Intelligent Computing Technology and Automation
Z. Hou (Ed.)

doi:10.3233/ATDE231271

Heterogeneous Data Integration and Transformation Scheme Based on XML and Agent

Lei XING[a,1]
[a] Huazhi Excellent Productivity Promotion Beijing Co., Ltd, Beijing, 100028, China

Abstract. In order to solve the problem of data integration and sharing in heterogeneous databases, a heterogeneous database integration and transformation scheme based on Agent and XML is proposed. This model utilizes XML technology to construct a middleware between the client and database server, using mobile agents as the logical model framework to handle the autonomy and distribution issues of member systems in multi database systems. Then, the communication problem in the middleware system is solved by encapsulating the sent KQML message into an XML format document. The system testing results demonstrate that each Agent in this method can independently complete their own functions and cooperate with each other, which effectively solves the problem of intelligent integration of local database data, and it also has good adaptability to complex network environments.

Keywords. heterogeneous data; XML; Agent; DTD; database

1. Introduction

Database integration is the need for collaborative cooperation between various systems or departments within an enterprise. Its goal is to achieve data sharing among various heterogeneous data sources, thereby effectively utilizing resources and improving the performance of the entire application system. With the emergence of distributed processing requirements for enterprises and the rapid development of Internet information networks, current network information systems have put forward higher requirements for data exchange efficiency, system load balancing, data exchange security, and transplantation between heterogeneous platforms. Therefore, database middleware has become the focus. In existing dynamic enterprise alliance networks or within universities, information queries between business systems often need to be conducted in a distributed data environment, where there are diverse heterogeneous data sources, including structured (data from traditional relational database management systems) Semi structured (XML data) and unstructured (such as email, video, audio). There have been some theoretical studies on the use of Agent technology in middleware both domestically and internationally, but there is still a lack of in-depth research on its application in database middleware. In recent years, the rapid development of mobile agent technology in the field of artificial intelligence has proposed a new approach and approach for the integration of heterogeneous databases; XML technology has gradually become an industrial standard for data representation and information exchange, providing a platform for data transmission for

[1] Corresponding Author: Lei Xing; Huazhi Excellent Productivity Promotion Beijing Co., Ltd, Beijing, 100028, China; queque0506@163.com

heterogeneous and different platform relational databases. In the process of querying these heterogeneous data sources, if you want to provide personalized and intelligent query services, you first choose Agent technology. Most existing methods use XML as the information carrier based on the independent existence of each heterogeneous information source, and design a multi-agent model for heterogeneous data interaction at the system application layer [1,2]. Agent technology can achieve the sharing and effective utilization of various heterogeneous databases, By using XML to achieve data exchange, the combination of the two can effectively improve user query efficiency in the integration process of heterogeneous databases

This article conducts in-depth research on the application of XML and mobile agents in heterogeneous data exchange, attempting to integrate dispersed resources into a unified platform or system through heterogeneous data exchange and sharing technology. We first analyzed the relevant technologies for heterogeneous data exchange and sharing, and explored the use of XML and mobile agents to integrate data of different types and formats into a unified data model. Then, using the mobile agent as the logical model framework, establish a virtual database and combine it with XML to implement middleware for data conversion. In this process, XML serves as the standard format for data exchange, while mobile agents serve as intelligent entities that perform data conversion and integration tasks, completing the merging and sharing of data information resources between different databases. Finally, through specific case testing and analysis, it is proved that this solution can achieve automatic data conversion and integration by using mobile agents in middleware, and provide advantages such as flexibility, automation, scalability, and efficiency.

2. Research on Heterogeneous Data Exchange and Sharing Technology

2.1 XML and Heterogeneous Data

XML (Extensible Markup Language) is a markup language used to describe and represent data, which uses tags and elements to organize data and adopts the concept of hierarchical structure. XML uses a document model, while databases use a relational model. XML can represent both unstructured and semi structured data, while databases are more suitable for storing structured data. DTD and Schema are specifications used to define the structure and data types of XML documents. They allow design developers to define elements, attributes, data types, constraints, etc. to ensure the structure of XML documents and the validity of data. By using DTD or Schema, design developers can define their own data models based on specific needs, thereby achieving flexible data representation and validation. XML also provides open languages and APIs, such as DOM (Document Object Model), SAX (Simple API for XML), and StAX (Streaming API for XML). These APIs allow design developers to programmatically read, write, and manipulate XML documents, enabling flexible processing and integration of XML data. Due to the enormous value demonstrated by XML, multiple database vendors have integrated XML resources and functionality into their products [3]. A basic function is to provide an XML translator to translate specific database format data into XML documents, which enables database management systems to meet a large number of web-based applications, of course, implemented through XML, as shown in Figure 1.

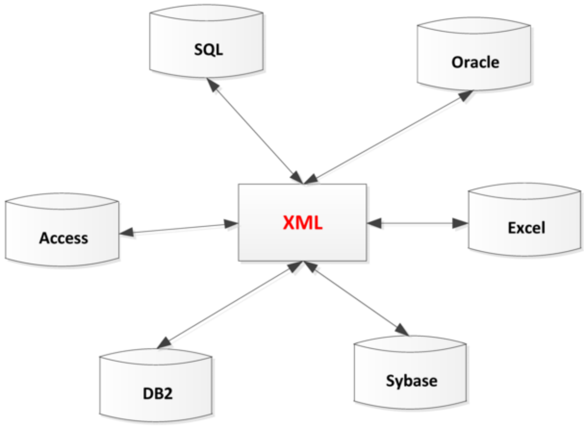

Figure 1. Principles of XML Translators.

2.2 Agent Technology

With the development of artificial intelligence, the term "Agent" has found its place in artificial intelligence research, describing entities that exhibit intelligent behavior and possess qualities such as autonomy, responsiveness, initiative, and social skills. Mobile Agent is a program that can autonomously determine its behavior in heterogeneous networks. It has characteristics such as autonomy, mobility, collaboration, security, and intelligence. Mobile Agent is a distributed computing technology different from DCOM and RMI. It can autonomously move from one host to another on the network and represent the user to complete designated tasks, such as searching, filtering, and collecting information. In traditional technology, every time the client queries heterogeneous data sources, it must interact through the Internet, resulting in a significant cumulative network delay that affects the efficiency of the system. In mobile agent technology, the mobile agent first interacts with the client locally, and after obtaining data such as running code, status, and attributes, the mobile agent moves to the target data source to run locally. Therefore, it reduces network interaction and improves the system's ability to adapt to heterogeneous, low bandwidth, and unstable network environments. It is of great significance for improving the overall performance and reliability of enterprise information integration systems [3].

3. Heterogeneous Database Integration and Exchange Model Based on XML and Agent

3.1 Overall Model Hierarchy

A good heterogeneous database integration solution must effectively address the heterogeneity of data, transparency of access, and autonomy of data. This article proposes a solution for heterogeneous database integration based on mobile agent and XML technology, which can effectively solve the problems encountered in heterogeneous data integration [4]. The implementation idea of this plan is:

- Data source integration: Firstly, it is necessary to integrate data sources from various heterogeneous databases. This can be achieved by using ETL (Extract, Transform, and Load) tools or custom data integration programs. Data integration programs can extract data from various databases and convert it into a unified XML representation.

- XML data view design: Next, a unified XML data view needs to be designed to represent the integrated data. This XML data view should be able to contain all the data that needs to be shared in various databases, and provide a consistent data structure and standardized data format.
- Data access interface: In order to provide a unified XML based data access interface, technologies such as web services or RESTful APIs can be used. These interfaces can receive requests from applications, extract corresponding data from the XML data view based on the parameters and conditions of the request, and return it to the application in XML format.
- Query and operation support: Virtual databases should support common query and operation functions, such as data retrieval, filtering, sorting, aggregation, etc. This can be achieved by implementing query languages (such as XPath or XQuery) and operation methods on XML data views.

This scheme uses XML as the global data schema and utilizes mobile agent technology to intelligently complete user query requests, effectively integrating heterogeneous databases and providing users with an information sharing platform that can transparently access and integrate multiple heterogeneous databases, as shown in Figure 2.

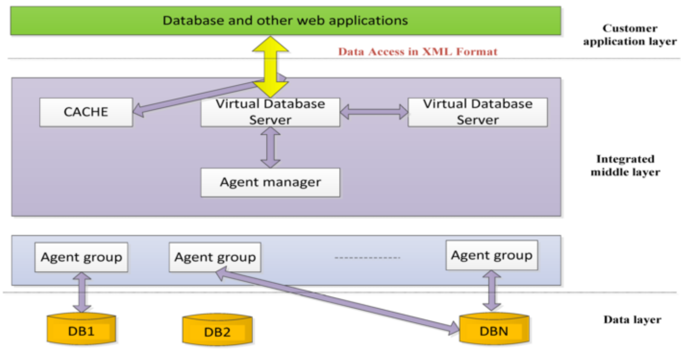

Figure 2. Heterogeneous data integration and transformation model architectures.

3.2 Data Integration Based on XML

We establish a standard basic database based on the established standards, and use Agent technology and XML data binding technology to collect heterogeneous data from the original management information system using the Agent data collection module. By standardizing the conversion of heterogeneous data, the XML message format is sent to the Agent data organization module in the data center, which parses the XML message. After successful data verification, it will be imported into the standard basic database as the basic data for the data center. Agent messages use common XML data vocabulary to describe all objects, and XML data binding and database technology have unique advantages. The primary point of integration between databases such as Oracle, SQL Server, and XML is to create XML files based on SQL data. The agent generates different scripts based on the collected data, thereby converting the database data into XML message format. The XML message processing function of the data organization module not only parses data information from the XML message, but also includes the query condition generation processing function, which generates query request XML messages based on data standards.

The specific implementation method for data integration using XML is shown in Figure 3. The input of the integration tool is data sources a_1 to a_n. Next, you need to design an XML data model to represent the integrated data. This data model should be able to include all the data that needs to be shared from various data sources, and provide a consistent data structure and standardized data format. By using XML parsers and transformation tools. The process of data extraction and transformation may involve operations such as data cleaning, data type conversion, and data merging to ensure data consistency and accuracy. If the representations of the same entities in different data sources are inconsistent, define rules to convert them into a unified format and store them in a policy library. Finally, integrate the data that has undergone conflict detection and resolution into a unified XML data model. In this way, a unified data access interface can be provided through the XML data model for use by applications. After receiving the XML file, the mobile agent can display the data to the user based on their needs and interface design. This can be achieved through the display function within the mobile agent or integration with other applications.

Figure 3. Implementation methods for data integration by XML.

3.3 Design of Communication Mechanism

Establish a registry to store and manage information for all agents. Each agent registers its name, communication address, communication port, and collaboration capabilities with the registry at startup. Other agents can obtain information about other agents by querying the registration center. Each agent can specify the names, communication addresses, communication ports, and collaboration capabilities of other agents in its own configuration file. At startup, the agent reads the configuration file and establishes connections with other agents. Design a communication module in each agent to manage and forward communication between agents. Each agent connects to a proxy server and provides its own information during registration. The proxy server is responsible for forwarding messages to the target agent. The communication design implementation framework is shown in Figure 4. Each communication module consists of three parts: an information table, a message cache pool, and a communication service manager. The information table is used to record some basic features in the agent [6].

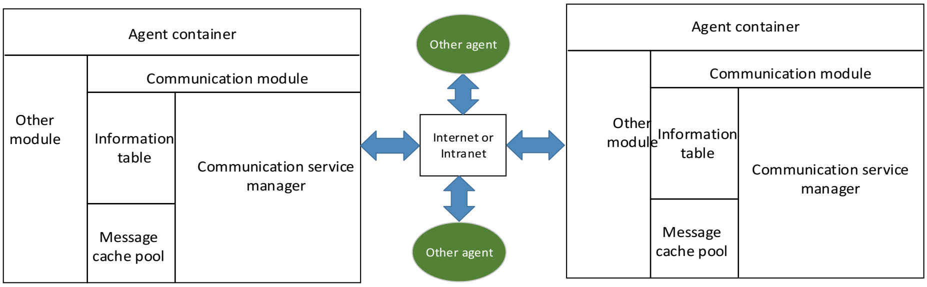

Figure 4. Agent communication implementation framework.

The information table mainly includes the following two parts:

(1) Static information tables can be stored using databases or files. Each agent writes its own static information to the static information table when starting. Other agents can query the static information table to obtain static information of other agents. Through a static information table, each agent can understand the identity, performance, IP address, and other information of other agents, thereby achieving better communication and collaboration;

(2) Whenever an agent starts and establishes communication with other agents, it adds its own information (agent ID, communication status, timestamp, etc.) to the dynamic information table. In this way, other agents can understand the existence and communication status of the agent. In order to maintain real-time updates of the dynamic information table, it is possible to regularly check the communication status of each agent. If an agent does not communicate with other agents for a period of time, its status can be updated to offline or unavailable. The communication service manager is mainly responsible for encapsulating and parsing messages, controlling message synchronization transmission, and avoiding causal errors caused by synchronization issues.

Agents in the system need to communicate with each other to collaborate and complete a complex task. In order for each agent to understand the content of communication and respond accordingly, it is necessary to define a common communication mechanism for them. The selection of KQML (Knowledge Query and Manipulation Language) provides a flexible way to represent and transmit knowledge, objectives, and operational requests. It adopts a message based communication model, where agents communicate by sending and receiving messages:

- Message format: KQML messages consist of a keyword and a set of parameters used to represent knowledge, objectives, and operational requests. Keywords are used to indicate the type and intent of a message, while parameters are used to convey specific information.
- Semantic expression: KQML provides a set of predefined keywords and parameters to represent common semantic concepts such as inquiry, notification, request, etc. In this way, agents can understand and interpret the meaning of messages between themselves.
- Communication Protocol: KQML defines a set of communication protocols that specify the sending and receiving methods of messages, as well as the interaction rules between agents. In this way, the agent can communicate according to the protocol, ensuring the correct transmission and processing of messages.

4. Case Analysis

The case study takes the data integration process of the teaching resource management system in a certain university as an example. Due to the different data warehouse methods used in various informationization processes, the school needs to fully utilize these shared data and information through data integration and transformation to organize, command, and coordinate teaching activities throughout the school, thereby further improving resource utilization and improving the overall level of teaching management. Firstly, it is necessary to write DTDs to define the structure and constraints of XML data based on the data structure and requirements of university teaching resource systems. DTD can define elements, attributes, entities, and constraints to ensure data consistency and integrity. Convert to a relational database: Once the DTD is defined, XML data can be converted to a relational database using the DTD mapping tool or a custom conversion program. This can be achieved by parsing XML data and creating corresponding relational tables and fields based on the rules defined by DTD. After converting XML data into a relational database, you can use data import tools or custom data import programs to import XML data into a relational database. During the import process, data integration and transformation can be carried out as needed to ensure data consistency and integrity. The structure of this DTD mapping into a relational database is shown in Figure 5.

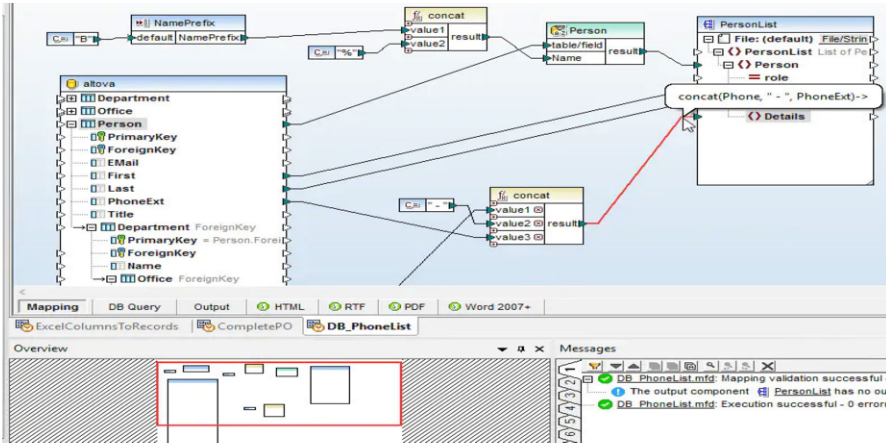

Figure 5. Mapping DTD to the structure of a relational database.

The specific implementation steps are described as follows:

(1) Install the Agent subsystem on each information node of the collaborative information subsystem, that is, on each node on the network Install the developed Agent subsystem on the local servers of each department. This can be achieved by transferring the deployment package or installation program of the subsystem to various departments and installing and configuring it on their local servers Testing and validation to ensure that the Agent subsystem runs normally on local servers in various departments, and collaborates and communicates with other systems

(2) Identify a suitable host or server to serve as a collaborative information system domain server. The server should have sufficient computing resources and storage capacity to support the operation and storage requirements of collaborative information systems. Run the installation program of the core agent and follow the instructions of

the installation wizard. Configure the core agent according to the requirements and configuration requirements of the collaborative information system.

(3) Based on the requirements and task allocation of the collaborative information system, determine the specific functions and tasks of each agent in the agent subsystem of each node, involving different functions such as data collection, data processing, decision-making, and task allocation. Based on the actual situation of each node's Agent subsystem, write configuration files or set interfaces so that users or administrators can easily configure the address information of the core Agent. This ensures that each node Agent subsystem can be correctly connected to the core Agent.

(4) Configure the address information of the core agent in each agent subsystem, register the information of each subsystem in the core agent, and achieve effective communication and collaboration between each subsystem and the core agent. The core agent can regularly check the registry or database to ensure that the information of each subsystem is up-to-date

(5) Deploy the mobile agent application of the subsystem and place the code and resource files of the mobile agent in the specified directory of the subsystem. Start the mobile agent running environment of the subsystem to ensure that the mobile agent can run normally and communicate with other components.

Each heterogeneous data exchange with XML requires a different transformation mechanism, and this article uses DTD mapping as an example to illustrate. Store the conversion mechanism in the core agent system and apply for conversion services to the core agent through each sub agent system. During the conversion process, internal intelligent user agents, local information agents, and collaborative information agents can be utilized to complete the conversion task. The sub agent can perform subsequent processing based on the returned results, such as storing transformed data, presenting it to users, etc. Part of the key codes for this process are depicted as follows:

```
#Build XML Request
Def build_ XML_ Request():
#Create Root Element
Root=ET.Element ('Request ')
#Add child elements and data
Agent_ Name=ET. SubElement (root, 'AgentName')
Agent_ Name. text='Agent1 '
Task=ET. SubElement (root, 'Task')
Task. text='PerformTask'
#Convert XML to a string
XML_ Data=ET. tostring (root)
Return XML_ Data
#Main program
If__ Name__='__ Main__ ':
#Define the communication address of the target agent
Target_ Agent_ URL=' http://target_agent.com '
#Build XML Request
XML_ Request=build_ XML_ Request()
#Send XML request and obtain response
Response=send_ XML_ Request (target_agent_url, xml_request)
#Processing response data
If response.status_ Code==200:
#Parsing XML Response
```

XML_ Response=ET.freestring (response. content)

…

According to the configuration of the experimental environment mentioned above and the testing requirements, prepare database information of different sizes, and the response time to the information agent. The experimental results is shown in Figure 6. It can be observed that as the amount of data increases, the information agent needs to process more data and perform more complex operations, resulting in an increase in response time. Therefore, we regularly monitor and evaluate the performance of the information agent, identify potential bottlenecks and issues, and make corresponding optimizations and adjustments. Immediately trigger the corresponding processing logic when data changes. Regardless of the method used, the key is that the information agent can timely perceive changes in data and transmit the changed information to the information management system. Therefore, the information agent has the ability to monitor and respond in real-time, and can quickly capture changes in data and transmit the changed information to the information management system. It can be said that this solution helps to improve the response speed and decision-making efficiency of the system, that is, the design of the middleware system can meet the needs of real-time data.

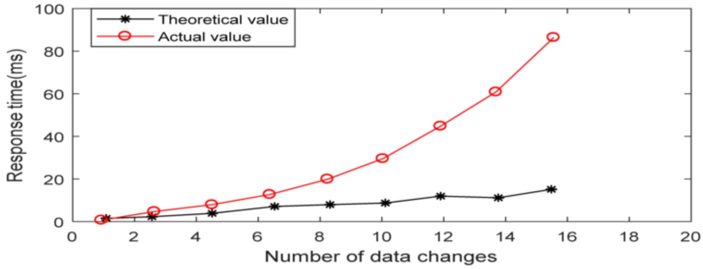

Figure 6. Response time curve of information agent

5. Conclusion

To solve the problem of heterogeneous integration and sharing of information in distributed network environments, this article proposes an information integration middleware framework model based on XML and mobile agent after in-depth research and analysis of related technologies such as XML and mobile agent. In information integration middleware, mobile agents are responsible for transferring and exchanging XML data between different data sources. This can be achieved by using appropriate communication protocols and technologies to integrate the mobile agent into a unified XML data model after receiving XML data. Through the XML data model, a unified data access interface can be provided for application use. The case analysis results indicate that this method can help achieve the integration and sharing of heterogeneous data, as well as the efficiency and flexibility of data exchange, and improve the overall performance and reliability of information integration systems.

References

[1] Huifang L. Research on the Integration of Heterogeneous Database based on Mobile Agent and XML. Computer Development & Applications, 201, 24(2): 38-40.

[2] Zhou Min. Research and Application of Heterogeneous Database Integration Based on Multi Agent and XML [D]. Guangdong University of Technology, 2023.

[3] Qi Z, Pan-Qing W, Xiao-Hui LI, et al. Research of Information Integration Middleware Based on XML and Mobile Agent. Computer Technology and Development, 2010, 154-157.

[4] Huifang L. Research on the Integration of Heterogeneous Database based on Mobile Agent and XML. Computer Development & Applications, 2011, 24(2): 38-40.

[5] Zhong-Guo W. To Implement the Heterogeneous Data Acquisition and Integration Based on Agent in Data-center Constructing of University. Computer Knowledge and Technology, 2009, 5(14): 3609-3610.

[6] Gao Fei. Research and Realization of the Middleware of the Heterogeneous Financial System, Ship Electronic Engineering, 2015, 35(9): 114-118.

[7] Al-Ghamdi N, Saleh M, Eassa F. Ontology-Based Query in Heterogeneous and Distributed Data Sources. Asian Journal of Information Technology, 2010, 9(9):183–195.

[8] Yu S Y, Yim J H. Design and Implementation of Data Replication Web Agent between Heterogeneous DBMS based on XML. Journal of Navigation and Port Research, 2002, 26(4):427-433.

[9] Bo, Chen. XML-based agent communication, migration and computation in mobile agent systems. Journal of Systems and Software, 2008, 81(8):1364-1376.

[10] Kumar P, Samanta T, Patra D, et al. Coordinately Tethered Iron(III) Fluorescent Nanotheranostic Polymer Ascertaining Cancer Cell Mitochondria Destined Potential Chemotherapy and T1?Weighted MRI Competency. ACS Applied Bio Materials, 2022, 5(3):1284-1296.

[11] Indumathi T, Krishnamoorthy N, Valarmathy R , et al. Green Synthesis of α-Fe2O3 Nanoparticles Mediated Musa Acuminata: A Study of Their Applications as Photocatalytic Degradation and Antibacterial Agent. Nano Biomedicine and Engineering, 2022, 14(3):254-262.