InfoHub: A Flexible System for Retrieving and Integrating Heterogeneous Information Sources*

Yu-Chi Chu
Dept. of Electronic Engineering
National Taiwan Ins. of Tech.
ycchu@sun.epa.gov.tw

Chih-Cheng Lien
Dept. of Computer Science
Soochow University
cclien@bigsun.cs.scu.edu.tw

Chen-Chau Yang
Dept. of Electronic Engineering
National Taiwan Ins. of Tech.
ccyang@selab3.et.ntit.edu.tw

Abstract

A modular approach to build a system, InfoHub, for retrieving and integrating information from diverse sources is presented. This paper gives an overview of the system architecture and describes the design and implementation of InfoHub. By incorporating the kernel and the auxiliary modules, the system can access the relevant information of a specific domain in an active fashion. Our system provides each information source a dedicated wrapper, which make the overall architecture extensible and flexible. We have already built a small prototype that has efficient and active capabilities for accessing information in a domain of air quality monitoring.

1 Introduction

Recently, information retrieving and integrating in the multiple network environment has become a very attractive research area due to the advances in database systems and communication network technologies. We are currently witnessing an explosion in the amount of information sources including traditional databases, flat files, image databases that are available online. Such an environment can be viewed as a set of heterogeneous information sources. World-Wide Web (WWW) browser on the Internet environment allow users to search through large numbers of information sources, but very little attention has been paid for combining, organizing, and integrating the related information. More importantly, some applications (such as emergence management, real-time control), require that we access the information via networks not only by passive query, but that access the relevant information automatically by pre-defined knowledge. Although some of the commercial DBMSs recently also provide certain functions like a trigger that can monitor the situation of database and play as an isolated active database in a certain degree. However, they provide very limited capabilities for the active retrieval and integration of information in the environment of heterogeneous information sources. For example, researchers and practitioners in disciplines, such as scientists and engineers in the Environmental Protection Agency(EPA), need access to existing environmental monitoring data and satellite image data stored in different databases and different sites to analyze and monitor global environmental processes. Unfortunately, existing DBMS tools and languages lack the facilities to aid these users in understanding and accessing the information they need directly, transparently and furthermore, automatically.

This paper describe the InfoHub, an flexible architecture which can retrieve and integrate information from disparate sources. Furthermore, by the pre-defined domain knowledge in the knowledge server, the mediator can plan to *pre-fetch* the relevant information. This feature not only makes the system perform active services, but also the overall efficiency of the system can be improved.

In our approach, the architecture of InfoHub can be divided into two subsystems, the kernel and the auxiliary, and each can be implemented separately. Besides, a mediator for one domain can also serve as an information source to other mediator. Hence, modularity can be achieved in a distributed environment. Furthermore, for a given type of information source, only one corresponding wrapper will be applied. Adding a new information source simply requires building a wrapper above it, as well as

*This work was supported by the National Science Council, R.O.C. under Grant Number NSC85-2213-E011-025, NSC86-2213-E011-023 and by the Environmental Protection Administration R.O.C. under Grant Number EPA-85-E3L1-09-01.

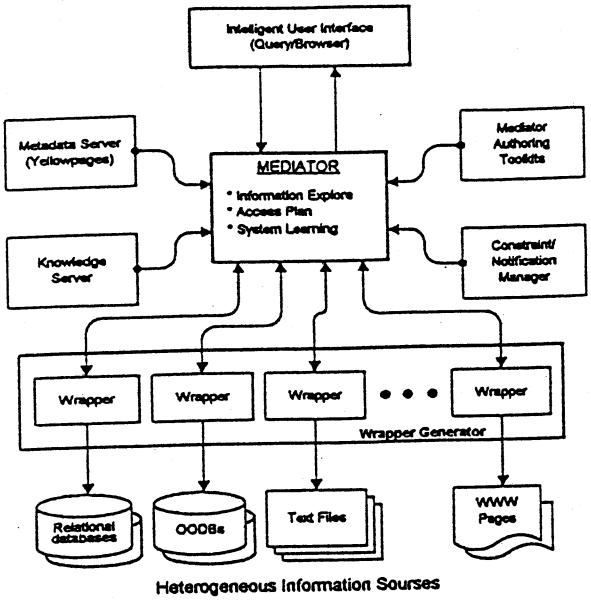

Figure 1: The architecture of the InfoHub

specify how it related to the mediator. This makes the overall architecture extensible and flexible.

2 Overview of the Architecture

Figure 1 depicts the architecture of the InfoHub. Generally speaking, the goal of our system is to provide the applications and users with transparent and effective accessing of the heterogeneous data from a multiple information sources. Moreover, the accessing behavior is not only in a passive characteristic, but in an active fashion.

The InfoHub can be roughly divided into two subsystems: the kernel module and the auxiliary module. The kernel module is composed of a set of wrappers, mediators, and an intelligent interface. The auxiliary module currently consists of a metadata server, a knowledge server, a constraint/notification manager and a set of the mediator authoring toolkits. Our idea is centered around "middleware" concept by using a number of software modules to bridge heterogeneity and autonomy among information sources. We adopt modular approach to develop the software, thus the whole system can be implemented flexibly and dynamically. In this section, we briefly introduce the various components of proposed system architecture by describing their functionality and interrelationships.

2.1 Information Sources

Information sources are a set of pre-existing databases, file systems, WWW servers and so on. These sources support a set of localized applications for the individual department use, and are autonomous and heterogeneous. Since the non-standard applications need new types of data, examples of reasonable pre-existing databases may include relational and non-relational databases, image databases geographical databases and so on. Such situation not only increases the heterogeneity of the information sources, but significantly adds to the interoperative problems in a multiple network environment.

2.2 Wrappers and Uniform Protocol

As shown in figure 1, each source has a corresponding wrapper that logically translates the underlying information to a uniform protocol. For wrapping the heterogeneity, the wrapper provides a uniform protocol (common interface) to mask the details of invoking, input and output format of each information source. It translates the information about the command text into a format understandable by the information sources, and send the text to the sources for execution; i.e., it manipulates request between mediators and information sources. Therefore, mediators needs no detailed information about information sources; that is, from the viewpoint of mediators, all wrappers appear the same. The wrappers enable mediators and information sources communication by providing the necessary format transformations.

For the InfoHub project, we adopted Loom knowledge representation language as an uniform protocol between wrappers and mediators. Loom is also known as a sort of *description logics* that can represent hierarchies of classes and relations, as well as classifying instances of classes and reasoning about descriptions of object classes. In addition, description logics has been indicated that can achieve enhanced access to data and knowledge systems[2].

Figure 2 depicts the process in a wrapper. The query processor first unpack the query from mediator, then translates Loom expression into the language originally handled by the target information source (for example, SQL in the case of relational database). The connection manager will connect underlying information source, send the query, and collect the result which place into buffer manager temporary. There are quite a few commercial products such as Open DataBase Connectivity (ODBC), and Java DataBase Connectivity (JDBC) can be used as the connection manager in our system. This will reduce a part of the ef-

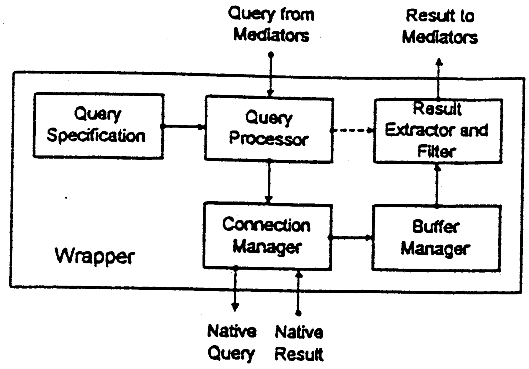

Figure 2: The components and process of a wrapper

fort of the implementation and make our system to commit the requirement of transparent connection. The result extractor and filter is in charge of to extract the relevant attribute and returns the result to the mediator. Since only one such wrapper would need to be built for a given type of information source, which reduces the complexity of the communication among heterogeneous information sources and makes it easy to add new sources.

2.3 Mediators

According to Wiederhold[11], who first proposed the mediators for intelligent information integration, a mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications. In InfoHub we are focus on three functions of the mediators. *Information explore* will select appropriate sources according the query from the system or user interface. *Access plan* will determine the appropriate order of data manipulation to generate the requested information. *System learning* in InfoHub currently support two forms of learning. First, it provides the mechanism for pre-fetch relevant information according certain domain knowledge. Second, it has the capability to cache frequently retrieved information. Both forms of learning can improve the efficiency of the system. For example, a mediator for "air quality" might know that related information sources are EPAs air quality monitoring database and the weather information in the Weather Bureau. When the mediator receives the query, say for the information regarding "currently air quality in certain area", it will know forward the query to those sources. Moreover, by pre-defined domain knowledge that wind direction and wind speed might effect the situation of air pollution, the mediator would generate a subquery for fetching the relevant information of wind direction amd wind speed. A simplified query algorithm for the mediator can be shown as follows,

```
Procedure Mediator(File*input_query){
   query_tokens=decompose_query(input_query);
   object_query=analysis_query(query_token);
   make_subquery(object_query);
   send_subquery();
   while((not success) and (not timeout))
      pause();
   result_file=generate_result();
   return(result_file);}
```

2.4 Intelligent Interfaces

In general, the goal of the interface in InfoHub is to provide a platform-independent tool for displaying and exploring the information that are returned as a result of queries or some certain applications. We have implemented a home-page based user interface for submitting queries and exploring the results. An important advantage of using home-page based interface is its widespread and popularity currently. Therefore, anyone on the Internet is able to use InfoHub for retrieving information via home-page based interface.

There are two features making our interface module in an "intelligent" fashion for the applications in specific domain. First, we maintain a case base and the related reasoner for dealing with the incomplete information of submitted queries. For example, a query is submitted to ask "current air quality", but does not specify which area is interested. When such a query is detected, the interface will invoke the case base to reason a suitable form according the relevant previous experience. Such function is important in a dynamic environment, our interface can help the new users interact with the system for the first time. Second, the interface provide a mechanism for intensional query answering[5], that means user can use kind of *fuzzy linguistic terms* to submit the queries. For instance, an user might ask for "the air quality in north of Taiwan". Obviously, the "north of Taiwan" is a fuzzy term regarding with spatial relation. The interface will employ fuzzy reasoning to handle such a query and tailor it in a suitable form, then send tailored query to the mediator for further processing.

2.5 Auxiliary Module

Broadly speaking, the goal of auxiliary module is to enhance the overall capabilities of InfoHub. There are currently four components to form the auxiliary module. The new components can surely be joined in the future for better efficiency and performance.

We briefly discuss the functions and ideas of these four components.

Metadata Server: ,One difficulty may raise when creates a mediator is that we may not know which information source contains the information desired. In addition, even if the data source is known, the mediator may not know how it can be accessed. The InfoHub solve this difficulty by building a metadata server. The metadata server stores the relevant data of each information source and serve as a role of *yellowpage* for the mediator. Our metadata scheme is adopted from the two-level metadata dictionary that proposed in [9]. In the lower-level metadata dictionaries, we store the information regarding the each information source including the universal retrieval location (URL), data model, local constraints, etc. In the high-level metadata dictionaries, we may store the global constraints, data directory, and the domain knowledge of information sources. We use catalog to represent the metadata. Since the catalog not only represents the relationship between data item, but it represents the metadata descriptively or procedurally, as well as has the capability of reasoning. Therefore, the metadata server can be viewed as a simplified expert system and it can store and reason the related knowledge of information sources.

Knowledge Server: A simplified knowledge base management system (KBMS), which maintains and reasons with models of specific domain will be constructed in the knowledge server. We adopt the rule-based neural network approach [7] to implement the KBMS. In this approach, the rule base for a certain domain are converted to a neural network by performing a mapping: each domain attribute or concept is mapped into a neural node and each rule is mapped into a connection. Then some specific learning algorithms are applied for adjusting these rules to a consistent and completed circumstances. Finally, the refined rules are restored into the knowledge base. By incorporating the knowledge server, the mediator therefore can enhance its capability and become more flexible in terms of selecting the most appropriate information sources to answer a query.

Mediator Authoring Toolkits: Constructing a mediator can be very complicated and time-consuming, so we provide a set of tool to assist the design, coding and editing during the implementation of mediators. Since the mediators in InfoHub are typically written in Loom language (it compiler is based on Common LISP), for the sake of compatible, most of the toolkits will be coded in Common LISP. These toolkits will present the system designer or manager with a number of options on how to gather information. When the designer selects one or more of these approaches, then the toolkits will automatically generate the relevant rules of retrieval and integration to be placed in the mediator. Thus, the process of constructing a mediator can be partly automated.

Constraints/Notification Manager: Information sources are scattered over multiple network environment and the connection among them are usually in a loosely coupled fashion. Such situation can hardly support a transaction monitoring or management across multiple information sources. For example, EPA keeps data about the stations for air pollution monitoring. This data should be consistent with the data of station in the Weather Bureau. Thus, EPA could integrate the monitoring data in same location to forecast the trend of air pollution. Both information sources, however may be set up in an entirely different system. Currently, such situation usually are monitored or enforced by humans, in an ad-hoc fashion.

The constraints/notification manager, CNM in InfoHub provide a mechanism to identify how the data item in each information source may be read, written, monitored and even notified. Since it is generally not possible to achieve the totally consistency in a loosely coupled environment, CNM adopt a "*soften*" guarantee strategy to ensure the consistency that similar to [3]. For example, CNM can guarantee a constraint is true every weekday, or a constraint is true if a certain "Flag" is set. Besides, when a constraint is violated under soften guarantee strategy, CNM will notify or alert the system and relevant applications automatically.

3 An Example

In this section, we will describe an application in the environmental monitoring domain to identify the operations of retrieval and integration could be automated and interoperated in a heterogeneous information sources environment. Let us suppose that Environmental Protection Agency (EPA) keeps a relational database, Air Quality Monitoring Database (AQDB), containing a relation called *PSIvalue* having the schema (Name,Date,Time,PSIvalue). Thus, this relation may contain a tuple of the form (Taipei, 10/5/1995, 17:25, 120) denoting that the place, Taipei, at 17:25 on 10/5/1995 has been detected a PSIvalue of 120 . Other tuple in the relation PSIvalue may be similarly interpreted. Suppose there is another relational database, Weather Database (WDB), maintaining by Weather Bureau containing a relation called *Wind* having the schema (Name,Date,Time,Speed,Direction). A tuple of the form (Taipei, 10/5/1995, 17:25, 20, N) in the relation specifying that the wind speed and wind direction in Taipei at certain time. In addition, a satellite image database (ImgDB) may be maintained by the Department of Map in the local government. To clarify the concept of the application, consider the following example:

Example: Report the wind speed and wind direction information from WDB, when the PSI-value in AQDB has kept great than 100 for one hour continuously. Moreover, if the wind speed is less than 5 miles per hour, then retrieve the image which cover the 10 miles radius of the place (such as Taipei) from the satellite image database.

Integrating above information could assist EPA to investigate and estimate the effect of the bad air quality; EPA therefore could identify which area of the city should be alerted. To solve this problem may require to access at least three information sources and furthermore, the system should detect the certain conditions and performs the relevant actions automatically. It may be necessary to monitor a **relational database** (AQDB) about the PSIvalue, to access a **second relational database** (WDB) for wind speed and wind direction information, to access a **image database** (ImgDB) in order to retrieve the satellite image which covers the area within 10 miles radius of the place. The query of this example can be expressed in Loom language as follows.

```
(retrieve(?URLaddress)
    (and
    (CityLoc.  ?Cityname)
    (AQDB.City ?Cityname)
    (WDB.City ?Cityname)
    (ImgDB.City ?Cityname)
%--------------------
    (TimeOcr.  ?Occur)
    (AQDB.Time ?Cityname ?Interval)
    (continuous ?Interval ?Occur 1hour)
    (WDB.Time ?Cityname ?StatusTime)
    (close ?StatusTime ?Occur)
    (ImgDB.Time ?Cityname ?CapTime)
    (close ?CapTime ?Occur)
%--------------------
    (AQDB.PSI ?Cityname ?Interval
      ?PSIvalue)
    (> PSIvalue 100)
    (WDB.Wspd ?Cityname ?StatusTime
      ?Windspeed)
    (< ?Winspeed ''5 miles/hour")
    (WDB.Wdir ?Cityname ?StatusTime
      ?Winddirection)
    (= ?Winddirection
      one-of(''E", ''W", ''S", ''N"))
    (ImgDB.Range ?Cityname ?CapRange)
    (= ?CapRange ''10miles")
    (ImgDB.URL ?Cityname ?CapTime
    hspace.2cm ?CapRange ?URLaddress)
    ))
```

4 Discussion and Related Work

Research in database community has focused on building federated systems [10] or multidatabase systems[8] that combine multiple databases and provide uniform access. These systems are usually to first define a global schema, which integrates the information available in the diverse local databases or information sources. Since building an integrated global schema is labor intensive and hard to maintain, this approach is not possible to scale to the large number of evolving information sources.

Recently, several systems for accessing multiple information sources are being built on the notion of a *mediator* or *intelligent agent*. The Carnot project[6] uses a knowledge representation system to build a set of articulation axioms that describe how to map between SQL queries and domain concepts. Yet after the axioms are built the domain model is no longer used or needed. It is unlikely to commit the flexible and reusable requirement. SIMS[1] also adopts Loom language as a tool for domain modeling. It requires every information sources to map to a class in the Loom system and selects the relevant sources using a set of transformation rules. Almost all the operations in SIMS are heavily dependent on Loom language, so it is hard to improve the system flexibility in a certain view. The components of auxiliary module in InfoHub are most codes in Common LISP and these components can be dynamically added on or removed. This ability allow InfoHub to provide greater flexibility. The TSIMMIS project[4] use an Object Exchange Model, OEM as a common model for information processed by its components. TSIMMIS is not likely a knowledge-based approach but in an object-based fashion. In contrast, the Loom language and Common LISP are adopted as the uniform protocol and communication language in InfoHub, thus provides InforHub more expressive power for retrieving information in a knowledge-based style.

One of the distinguish features of InfoHub is that it provides an active fashion to retrieve and integrate the relevant information from multiple information sources. Most of the systems mentioned do not support such function, they only process the query-answering in a passive characteristic. Currently, a prototype of InfoHub has been built that can be used in the domain of air quality monitoring. Future work will focus on building the auxiliary module to extend the capabilities of the system. Furthermore, the ability of *knowledge discovery (data mining)* in InfoHub will also be explored in the future.

References

[1] Arens, Y.,C. Knoblock, and W. Shen, "Query reformulation for dynamic information integration," *Journal of Intelligent Information Systems,* vol. 6, no. 2/3, pp. 99-130, 1996.

[2] Borgida, A. "Description logics in data management," *IEEE Trans. on Knowledge and Data Engineering,"* vol. 7, no. 5, Oct. 1995.

[3] Chawathe, S. S., H. Garcia-Molina, and J. Widom, *Constraint management in loosely coupled distributed databases* Technical Report, Computer Science Department, Stanford University, 1993.

[4] Chawathe, S. S. *et al.,* "The TSIMMIS project: Integration of heterogeneous information sources," in *Proc. of IPSJ,* Japan, 1994.

[5] Chiu, W. W., R. C. Lee, and Q. Chen, "Using type inference and induced rules to provide intensional answer," *Proc. of IEEE Int'l Conf. on Data Engineering* pp. 396-403, 1991.

[6] Collect, C. M. N. Huhns, and W. M. Shen, "Resource integration using a large knowledge base in Carnot," *IEEE Computer,* pp. 55-62, Dec. 1991.

[7] Fu, L. M. and L. C. Fu, "Mapping rule-based systems into neural architecture," *Knowledge-Based Systems,* vol. 3, no. 1, pp.48–56, Mar. 1990.

[8] Litwin, W., L. Mark, and N. Roussopoulos, "Interoperability of multiple autonomous databases," *ACM Computing Surveys,* vol. 22, no. 3, pp. 267-293, 1990.

[9] Pan, M. J., S. K. Chang, and C. C. Yang, "A two-level metadata dictionary approach for semantic query processing in multidatabase systems," *Int'l Journal of Software Eng. and Knowledge Eng.,* vol. 3, no. 2, pp. 231-255, 1993.

[10] Sheth, A. P., and J. A. Larson, "Federated databases systems for managing distributed, heterogeneous, and autonomous databases," *ACM Computing Surveys,* vol. 22, no. 3, pp.183-236, 1990.

[11] Wiederhold, G., "Mediation in the architecture of future information systems," *IEEE Computer,* vol. 25, no. 3, pp. 38-49, Mar. 1992.