directly to physical schema without loss of information in the query, that is, the user's query needs not be rewritten in such a way to accommodate one ontology with another stored in dispersed sources, thus eliminating potential loss of information in query transformation process;

(3) The proposed approach defines domain ontology components based on object-oriented and set theory which aims to be a standard model. In so doing, the domain ontology can be applied to real world metadata dictionary implementation by means of independent tools and languages; and

(4) The domain ontology is expressed in XML-based architecture that is easy for agents to gain information from the metadata dictionary. This representation provides a means for consolidating data retrieved from various sources, while retaining consistent identification of the data semantic. Such a configuration is suitable for representing data from the HIS in a web-based environment.

The advantages of SIGA over other approaches are:

(1) Since SIGA is designed based on mobile agent architecture to connect the client and server machines, the overhead of reconnecting the server and resource consuming are eliminated. Meanwhile, the client is also relief from loading the data source connectivity.

(2) SIGA covers almost all useful features that other approaches provide.

(3) Since the data models of FIS of SIGA are designed based on object-oriented model when representing the domain ontology components and XML data model for

metadata dictionary components, SIGA support for flexibility, scalability, and interoperability is extensive. Besides, the mobile agents that are incorporated to the reference architecture provide high degree of robustness.

However, SIGA still has some inherent drawbacks:

(1) The preliminary design of SIGA supports only structure and semi-structure sources. It has to be extended to support unstructured data sources.

(2) The query processing design still lacks optimization capability to enhance query processing efficiency.