Research of Campus Heterogeneous Database Middleware Based on SOA

Song Haige¹, Zhang Zhibin²

¹College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China
Email: shg@hpu.edu.cn
²Department of Modern Education Technology, Henan Polytechnic University, Jiaozuo, China
Email: Zhangzhibain@hpu.edu.cn

Abstract—Along with the rapid development of campus information technology, many universities precipitates a great deal of information resources, how well make use of existing resources to build a centralized information resource management system is a serious problem. This paper designs a deep integration architecture of campus information based on SOA, adopts middleware technology to implement the integration scheme that highly requires a real-time data exchanging. The paper constructs a central database, which can synchronize its data to the corresponding application databases. Through the secure and reliable public data exchanging, all applications can be integrated on the basis of sharing public data and the data integrity, the data accuracy and the data consistency can be effectively ensured.

Index Terms—SOA; central database; data synchronization; middleware; public data exchanging

I. INTRODUCTION

With the rapid development of information technology in universities, many universities purchase and develop a number of applications after decades of information construction at the same time a number of information resources are precipitated. However, all these applications cannot be exchanged via the Internet and cannot share data eventually forming many islands of information, leading to repetitive construction and work. Therefore, the depth of information integration construction must integrate existing information resources and develop new resources, build a centralized information resource management mechanism to ensure that all applications can share data and achieve a real-time exchanging.

Data with different data sources and heterogeneous platform interfaces can be described in a unified and transparent data mode through the coupling way. System resources can be connected, integrated and collaborated though Web service in the process of data integration, as is a serious problem to be solved in the research of campus information integration platform. SOA (Service-Oriented Architecture) that has coarse-grained, loosely coupled, composite structure based on Web Services in particular provides a new solution for applications and data integration.

In this paper, the design of a deep degree of integration architecture for campus information based on SOA is proposed. New integration project can realize highly a real-time data exchanges through middleware technology. The paper constructs a central database, which can synchronize its data to the corresponding application databases. Through the public data exchanging system, all applications can be integrated on the basis of sharing public data.

II. SOA ARCHITECTURE

A. SOA Architecture

Service-oriented architecture is a component model [1], it can connect applications with different functional units (called services) through these well-defined interfaces and contracts. Interface that is defined in a neutral way can be independent of the handwork platforms on which they run, or the operating system and programming languages in which they are written, so that services based on such system can interact in a uniform and common way. SOA is regarded as an architectural style that emphasizes implementation of components as modular services that can be discovered and used by clients.

1) Services mainly have the following characteristics:
   a) Services may be individually useful, or they can be integrated—composed—to provide higher-level services. Among other benefits, this promotes re-use of existing functionality.
   b) Services communicate with their clients by exchanging messages: they are defined by the messages they can accept and the responses they can give.
   c) Services can participate in a workflow, where the order in which messages are sent and received affects the outcome of the operations performed by a service. This notion is defined as “service choreography.”
   d) Services may be completely self-contained, or they may depend on the availability of other services, or on the existence of a resource such as a database. A service might perform a task without needing to refer to any external resource, or it may have pre-loaded all the data that it needs. Conversely, a service that performs currency conversion would need real-time access to exchange-rate information.

B. Three Important Roles of SOA

SOA architecture is composed of service provider, service requester and service registry [2]. Basic operations include service registration and publication, service discovery and binding, as shown in Fig. 1. Service provider publishes service information to service registry.
A. Web Service Concept

Service requester locates a service that meets its needs through searching for service. Once service requester search for the suitable service, it will directly activate the service according to the description of information in registry.

B. Web Service Key Technology

1) Service requester: Service requester may be an application, a software module or another service in need of other services. It initiates a service inquiry from registry and binds a suitable service, then invokes the service according to interface contract.

2) Service provider: Service provider, which may be a network addressable entity, accepts and implements the user's request. It will publish own services and interface contracts to service registry so that service requester can discover and access the services.

3) Registry: Registry is a supporter of service discovery. It contains a repository of service and allows interested service requester to search for and access service provider’s interface.

III. WEB SERVICE

A. Web Service Concept

With the development of the Internet and relative technology, Web Service [3] is a product in a certain developing stage. Web Services have the interoperability on the complete different platforms, which is intended to achieve interoperability among all the applications through the Web standard.

Web services are modular components that may provide information to applications rather than to humans, through an application-oriented interface in a web environment. The information is described using standardized XML, so that it can be parsed and processed easily rather than being formatted for display.

Web services publish details of their functions and interfaces, but they keep their implementation details private; thus a client and a service that support common communication protocols can interact regardless of the platforms on which they run, or the programming languages in which they are written. This makes Web services particularly applicable to a distributed heterogeneous environment.

B. Web Service Key Technology

The key specifications used by Web services are:

1) XML(eXtensible Markup Language)—a markup language for formatting and exchanging structured data. XML language can transform data with different formats into the same structure and provide a unified data format for web service.

2) SOAP(originally Simple Object Access Protocol, but technically no longer an acronym)—an XML-based protocol for specifying envelope information, contents and processing information for a message.

3) WSDL(Web Services Description Language)—an XML-based language used to describe network service, or endpoint. A WSDL document can be used to dynamically publish Web service, to find a published Web Service and bind Web Service.

4) UDDI(Universal Description, Discovery and Integration) — a soap-based client function for a framework for describing and finding a web service. UDDI can access the agreements of registered information through registry.

Many other protocols that focus on security, asynchronous communication and semantic expression are gradually being added to Web service.

IV. DATA INTEGRATION BASED ON SOA

A. Data Integration Technology

Typical data integration solutions can be divided into two categories: One is the materialized method, and the other is the global model method.

Data Warehouse belongs to the materialized method whose integration strategy is to pre-process and convert data copies coming from several heterogeneous data sources according to a centralized and unified view requirement in order to conform with the model of data warehouse. The data-sharing integration of heterogeneous database based on the data warehouse model has the advantage of on-line analysis and data mining. Disadvantage is the duplicate storage and updating difficulty of data. This method generally applies to large enterprises for analyzing its vast historical data.

Middleware system is a global view, which presents a global model in the middle layer to hide data details of the underlying layer so that Integration of data source is regarded as a unified whole by users. The actual data is not stored in the middle layer under this system. It is suitable for the integration environment that is relatively fast for the speed of updating data, is impossible or difficult to load all data from data sources. When user submits a query statement, middleware will separate it and send it to the underlying servers with different data source. Because the difference in the types of data source servers, it can complete the consistency of service interface for data sources of heterogeneous databases through wrapper function layer. User's query based on the global model do not need to know the characteristics of each data source, middleware will divide query statement into sub-queries based on each local model of data source. Data integration project based on middleware, because of the advantages of real-time data exchanging and flexible scalability, is widely adopted. Considering middleware technology, the design of data integration layer based on SOA architecture is proposed.

B. System Architecture

System architecture [4] is divided into user layer, data
integration middleware layer and resource layer, which is shown in Fig. 2. Each data accessing operation that is initiated by transaction service can invoke this software layer. User layer may access the bottom of heterogeneous data source through an application interface that is provided by the middle layer. Data integration middleware layer, which is core layer of this architecture. This paper adopts the standard SQL language as query language to eliminate the difference between data sources of heterogeneous databases in the underlying layer. Resource layer, the bottom layer is mainly used to store and manage persistent data whose type may be text files, XML documents, relational database.

Main flow of the system is as follows: user layer submits the standardized SQL query statements, directly queries in the central database, gets the suitable query results and returns them to user; In order to ensure the data consistency between the central database and the underlying databases, the middleware layer of data integration adopts "receiving adapter \rightarrow filter \rightarrow converter \rightarrow router \rightarrow sending adapter" to complete the data exchanging process. Firstly, receiving adapter receives the data that waits to be updated from the application databases in resource layer and then submits it to filter. Secondly, according to configurable business rules and information standards filter filters out these data inconsistent with rules and standards and then submits the matching data to converter. Thirdly, converter encrypts and decrypts these data, generates the corresponding data packet and submits it to router. Fourthly, according to the updated plans router updates the corresponding data of the central database. Finally, according to the synchronous scheme synchronous component submits the data packet to router, synchronizing to the corresponding data of the corresponding application databases.

C. The Introduction of Main Components

1) Standardized query: This module may parse the submitted SQL query statement by user and verify its correctness of SQL syntax.

2) Adapter: Adapter is actually a interface by which information with different formats can be converted.

Adapter is available to support the following protocols or services: HTTP (Hypertext Transfer Protocol) and HTTPS (secure hypertext transfer protocol), JDBC (JAVA Database Connectivity), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SOAP (Simple Object Access Protocol), WSDL (WEB Services Description Language) and so on.

3) Central database [5]: Through the secure and reliable public data exchanging, all applications can be integrated on the basis of sharing public data and the data integrity, the data accuracy and consistency can be effectively ensured in the process of information integration.

4) Filter: According to configurable business rules and information standards, filter analyzes and deals with the data that waits to be exchanged, filters out these data inconsistent with rules and standards.

5) Converter: It converts the data to a suitable data format that can be received by receiver based on information standards.

6) Router: Based on configurable routing policy, the data can be securely exchanged and reliably transmitted between the application systems.

D. Public Data Exchange System

The data changing tracking components that are deployed in each application server are used to track the changing of data according to the data tracking schema, and then generate a relative data changing packet and submit it to the central database. Simultaneously, the data changing synchronous component is deployed in the central database server, which generates the sequence packets of updating data that are synchronized to each specific application database responding to the data Synchronization schema.

The logical structure of common data exchanging system is shown in Fig. 3.

1) Public data exchanging system function

a) Information standards: A series of standards should be set, such as the data standards, code standards...
and so on in order to provide a basis for the public information exchange. Metadata and the central database should be well maintained in order to provide a safe and reliable hub for the public data exchanging.

b) Data exchanging management: The automatic function of data exchanging and the supplementary function of data exchanging can be provided. Data changing tracking component, data updating component and data synchronous component should be effectively deployed and managed. If necessary, the abnormal data can be restored according to the log of data exchanging.

2) Public data exchanging process

Firstly, data changing tracking component tracks the changing of data from service provider and then generates the data waiting to be updated. Secondly, data updating component according to configurable data rules filters the data and converts it to the standard format data and then updates the corresponding data of the central database based on the updated plans. Finally, data synchronous component according to synchronous scheme synchronizes to the corresponding data of the corresponding application database.

The process of public exchanging is shown in Fig. 4.

The introduction of public data exchanging process is as follows:

a) Service provider: Service provider is an application system that provides a business data (to the data item). Any of the data (to the data item) has a unique data provider.

b) Service requester: Service requester is also an application system that uses a business data (to the data item). Any of the data (to the data item) can have multiple data providers.

c) Data changing tracking component: It is used to intelligently track the changing data of the application databases, generate the data changing packet and then submit it to the central database.

d) Data updating component: This component can analyze, filter and convert the data that waits to be exchanged according to the regular data conversion mode.

e) Data synchronous component: This component can synchronize the data that waits to be exchanged to the corresponding application database.

V . CONCLUSION AND FUTURE WORK

At present, campus information integration is lack of a unified architecture and standard. As the next-generation architecture, SOA is one of the best schema that is used to solve heterogeneous system integration. This paper proposes a new data integration middleware layer based on SOA and describes main components in details. It adopts schema based on XML to establish global model and local model and implements the deep integration of relational database. But some specific details need to be further improved, future work will mainly be improved in the following areas:

1) The type of data source needs to be increased. This paper discusses only integration query of relational database, and now, especially in Network, a lot of information is based on XML or other text format that will be the focus of future data integration research.

2) In practice, most of databases are distributed more scattered, network situation is more complicated, one of databases may be no longer respond, or result set is lost during transmission via network. Considering the complexity of network situation, query between heterogeneous databases should be tested and optimized in a distributed environment.

3) The accuracy and completeness of result set merging should be tested. The accuracy of a single data query is easy to implement, but the complete verification for result set merging of many databases has not yet to find a suitable way.

In short, data integration based on SOA is a very complex problem referring to a wide range of knowledge domains, many issues need to be overcome and be improved in the future of study and research.

REFERENCES