The Research of Heterogeneous Database Middleware Model

YE Xiao-Tong
Department of Computer
Shaoxing College of Arts and Sciences
Shaoxing Zhejiang 312000
CHINA

Abstract: In this paper, we present a database middleware with Agent characteristics. The middleware inherited the Agent's self-studying nature, cooperating characteristics and so on. Data query and data sharing among heterogeneous database could be achieved by the database middleware cooperation, the heterogeneous data source could be dynamically expanded and "plug and play". The middleware has been applied in the "Court Related Case Search System", and the result demonstrated it worked well.

Key-Words: Heterogeneous Database; Middleware; Agent; Model; Data Sharing

1 Introduction
With the advent of the information age, more and more information applications based on database system has been developed and deployed. Due to the historical reasons of development of database technology, various enterprises adopt different database system, or even various business sectors in one enterprise using different database systems. Now the Integrated application system in enterprise has to be faced the heterogeneous database problem.

The most direct method to solve the problem of heterogeneous database is integrating them into a unified database, but the integration process is very complicated. At present, some companies can provide heterogeneous database middleware products based on the principle of integration, such as IMB’s "DB2 DataJoiner" and Sysbase's "Enterprise Connect", but these products are only suitable for particular database system. In practice, many applications only need the data in heterogeneous database, so the best way to solve this problem is using an agile and effective heterogeneous database middleware, which can provide data sharing service. The researches on these middleware are absorbed in ontological theory and CORBA technology[1][2]. This paper presents a design of heterogeneous database middleware based on software agent with features of intelligence and cooperation.

This paper is organized as follows. In Section 2 we mainly analyze the structure of software agent model and introduce the principle of work of every component in software agent model. We present in Section 3 a heterogeneous database middleware model based on software agent, and explain the work process of the model. In Section 4 we analyze the key points about the heterogeneous database middleware model, and reveal the design of communications in models. Finally, we conclude in Section 5.

2 Software Agent Model
Software agent model[3][4] is composed of five components, as illustrated in Figure 1:

![Figure 1: Software Agent Model](image)

(1) Knowledge Base: Agent should take all actions with rationality on the basis of Agent’s knowledge, including self knowledge and the understanding of the external world model. As a result, there should be a component for the storage of the knowledge as a basis for Agent to carry out the next step with initiative.
(2) Task Schedule Component: As a manifestation of the vitality of Agent, the planning component can generate a plan of action according to the content of Knowledge Base, triggered by a change in the state of Knowledge Base or the generation of an external request.

(3) Task Execute Component: is to realize Agent’s ability to respond. A component is required to take some action to adapt to the environment demand after Agent perceives a change in the external world. An Agent may need simultaneously a number of tasks to be done to meet their own and external requirements, so the task implementation component should be equipped with synchronicity and concurrency.

(4) Communications Component: is the embodiment of Agent’s sociality. An Agent’s collaboration with other Agents is built on the basis of interaction requiring a communication channel, so the Agent should have a communication component. Besides, changes in the external world can be perceived by Agent through communications.

(5) External Sensing Component: To improve its adaptability, Agent needs a sensing component to the external world for the collection of changes in the external world model and updating of contents of Agent’s Knowledge Base, as a foundation for Agent’s ability to respond.

An Agent, composed of the above five components, can effectively sense the external world, and complete tasks according to external requests individually or in collaboration with other Agents with intelligent and collaborative features.

### 3 Software Agent-based Database Middleware Model

The design of the database middleware in this paper has absorbed the characteristics of Agent model and applied the software Agent-based idea to the design of middleware with intelligent and collaborative features. Database middleware model is composed of the following components, as illustrated in Figure 2.

(1) Middleware interface: provides task request interface for users and other middleware and returns the results.

(2) Middleware registration component: similar to the external sensing components of Agent, receives information from other middleware and to register or update in the Knowledge Base to lay the foundation for the interaction with other middleware, i.e., to build a collaborative environment for middleware.

(3) Database component: all the operations of middleware on the database, including the database connection, SQL statement conversion, SQL statement implementation, are encapsulated in this part.

(4) Task scheduling component: the task scheduling component is the core of the middleware. After receiving an external request, the middleware will complete the task according to the content generation task module in the Knowledge Base, and return the combined results to the external request.

(5) Task component: As the task completion entity generated by the task scheduling component, it can complete the task within the middleware, or interact with other middleware to complete the task.

![Software Agent-based database Middleware Model](image)

Middleware’s specific workflow can be described in three phases: initialization, response to users’ requests, and response to requests of other middleware.

(1) Initialization: After the startup, the middleware creates a connection with the local database based on the database connection setup, starts DB Agent to broadcast in the network the information from the middleware and the local database server, and starts Registration Agent to monitor the information from other middleware in the network to update the Knowledge Base. During the running process of the middleware, DB Agent and Registration Agent will perform periodic...
broadcasting and monitoring as a foundation collaborative environment construction.

(2) Response to users’ requests: After receiving a user’s request, the task scheduling component generates a Main Task to fetch the collaborative middleware information from the Knowledge Base and generates, according to the number of collaborative middleware, Sub Tasks, each corresponding to a collaborative middleware. Sub Tasks establish connections with and deliver the inquiry request to the corresponding middleware. After the completion of inquiry request task, the corresponding middleware returns the results to Sub Tasks, and finally, each Sub Task returns the results to the Main Task, which then returns the combined results to the user.

(3) Response to requests of middleware: Upon receiving and accepting a request of another middleware after verification, the task scheduling component generates a Query Task. The Query Task converts the request into a local SQL statement in accordance with the mapping rules in the Knowledge Base, implements the task through the database component, and then returns the implementation results to the other middleware.

4 The Key Point of Model

A middleware for collaborative work needs to address the issue of the collaborative environment construction for the middleware at first. A good communication method is the key point. Secondly, the database heterogeneity will be resolved mainly to settle the heterogeneity of the data mode and the local DBMS (Database Management System). The data mode heterogeneity can be regarded as the heterogeneity of the data representation since the same information has different representation form in various DBMS. Local DBMS heterogeneity refers to the difference in type, i.e., access mode and data type of each DBMS. Apart from the above issue, data inquiry and security should also be taken into consideration.

The understanding of the latest status of other middleware, such as locations, ports and database status, etc. is a prerequisite for the collaborative completion of tasks by middleware and a basis for the whole system to perform coordination timely and accurately.

The existing solution for reference: After the startup of a middleware registration server in the application system, each middleware should register to the server immediately upon the startup and provide necessary registration information, including server name, IP address and monitor port number, etc. From then, the middleware should find other ones capable of collaboration in the registration server and implement communication collaboration with them if a collaborative work is needed.

The deficiency of this solution is the higher application cost in contrast to the lower system stability since a separate registration server is required to provide services. If a registration server failure occurs, middleware will fail to work in collaboration, and the whole system will not be able to operate normally. Thus, this article will, in reference to the above-mentioned solution, use the collaborative feature of Agent to complete the middleware collaborative environment construction. And the main idea is to design a DB Agent and a Registration Agent inside the middleware for the collaborative environment construction as follows:

(1) When the middleware starts running, DB Agent loads the information of the local database server and the middleware, and sends the loaded information in a broadcasting way to the network.

(2) When the system is running, DB Agent broadcasts the information of the local database server in the network in a fixed time interval. Every time before the broadcast, DB Agent detects the local database status for other middleware to update the status of each database in the network and avoid unnecessary operation requests.

(3) Registration Agent monitors the broadcast information from other middleware’s DB Agent in the network, and check if this database information is already included in the internal database server list after receiving the information. If so, the status symbol of this database should be updated; if not, the information should be added into the list.

Thus internally, each middleware possesses a list of data source servers in the network with their updated information. A middleware can communicate with other ones for the collaborative completion of tasks based on the information list.

5 Conclusions and Future

In this paper, we propose a database middleware Model based on Agent and explain the key points in middleware design. The structure of middleware located among various heterogeneous databases and application systems will greatly reduce the workload of the upper-level application system development, and enable the good expandability and flexibility of the upper-level application system. In the development of “Court Related Case Search System”, we face a challenge of heterogeneous database access
since databases of intermediate courts are heterogeneous from those of local courts, the solution based on this database middleware solve the problem. The next step is to perform a further study and improvement of global inquiry control and security on the basis of the existing work.

References: