An EAI Integration Framework Based on J2EE

Zhongxian Bao, Pengshou Xie, Aihong Zhu, Hong Zhang
College of Computer & Communication, Lanzhou University of Technology, Lanzhou 730050, P.R.China

Abstract—J2EE technologies to provide a range of middleware services to simplify a number of complex questions related to enterprise solutions development, deployment and management. J2EE Connector system can reduce the range of integration and simplify application development, so it gets easier to implement EIS integration by means of tools. Combining The J2EE platform with J2EE Connector architecture, an enterprise integration framework is built in the this paper, which includes client layer ,middle layer and enterprise information system layer . in the framework, business logic is separated from the client side and server side. As a result, the framework can easy meet the increasingly complex and changing requirement of business logic , and this brings the convenience for development , maintenance and use.

Index Terms—EAI Integration Framework, J2EE Connector architecture, EJB Component, Distributed System Architecture

I. INTRODUCTION

Enterprise Application Integration(EAI) is a kind of technology, by means of interconnection of the existing enterprise applications and data source integration, it implements the information exchange, data share and business collaboration[2]. Using EAI, the business processes, standards, software and hardwires in two or more enterprises will be integrated seamlessly, so these enterprises look like a whole[3]. the technique and content of EAI have different classifications methods from different angles[4], for example, there are data oriented integration and process-oriented integration according to the objects of application, and there are platform integration, data integration, Component integration, application integration and business process integration according to the using tools and technique.

Traditional enterprise management mode is function oriented, communication of departments in a enterprise is closed and mechanical. For the purpose of improving enterprise organization form and management mode from the concepts and structure, accelerating enterprise development and reducing cost, applications integration technique is introduced to enterprise. From this perspective, EAI can be divided inner-enterprise integration and inter-enterprise integration[5]. The inner-enterprise application integration starts from management, integrates functions of various departments ,and make the enterprise management complication and networked. The inner-enterprise application integration request the enterprise managementrunning in a higher level. But inter-enterprise integration tries to find a balance point, lays a cooperation foundation for the cooperative enterprises with different requirement.

The comparatively practical integration methods include message broker, application adapter and EIS (Enterprise Information System) connector(include the synchronous adapter and the Asynchronous adapter), 2-layers client-server method and application oriented method etc[6].

J2EE (Java2 Enterprise Edition) platform is one of the dominant platform to support distributed enterprise application, it is a kind of EAI platform based on the open standard. Using the J2EE platform as the enterprise integration services platform, can simplify development of application model which Component based. Thus, J2EE has been widely accepted as preferred platforms for the development of enterprise level server-side solutions.

II. ARCHITECTURE AND SAFETY MECHANISM OF J2EE

A Layered architecture of J2EE

J2EE is a kind of architecture to simplify development, deployment and management of a number of complex enterprise solutions with multilevel. J2EE using component design ideas, provides a component-based approach to implement multi-layer distributed application model, application logic is divided into components according to the functions of various application, and these components distributed in a different machine according to their level. In addition, the J2EE server provides underlying services in the form of a container for every component type. Because you do not have to develop these services yourself, you are free to concentrate on solving the business problem at hand. A container[7] in the J2EE architecture is a standard component execution environment that provides special services for components, it is responsible for loading components and allows the client program call it distant, it also provides component pooling and transaction coordination, data storage access control services etc. The component in the J2EE architecture is an application or a software unit that container supported. The J2EE multilayer architecture is usually divided into client layer, middle layer and database layer[7,8], As shown in Fig1.

The client layer supports a variety of client types, their clients can be a Web Browser using static HTML a dynamic pages that JSP (Java Server Pages) supported, and it can also be relatively independent of the client-side Java applications, such as some of Applet running in the network system.

The middle layer include two containers which are the Web container and the EJB (Enterprise Java Beans) container, it is also include the business logic of
Applications and some related services, for examples of JNDI (Java Naming and Directory Interface) and JMS (Java Message Service) etc.

The EJB container is mainly composed of EJB application components based on J2EE. The EJB component is a server-side component, it allows to develop distributed object-oriented enterprise applications in Java, and to simplify the whole developing process. The business logic codes to solve or to meet some special fields were performed by the developing process. In essence, the entity Bean have the function to encapsulate the memory in form of object for the other components use. In essence, the entity Bean have the function to implement the data access logic.

The interaction between the Web container and the EJB container is realized by means of RMI (Remote Method Invocation/IIOP/Internet Inter-ORB Protocol). In which ORB (Object Request Broker) is a kind of middleware that can locate service object and use it.

The Database layer consists of one or more Database systems, and in which, it may be include stored procedure and data access relative logic model. The middle layer connects the distributed heterogeneous database via JDBC (Java Database Connectivity).

B The Security Mechanism of J2EE Architecture
Access control is the core of J2EE security service, the essence of access control is that the subject access to objects is limited, whether a subject can access the object according to the subject access right [9,10].

J2EE provide a development standard for applications, the application can be no hard-coded security strategy, so applications can be bind with a declarative security strategy and put it into application components assembly sets. The security model using J2EE security strategy can be applied to any operating environment which is compatible with J2EE, and it can be deployed in any application server which is compatible with J2EE. In the J2EE security model, identification and authorization and data integrity and data confidentiality etc mechanisms have been comprehensive applied, and the model has been used in various layers of the architecture to ensure the safety of the whole system. In the J2EE environment component safety is responsibility of the container which the component is in, when the programmers develop the component procedure, the safety code was not or seldom considered. Just for the safety logic and business logic are relatively independent, the enterprise applications in the J2EE environment can get a good flexibility and expansibility. The J2EE security model is role based model, in the model, the resource access right is assign to the roles and each role assigns the right to its users. To check whether the user has access right, the server checks which role does the user belongs to and whether roles have this right. When there are interactions between the client-side container and the EJB application server, the security mechanism provides two ways to realize the interconnection, they are authorization interface and Authentication interface.

J2EE platform provides a simple development environment for programmers, it is expandable and easy to integrate with existing system and Legacy system. At the same time, programmers can flexibly select the servers, the tools and other components according with themselves demand. And then, it provides some system level services such as naming, transaction and security. In these services, transaction is a kind of very important items in J2EE, one of the transactions API (Application Program Interface) provides various interface to meet different user demands.

J2EE container also provides audit functions in order to facilitate the container evaluate the security strategy imposed to itself, it is a security-related event logging behavior to enable the user or the system accountable for their actions.

III. THE EAI INTEGRATION FRAMEWORK BASED ON THE J2EE PLATFORM
A THE ENTERPRISE APPLICATION INTEGRATION FRAMEWORK BASED ON THE J2EE PLATFORM
J2EE platform has its advantages of powerful transplantable capability and good cross platform capability, it has advantages of good maintainability and high security. The relative Independence between different layers in J2EE architecture make it more agility, the component and OOP have been introduced to make
the models easy to reuse, so programming in the J2EE environment can simplify the developing procedure and increase the programming efficiency. J2EE connector architecture can lessen integration area and simplify application development, and EIS integration by means of developing tools becoming easier[11]. So that, combination of J2EE platform and J2EE connector architecture to complete the EAI as shown in Fig. 2, can provide a useful reference for drafting an EAI solution and give an advanced practicable method for constructing an enterprise system in a more rapid and more effective way, and select a good platform for developing enterprise level server-side program in the EAI project.

The client layer downloads static HTML pages or dynamic HTML pages which generated from JSP or Servlet, runs a independent client program or Applet programs oriented network, and accesses Enterprise Beans bypass representation layer.

The middle layer includes two layers, they are the representation logic layer and the business logic layer. The representation logic layer consists of JSP pages, Applets and Servlet which can display HTML pages. It calls Servlet and HTML page corresponding to the JSP page, and pack them with Web component when applications were composed. The business logic layer reflects and deals with business process, business codes to resolve or to satisfy a special field were executed by the Enterprise Beans in business logic layer.

The enterprise information system layer runs enterprise information system software, such as ERP(Enterprise resource planning), PDM(Product Data Management), DBMS(DataBase Management System) and other Legacy information systems[12].

Servlet is a server side program written in Java, it has mainly functions which to browse and to modify data interactively, to generate a dynamic Web content, and it is independent of the protocols and platform. JSP bases on the text, generates a Servlet by means of container, so the content and its display is separated in it. EJB defines a Bean class component and relates to the Web layer via distance call. JDBC is a interface independent of the specific DBMS. It provides the common mechanism that access to SQL database and storage structure, a common bottom application programming interface that support the basic SQL function. J2EE components are easy to implement the migration between various vendor products that the J2EE-compatible container/server.

J2EE Connector architecture defines a series of mechanisms called agreement, that ensure the EIS and the application server and enterprise applications can be integrated easily. At the same time, these mechanisms are between the J2EE application server and EIS, and they have been designed to be scalable, secure, and transaction oriented. The connector architecture also defines a client interface API, allows J2EE application components access different EIS structures.

If EIS vendors want to join the connector system, he must support the connectors agreement which provides the EIS resource adapter can be embedded into the J2EE-compliant application server. J2EE Connector system is a integration technology based on J2EE. So that, application integration project will not only be able to connect to an existing enterprise systems, but also be able to connect to the Web and other applications. The connector system is mainly used to simplify the integration of J2EE components and the EIS integration process.

B the J2EE key technologies in the integration framework

For enterprise level distributed applications, J2EE defines abundance of technical standards, and defines responding development tools and API according to these standards, so it provides a strong support for the development of enterprise-level applications. These technologies include database access, distributed communications, native interfaces etc[13,14,15].

Java Native Interface: Java Native Interface (JNI) has been integrated into the standard J2EE platform, this allows Java code directly interacting with local shared binary libraries of a specific operating system and hardware platforms, so JNI provides a method can integrate traditional application codes written in other languages to Java enterprise applications. If C or C++ is the target platform language, create a JNI-based code conversion is very convenient, because Java platform has provided the code conversion support of these languages. Deficiencies of using JNI is that memory resources and the thread resources in JNI and threading code values in shared libraries are managed in different ways, we must carefully consider the mismatch problems of these resources when we using JNI. Therefore, using JNI as the main method of EAI solutions is suitable for simple enterprise application integration.

EJB technology: EJB is an important part of J2EE, it is a server-side component, which makes use Java language to develop distributed, object-oriented enterprise applications, and can simplify the whole development process. Usually, EJB components cannot run separately, but running in EJB container. EJB container is running environment that the EJB component exists and executes, it manages the EJB components include its security, its
concurrency, its transaction management and other details, so EJB components is free from the constraints of the server relevance, the application system can be developed in an EJB environment, and then be deployed in other environments, and with the change in demand, application systems can be migrated without modification to other more powerful, more complex server. Using EJB component technology can effectively carry out software reuse and improve developer productivity and reduce software costs in development and maintenance, improve software quality and reduce the complexity of systems construction.

Database Access: Whether the traditional enterprise information systems or in the future, databases occupy an important position. Development of distributed system requires database access with good flexibility and scalability. JDBC is an interface that is independence of the specific database management systems. It provides a common mechanism to access SQL database and storage structure, and provides a common bottom application programming interface to support the basic SQL function. It is provides a uniform user interface for different database and provides a variety of database connection. JDBC API provided by J2EE makes a variety of database operations easy and feasible.

Distributed Communication Technology: J2EE framework provides a variety of communication modes for web applications and EJB applications. Java RMI (Java Remote Method Invocation) can implement the distance communication between Java objects. Java IDL (Java Interface Definition Language) can implement the distance object communication that met to CORBA (Common Object Request Broker Architecture) specification. JNDI (Java Naming and Directory Interface) provides a standard naming interface to access remote objects in distributed systems. JMS (Java Message Service) defines a set of specifications for developing messaging middleware applications. Java clients and Java middle layer accessing JMS messaging system implement a complex application as long as of the definition a JMS simple interface without paying attention to the underlying technical details.

IV. CONCLUSION

An open enterprise application integration framework based on the J2EE platform, not only can meet the needs of enterprise application integration, but also can minimize the complexity of enterprise application integration because it introduces Web Services support, it can improve the flexibility of the system too. The implementation of EAI software, not only largely eliminates information discontinuous between different information systems communication so that they can achieve very good seamless integration, but also truly reflects its functions in the enterprise management. From the technology details point of view, there are some shortcomings and deficiencies. For example, if you wanted to improve integration performance, you must develop applications with a large number of robust codes. In the framework of each component must be able to manage the data which in it and a variety of business rules with the correct method, and implement business constraints.

REFERENCES