CHAPTER 3
SIGA: SEMANTIC INFORMATION GATHERING APPROACH

The inherent variations in data definitions imposed by each host pose a great challenge for integrating efforts on distributed processing. This chapter introduces the SIGA architecture that serves as a unified framework for accessing and integrating the HIS in a web-based environment. Such extensive supports render the target environment a transparent homogeneous web-based repository. First a design overview of a reference architecture of SIGA system is introduced based on layered-architecture that supports portability. Next the functional details of the components in each layer will be described, as well as the agent architecture. With the benefit of agent mechanism, mobile agents can operate autonomously and asynchronously within and across layers, thus enabling the system to sustain the effects of intermittent network operation or failure in information sources. The proposed reference architecture aims to enhance the client/sever model by eliminating the overhead of reconnection by means of mobile agents to interconnect between the clients and their corresponding servers. This relieves both clients and servers from their respective communication processing. Any communication between the client and the server goes through this mobile agent. A system developed and deployed based on the reference architecture will provide flexibility, robustness, scalability, interoperability, and portability for information gathering from the HIS.

3.1 An Overview of the SIGA Reference Architecture

A reference architecture for integrating HIS on the WWW (Arch-Int and Sophatsathit, 2002) of SIGA (Arch-int and Sophatsathit, 2003) is designed based on layered-architecture as depicted in Figure 3.1. The architecture of SIGA consists of four main layers, namely, presentation, mediator, search, and resource layers. The main objective of employing layered-architecture is to support component *portability* which enables a system to be run on a variety of platforms. Each layer is described briefly below.

The presentation layer is the highest layer of the reference architecture that encompasses various web client components. This layer is designed specifically to enable the users posing their requests over a unified-query form which encircles the virtual schema of the metadata dictionary. The web client components communicate with the user interface agent of the mediator layer by forwarding the users' requests and receiving the returned results from the user interface agent in the XML format. In so doing, cross systems interoperability can be achieved with the XML flexible data model.

The mediator layer is the core layer of the reference architecture. The primary responsibility of this layer is to bridge the gaps between format-specific components at the search layers and the presentation layer. This layer consists of three principal components, namely, the user interface agent, the managing agent, and the metadata dictionary.

(1) *The user interface agent* is a stationary agent responsible for generating and forwarding a global transaction, which includes the visual user requirements, to the managing agent;

(2) *The managing agent* is a stationary agent responsible for receiving the global transaction and decomposing it into sub-transactions corresponding to the underlying physical information sources, as well as initiating the search agents for transmitting these sub-transactions to the physical destination sources. In the reverse direction, the managing agent integrates the multiple results delivered by the search agents into unified XML-based data. The inconsistency and redundancy in the data are eliminated from the combined result through the decomposition and integration processes given in Chapter 8.

(3) *The metadata dictionary* is a repository for storing all data descriptions of the application domain at the conceptual level and of the associated physical sources at the physical level. These data descriptions are used by the user interface agent and the managing agent in accessing and integrating the real data of the underlying physical sources. The metadata dictionary management is designed to support *flexibility* in manipulating the metadata structure and *scalability* of adding or dropping new physical information sources without affecting the overall system

configurations. Details of the metadata dictionary design and management can be found in Chapter 4.

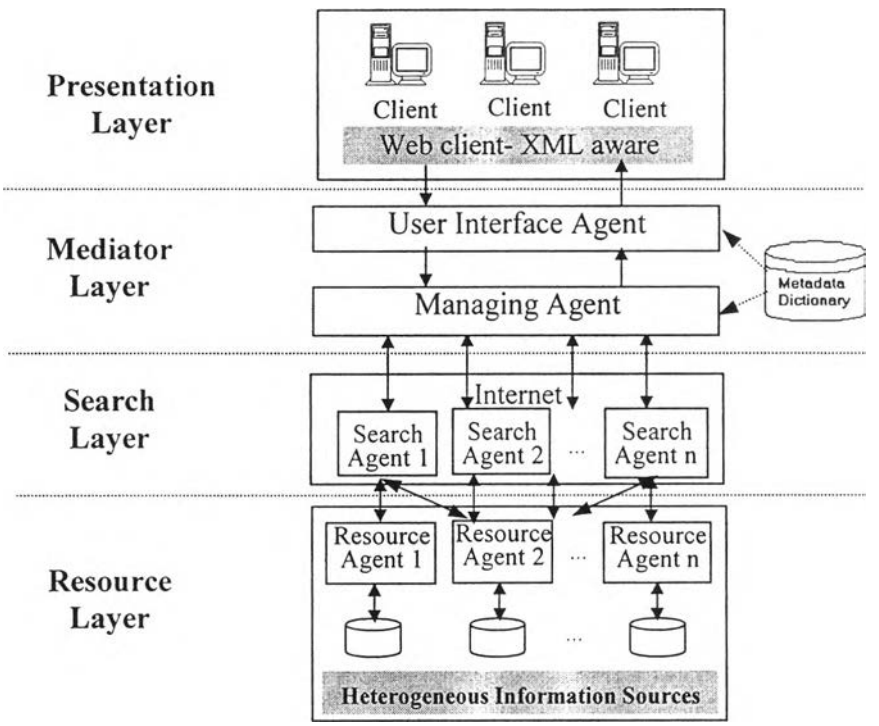

Figure 3.1 The reference architecture of SIGA.

The search layer consists of search agents which are mobile agents that enable tasks to be embedded into each search agent, which can then be dispatched into the network. Each search agent is responsible for packing and carrying a sub-transaction and necessary physical source configurations to a destination source at the resource layer. After being dispatched, the search agent become independent of the creating process and can operate asynchronously and autonomously at the resource layer. In case of the communication network failure, the search agent can continue their operations and return the desired results back to the mediator layer when the connection is reestablished. Hence, the mobile agent incorporated with the reference architecture provides greater *robust* and *fault-tolerant* distributed systems.

The resource layer encompasses HIS and resource agents which are stationary agents and are often referred to as agent hosts. Each resource agent connects directly with a physical information source via the middleware mechanisms, such as ODBC/JDBC, HTTPD, C++

Interface, etc. The main responsibility of each resource agent is to provide an interface wrapper to convert the incoming sub-transaction into an appropriate data manipulation language regulated by the proprietary information source. In the reverse direction, the resource agent transforms the results obtained from the execution of each sub-transaction into canonical XML-based format, encodes them, and passes them on to the search agent before shipping back to the managing agent.

3.2 Presentation Layer

The web client component provides location and heterogeneous transparencies to users for maximal flexibility. The users can freely maneuver their search requests without any a priori knowledge of local schemas, query languages, and data models of the information sources. The responsibilities of this layer can be summarized as follows:

(1) Support graphical user interface for the users to view the virtual schema consolidated by the user interface agent, and to create their customed views;

(2) Provide a unified-query form for the users to pose their requests against their views;

(3) Formulate and validate users' requests to insure that correct information is passed to the user interface agent; and

(4) Display the returned results from the user interface agent in XML format.

3.3 Mediator Layer

The three main components of the mediator layer, that is, the user interface agent, the managing agent, and the metadata dictionary can be described in details as follows:

3.3.1 User Interface Agent

The user interface agent is a stationary agent responsible for providing the virtual schema to the users, generating a global transaction associated with the user's requests, as well as validating the syntax by means of the metadata dictionary. This agent is made up of three modules (see Figure 3.2) as follows:

(1) *User information classification* which is responsible for classifying a local or a remote user's request obtained from the presentation layer.

(2) *Global Transaction Generator* which is responsible for converting a user's request into "a global transaction" by means of the metadata dictionary, as well as validation and simplification. A global transaction is a visual user requirement represented in standard SQL format using virtual attributes and relations that associate with one or more physical information sources.

(3) *User Package Generator* which is responsible for consolidating the following information in a user's package to be forwarded to the managing agent:

- User's information (e.g., name, address, etc.),
- Type of a user (e.g., local or remote), and
- A global transaction.

The user interface agent subsequently forwards the results obtained from the managing agent to upper layer web clients.

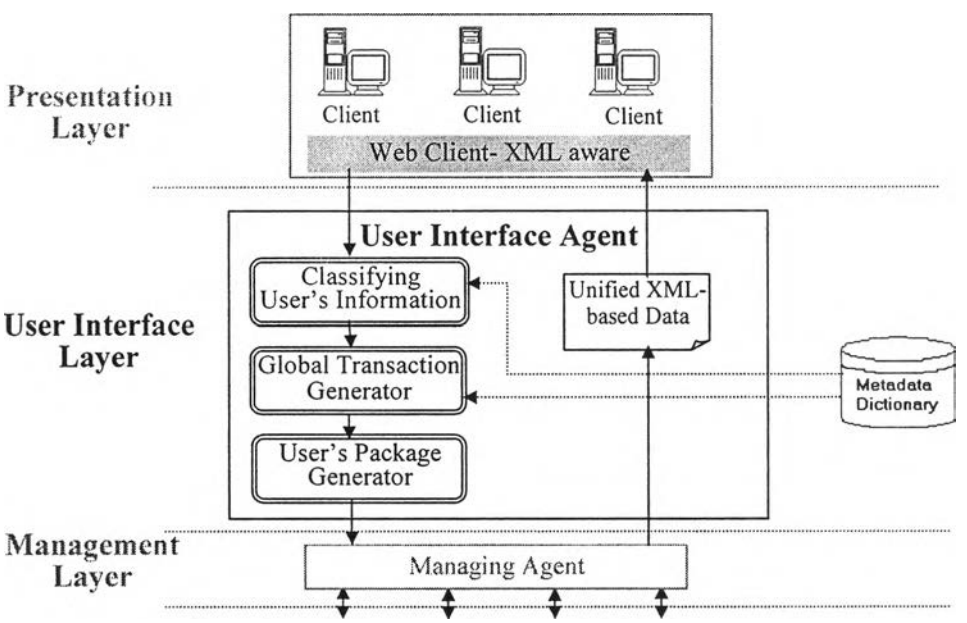

Figure 3.2 The user interface agent architecture.

3.3.2 Managing Agent

The managing agent is a stationary agent responsible for generating and transmitting sub-transactions through the search agents, as well as integrating the results returned by the search agents into the unified XML-based data, which will be referred below. The managing agent is made up of four main modules (see Figure 3.3) as follows:

(1) *Privilege Analysis* determines user's privilege to access the information sources based on relevant data descriptions defined in the metadata dictionary.

(2) *Sub-transactions Generator* decomposes a global transaction into several sub-transactions and assigns each sub-transaction to the corresponding information source. All references of virtual attributes and relations in the global transaction are replaced with actual attributes and relations of the destination sources for each sub-transaction with the help of the metadata dictionary.

(3) *Search Agent Creator* initializes a number of search agents according to the number of sub-transactions to operate at remote sources. In addition, the search agent creator encodes and packs all necessary information through the serialization process in each search agent before shipping the search agents to their destination sources. The information includes:

- User's privilege to information sources,
- Sub-transaction corresponding to the specific information sources,
- The address of the destination sources, and
- The authentication information of search agent to be trusted by agent host at the destination sources.

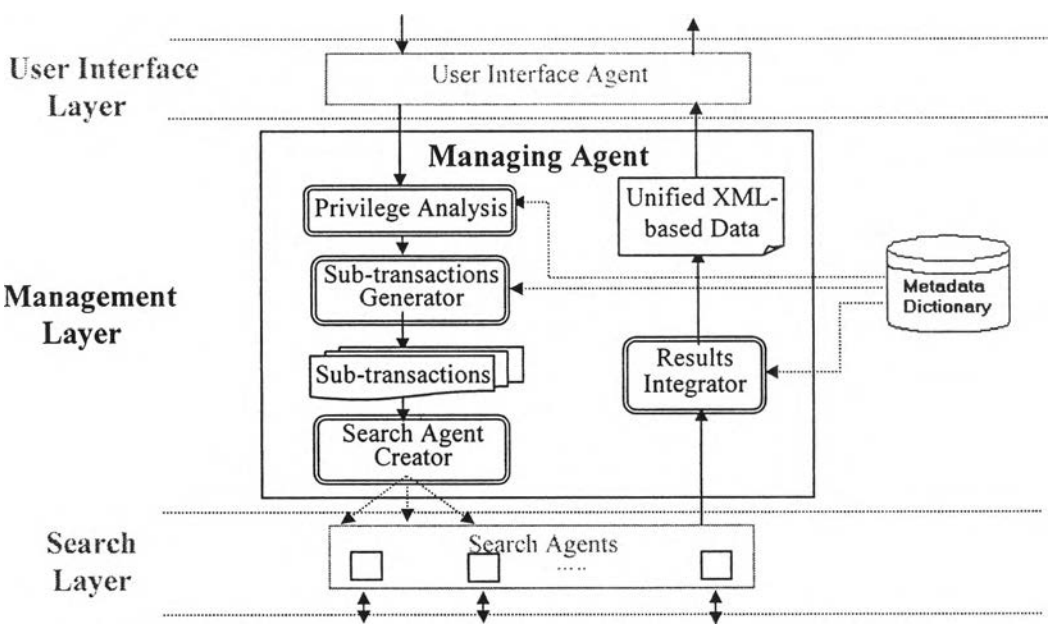

Figure 3.3 The managing agent architecture.

(4) *Results Integrator* is responsible for combining or integrating the multiple results delivered by the search agents to the "unified XML-based data." This reference architecture is designed on the notion of virtual integration architectures that temporarily materialize the query results at the time the query is posed (Busse, Kutsche, Leser and Weber, 1999). A query will be translated to sub-queries or sub-transactions which are propagated dynamically to the destination sources by means of the metadata dictionary to guarantee the obtained results to be fresh at query time. The global transaction decomposition and integration processes described in Chapter 8 provide a means to eliminate the data inconsistency and redundancy from the combined result. The internal process of the result integrator is depicted in Figure 3.4.

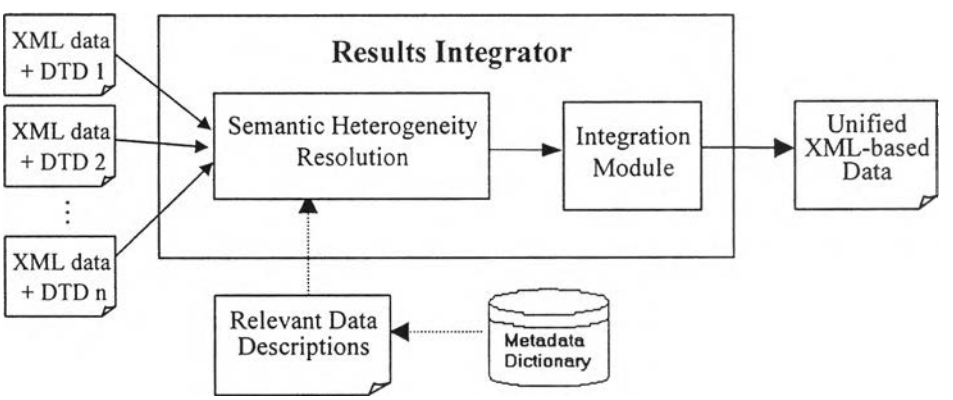

Figure 3.4 The internal process of result integrator module.

3.3.3 Metadata Dictionary

The metadata dictionary is designed based on the domain specific metadata using ontology as an assistant mechanism for accessing and integrating data from the HIS, and providing a means for transforming data to the user's views. The ontology serves as a unifying framework for reconciling the HIS to be a homogeneous logical view in which different data models and terminologies are uniformly represented. The main functionalities of metadata dictionary can be summarized as follows:

I21507715

- Providing virtual schemas of the application domain in which the users can pose their queries expressed, whereby abstracting the users from the underlying physical sources,
- Providing mapping mechanisms to map the virtual schemas onto the associated physical schemas, and
- Providing physical source configurations that are necessary for search agents in accessing the HIS. Examples of such physical source configurations are the physical source names, the network location of each physical source, the physical entity names residing in each physical source, data models of each source, query languages, owners, and permission of each physical entity, etc.

To maintain consistency and accuracy of the stored data descriptions and other information, the metadata dictionary will be updated when there are modifications of the data description, data model, constraints, fragmentation and replication of the information sources, or changing the user's access privilege.

3.4 Search Layer

The main responsibilities of these search agents in this layer are as follows:

(1) Carry sub-transaction and related user's information sent from the managing agent to the destination source.
(2) Communicate with the resource agents to acquire the required information.
(3) Communicate with other search agents to exchange intermediate data among themselves, thus avoiding unnecessary data transmission between the search agents and the managing agent.
(4) Pack and encode, through a serialization process, the results obtained from the resource agents before dispatching back to the originating managing agent.
(5) Collect and ship the results obtained from the resource agents back to the managing agent.

3.5 Resource Layer

Each resource agent carries out the following functions (see Figure 3.5):

(1) *Unpacking Request* receives and unpacks the request from the search agent and decodes it accordingly.

(2) *Verification* verifies the authentication credentials of the search agent and authorizes checking of user access privilege to designated local information sources.

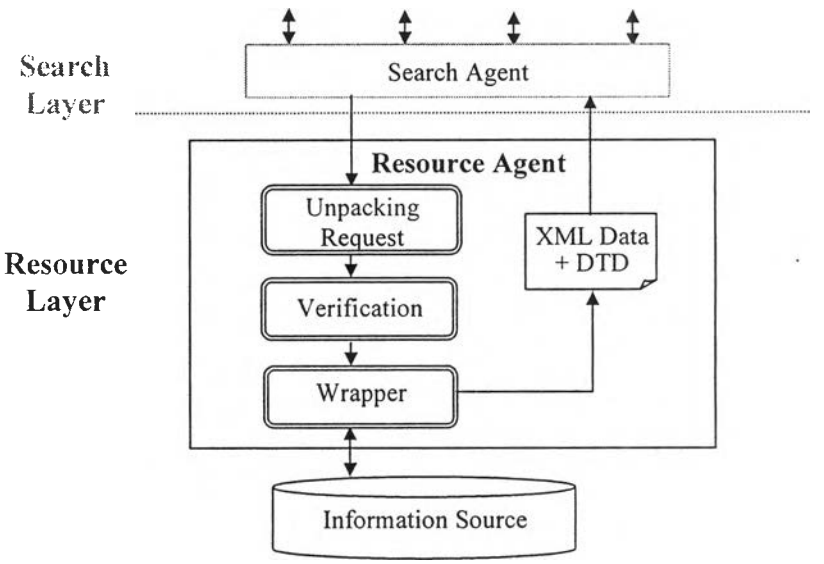

Figure 3.5 The resource agent architecture.

(3) *Interface Wrapping* encapsulates in/out parameters passing between the search agent and the heterogeneous data as follows:

- Establishes a connection with the information sources via middleware mechanisms, such as ODBC/JDBC, HTTPD, C++ Interface, etc. This architecture chooses JDBC as a data source connectivity.
- Converts the incoming sub-transaction to an appropriate data manipulation language regulated by the proprietary information source; and
- Transforms the results obtained from the execution of each sub-transaction into canonical XML format (XML data and XML-DTD) before decoding and passing on to the search agent.