Study on the Integration Framework and Reliable Information Transmission of Manufacturing Integrated Services Platform

Pengshou Xie
School of Computer and Communications, Lanzhou University of Technology, Lanzhou, China
Email: xiepsh_lut@163.com

Zhiyuan Rui
School of Mechanical and Electronical Engineering, Lanzhou University of Technology, Lanzhou, China
Email: ruizhy@lut.cn

***Abstract*—To achieve integration on heterogeneous application systems, and ensure reliable transmission of information in the manufacturing integrated service platform, some technologies and methods are studied, such as the hierarchical architecture, center architecture model, security mechanism of J2EE, the integrated architecture, asynchronous communication mode, and advantages of reliable transmission of the information oriented middleware. Combine J2EE platform and J2EE connector architecture, an integrated framework is proposed, which includes client layer, middle layer and heterogeneous information system layer. With the information oriented middleware and its information broker, the framework not only can promote business logic which is separated from the client side and server side, meet easily the increasingly complex and ever-changing needs of the business logic, but also can achieve reliable information transmission. As a result, it will make development and maintenance and application of the manufacturing integrated service platform become more convenient, and will provide the integrated heterogeneous application systems with reliable communications services.**

***Index Terms*—integrated services, heterogeneous application integration, distributed system architecture, integrated framework, information oriented middleware, asynchronous communications, reliable information transmission**

I. Introduction

The manufacturing integrated services platform(abbr. MISP) is a services platform based on service oriented architecture, which contains several modules such as the virtual designing and manufacturing centers, the sharing system of advantages of resource for manufacturing, the e-business service system, the self-designed system for industry, the platform of public services, the platform of management and maintenance, etc. The target of MISP is to break through the traditional service model based on information delivery and provide integrated services to manufacturing industries, such as key product innovation, network manufacture, sharing of preponderant human resources and facilities and technology, commodity catalog display, logistics information management. However, the platform involves in a variety of heterogeneous application systems, for instance, the CAD, PDM, ERP, SCM, CRM, collaborative management. These software systems provided by different software use different data format, term names and definitions, even with the same concept. Which leads application system integration is very difficult to achieve.

With business expansion of MISP, a variety of distributed systems need to be interconnected and work collaborative, in order to meet needs of integrated services. At the mean while, the development of information technique and network provide us a strong technical support for the automation of data flows and business operation, and development of electronic business. The various applications can run on different computers, even on cross-system or across networks. How to integrate these applications effectively, maintain a good communication between different applications, and truly display advantages of the cross-platform application integration [1], are key issues to be solved. Therefore, solving the problem of heterogeneous application integration has become one of the important works of the project.

More and more communications between the applications not only require that sender and receiver applications are online simultaneously, but also require that they know the application interface of each other. But, the actual situation is: the applications were not always online; faults of network hardware are inevitable; there are some network information congestions [2] caused by sudden network data flow; some applications need to be processed immediately, some others need not. Therefore, information transmission reliability between applications is getting increasingly outstanding.

J2EE (Java2 Enterprise Edition) [3] platform is one of the dominant platform to support the distributed enterprise application. It is a kind of EAI (Enterprise Application Integration) [4] platform based on the open standard. Using the J2EE platform as the enterprise integration services platform, one can simplify the development of application model which is component based. Therefore, J2EE has been widely accepted as preferred platforms for the development of enterprise level server-side solutions.

doi:10.4304/jcp.8.1.146-154

II. THE INTEGRATION FRAMEWORK OF MISP

A. The Hierarchical Architecture of J2EE

J2EE is a kind of architecture to simplify development and management of a number of complex enterprise solutions with multilevel. J2EE using component design ideas, provides a component-based approach to implement the multilayer distributed application model. The application logic is divided into components according to the functions of various applications, and these components distributed in different machines according to their level. In addition, the J2EE server provides underlying services in the form of a container [5] for every component types. The container 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 calls 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 supported by container. The hierarchical architecture of J2EE is usually divided into client layer, middle layer and database layer [6], as shown in Fig. 1.

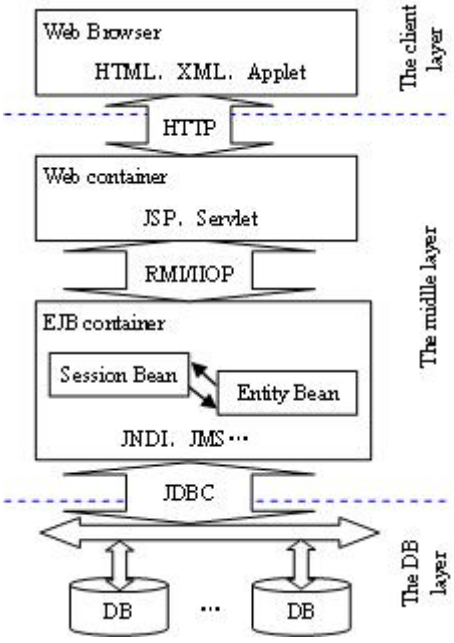

Figure 1. The hierarchical architecture of J2EE

The client layer supports a variety of client types, these clients can be a web browser using static HTML or dynamic pages that JSP (Java Server Pages) supported, and also can be relatively independent of the client-side Java applications, such as some of applet running in network systems.

The middle layer includes two containers which are the web container and the EJB (Enterprise Java Beans) [7] container. It is also include the business logic of applications and some relative services, for example, JNDI (Java Naming and Directory Interface) [8], JMS (Java Message Service) [9], etc.

JSP and Servlet in the web container accept the HTTP requests, from the clients, and response them. Then it sends the treated client information to the Beans of EJB for encapsulation.

The EJB container is mainly composed of EJB application components based on J2EE. The EJB component is a server-side component. It allows us to develop distributed and object-oriented enterprise applications in Java, and to simplify the whole developing process. The business logic code to solve or to meet some special fields is performed by the enterprise beans running in the EJB container.

The session bean is a kind of business process component. It represents business process object, and executes the business logic, arithmetic, rules and the business workflow. Also, it is a reusable component with business process logic. At a given time, session bean is available for only one client and cannot be shared with other clients.

The Entity bean is a kind of data component, and it represents database or data object of enterprise applications. In this bean, the bottom data is mapped to the memory in form of object for other components used. In essence, the entity bean has the function to implementation 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) [10]. 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) [11].

B. The Center Architecture Model of J2EE

Usually the center architecture model of J2EE applications can be classified as "Model-View-Controller"[12], as shown in Fig. 2.

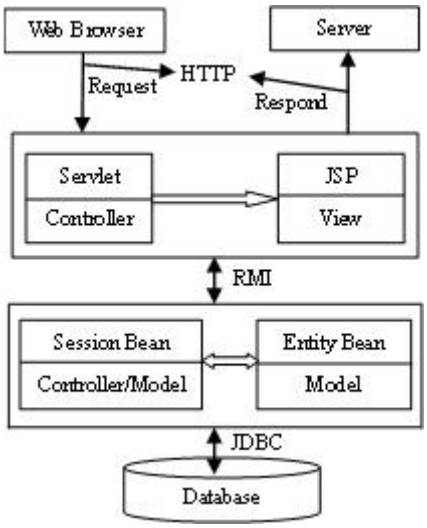

Figure 2. The center architecture model of J2EE

View is a screen, which is presented to the users. It should have application data to present information to the

users. However, the view does not contain the determined data source. Model contains and manages the determined data source from all application objects. So, when the model updates the data, the changes must be notified to the view. In this way, the view can transmits and displays the correct and updated information to the users.

Controller component is used to control user actions processing. The user action on the screen is split with the controller component, and these actions are handled by the application. So, the application design can be more flexible. There has been a demarcation point between the model and view. This is to loosen the coupling of view and model (front-end and back-end), when the controller component handles the user's requests and calls of the function.

Model is the application's business logic components. The data from the model can be used as value objects, XML data, event objects, etc., which is passed to the Web container. "Model-View-Controller" is based on event-driven. In this model, users can browse the data and receives the updated data simultaneously. Or, when data is changed, the view or the controller will need to return to the backend model to obtain new data, and to respond by the view in the ultimate. Therefore, the interaction with the user includes the view and controller components.

C. The Security Mechanism of J2EE Architecture

Access control is the core of J2EE security service, and the essence of access control is that the subject access to objects is limited. Whether a subject can access the object is according to the subject access right [13].

J2EE provide a development standard for applications. The application can be no hard-coded security strategy, and 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, some mechanisms have been comprehensive applied, which are identification, authorization, data integrity and data confidentiality, etc. 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 has the responsibility of the container in which is the component. When 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 the role based model [14]. In the model, the resource access right is assign to the roles and each role assign the right to its users. To check whether the user has the access right, the server checks which role 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, which are authorization interface and authentication interface.

J2EE platform provides a simple development environment for programmers; it is expansible 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, and one of the transactions API (Application Program Interface) provides various interfaces to meet different user demands.

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

D. The Integration Framework of MISP

J2EE platform has its advantages of powerful transplantable capability and good cross platform capability, and has advantages of good maintainability and high security. The relative independence between different layers in J2EE architecture makes 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 [15] can lessen integration area and simplify application development, and EIS (Enterprise Information System) integration by means of developing tools becoming easier [16]. So that, combination of J2EE platform and J2EE connector architecture to complete the EAI (as shown in Fig. 3), 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 project.

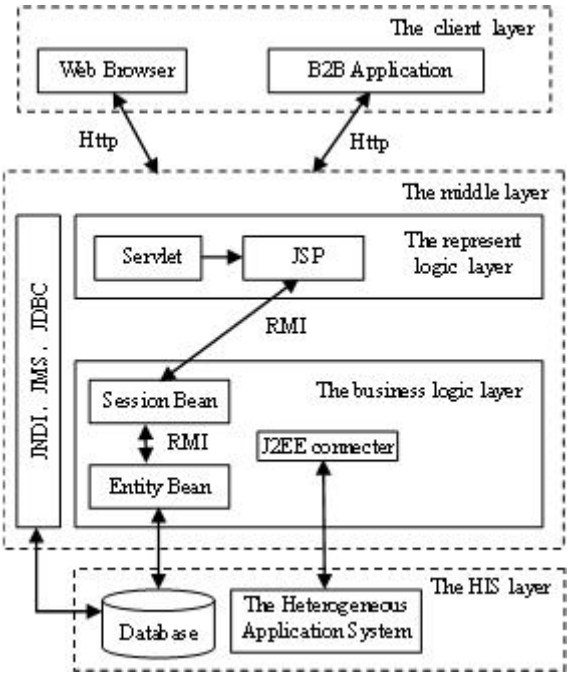

Figure 3. The integration framework of MISP based on J2EE

The client layer downloads statics HTML pages or dynamic HTML pages which generated from JSP or servlet; runs an independent client program or applet programs oriented network; and accesses Enterprise Beans bypass representation layer.

The middle layer includes two layers, which are the representation logic layer and the business logic layer. The representation logic layer consists of web page, applet 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 heterogeneous information system layer runs enterprise information system software, such as ERP, PDM (Product Data Management), CAD, DBMS (Database Management System) and other Legacy information systems [17].

Servlet is a server side program written in Java. It has mainly functions which to browse and to modify data interactively, to generate dynamic web content, and it is independent of the protocols and platform. JSP bases on the text, generating 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 an interface independent of the specific DBMS. It provides a common mechanism and a common bottom application programming interface, the mechanism access to SQL database and storage structure, the interface 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, which ensure the EIS, 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, and allows J2EE application components access different EIS structures.

If EIS vendors want to join the connector system, he must support the connector's agreement which provides the EIS resource adapter that can be embedded into the J2EE-compliant application server. J2EE connector system is an 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.

E. Some Key Technologies of 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 [18], distributed communications [19], native interfaces [20], etc.

1) Java Native Interface

Java Native Interface (abbr. JNI) has been integrated into the standard J2EE platform, which 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 are memory resources and the thread resources in JNI.Threading code values in shared libraries are managed in different ways, so 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.

2) EJB Technology

EJB is an important part of J2EE. It is a server-side component, which 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 a running environment that the EJB component exists and executes, and it manages the EJB components including its security, its concurrency, its transaction management and other details. EJB components are free from the constraints of the server relevance, so the application system can be developed in an EJB environment, and be deployed in other environments. 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, improve developer productivity, reduce software costs in development and maintenance, improve software quality and reduce the complexity of systems construction.

3) Database Access

Both in the traditional or future enterprise information systems, database occupy an important position. Development of distributed systems 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.

4) Distributed Communication Technology

J2EE framework provides a variety of communication modes for web applications and EJB applications. Java RMI 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 provides a standard naming interface to access remote objects in distributed systems.

JMS 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 the definition of a JMS simple interface without paying attentions to the underlying technical details.

III. Reliable Information Transmission of MISP

A. Information Oriented Middleware

Synchronous communication [21] is usually used in a communications environment that needs to respond and confirm between sender and receiver. Because each respond to the sender and receiver is required, they are a tightly coupling relationship. Since synchronous communication requires a reliable network environment, it is not suitable the MISP for information transmission.

When use asynchronous communication [22] to achieve the information transmission, the client can achieve trans-information in the case that session connection is not established. In other words, when the information is sent, the client of receiving end does not need to run; the information broker may used as a recipient of the information. If it fails, the information broker will continue to re-send. This asynchronous method can be used in that the network conditions are not very reliable, and the congestion problem of the information can be solved well.

The middleware is a software layer between operating systems (includes the underlying communication protocols) and various distributed application systems [23]. It implements the request of communication between software systems by means of predefined interface and information. The middleware is a support tool to build enterprise application integration. With the middleware, the interconnection and sharing issues of multiple heterogeneous data sources in the network distributed environments are solved, and collaborative work between various applications is implemented, as shown in Fig. 4.

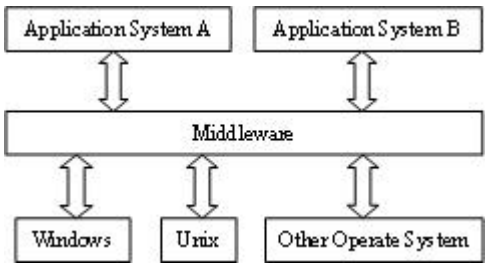

Figure 4. The place of middleware in application system

The basic objectives of the middleware is to help programmers can develop some application systems, such as the distributed Client/Server applications, end-and-end applications, even distributed object oriented applications, without deeply understanding of the communication protocols, accessing method of heterogeneous database and variety of operating environment.

Information Oriented Middleware (abbr. IOM) is a kind of middleware which supply information transmission service for applications. Using effective and reliable information transmission mechanism, IOM passes information cross the different hardware networks, different operating systems and applications which worked on the various network protocols. Based on the digital communication technology, IOM integrates distributed applications (as shown in Fig. 5). Just because these functions, IOM plays a very important role in MISP.

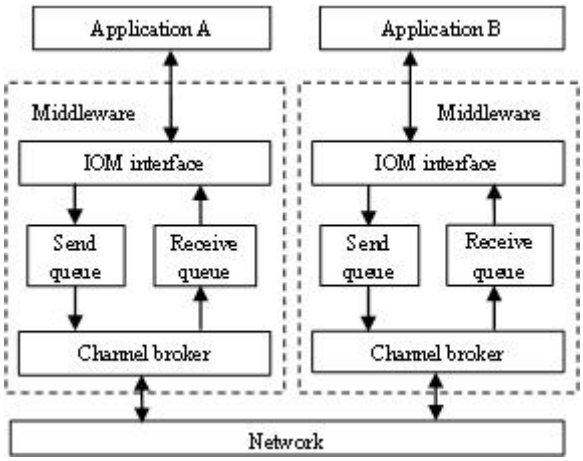

Figure 5. General frame of IOM

There are many advantages of IOM. It can connect the server and client in synchronization model or asynchronous models, and it can pass or store-forward information anytime. For above reasons, it is better than remote procedure call. In addition, IOM does not occupy substantive network bandwidth and tracks transactions. By means of storing transactions in disk, it can recover system after fault occurs. So, IOM is suitable for the distributed environments which require reliable data transmission crossing many processes.

In the information transmission mechanism, there are two important concepts, one is information, and another is queue.

The information is needed to be passed between the sender and receiver, and it can be various forms media, for instance text, sound, image, and so on. The content and format of information usually defined by the mutual negotiation that both in the sender and the receiver.

The general format of information is composed of information header, effective payload of information, and information behavior, as shows in Fig. 6.

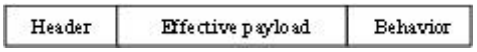

Figure 6. The general format of information

The information header includes the source and destination of the information. The effective load means actual data witch included in the information, for examples, refresh database, query database, and insert a table, etc. In the header, the receiver address, information length, information type, the blocking level of information transmission, and the other control information of the infor-

mation, for examples, the lifecycle of information and the value of time out, are included.

The queue is a sharable storage area for both the sender and the receiver. It may be in the memory or in a physical file. The information can be sent in two modes: express mode and recoverable mode. In the express mode, to realize the fast information transmission, the information is saved in the memory rather than physical disk, to catch a high-speed processing ability; but in the recoverable mode, to get a better fault recovery ability, at each step in the passing procedure, the information is written into the physical disc.

1) Information Communication Model

In IOM, asynchronous communication model was be used. The information sender only sends information with pay no mind to waiting receiver's answer, at the same time, the receiver needless to deal with the sender requirement at the IOM of information is received. The logic relation and the structure of information are defined by the program itself. In IOM the asynchronous mode deals with the information in two methods: information queue model and publish-subscriber model [24].

2) Information Queue Model

In the information queue model, many senders save his information in one queue, and many receivers fetch his information from the same one queue. The queue is deal with in continuous, and usually FIFO (First-In First-Out) rule is followed [25]. The information can be fetched from the queue by only one receiver, once the information is fetched, the other receivers cannot get it. Fig.7 is an example of information queue model. In this example, an application system sends information to IOM server in information queue model. I1，I2，I3，I4 are some information, and they follow the FIFO rules. The receiver applications only fetch the information at one time. For instance, I1 is first out, and is fetched by application E, and then I2 is fetched by application D. The information cannot be fetched by two or more application systems.

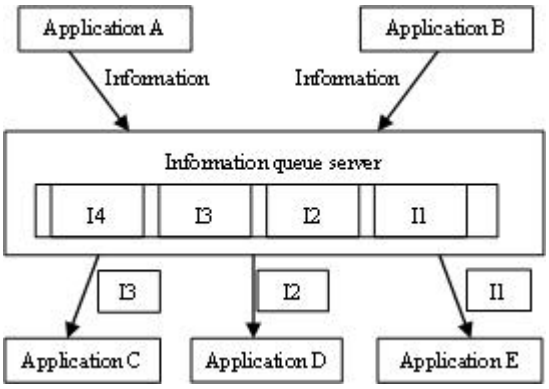

Figure 7. Information queue example

3) Publish/Subscriber Model

In Publish/Subscriber model, information is usually generated as an event result, so this model is also called event-driven information model. In this model, the publisher is used to generate information, and the subscriber subscribes the information that they interested, receives the information he subscribed to, and triggered by some certain events.

Basic factors of Pub/Sub model include publisher, subscriber, information title, and passing mechanism. Pub/Sub model permits many subscriber fetch the same information. Fig.8 shows the information model of Pub/Sub.

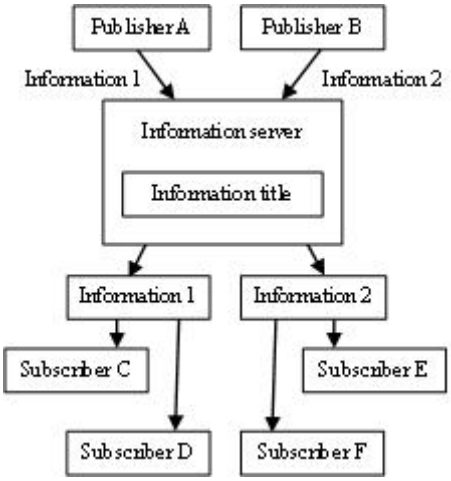

Figure 8. Pub/Sub information model

B. Reliability of Information Transmission

As a part of IOM, functions of information broker middleware (basic structure shows as Fig. 9) become increasingly obvious in MISP. For the reason of the distributed business units scattered, various applications run on the different software and hardware platform, in distributed environment.

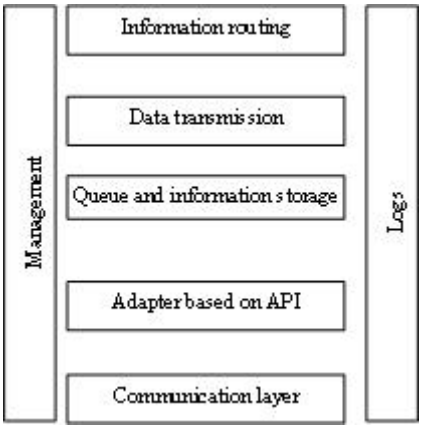

Figure 9. Basic structure of information broker

Information broker middleware mainly supply the necessity functions of applications integrated, such as data passing, collection, translation, filtering, mapping, and routing, etc. It shield difference of hardware, databases, information formats and communication protocols, ensure high effective and convenient communication from one application to others.

The steps of information transmission and dealing with information in persistence and reliable shows as follows: (as Fig. 10 shows).

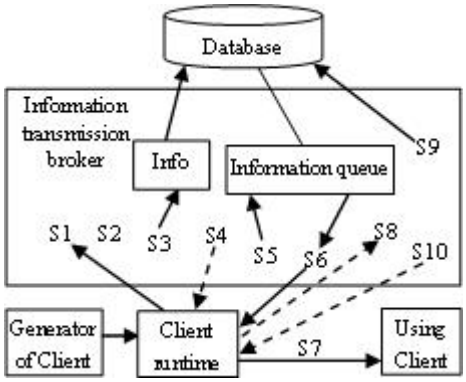

Figure 10. Information transmission step

In Fig.10, the continuous line and arrow represents effectively loaded information, the dash line and arrow represents control information.

1) Information Generation

Step1: The client runtime sends the information from generator to the broker through the connection.

In the case that passing mode was set as persistence mode (an assuring transmission, means to transmit information and transmit it only one time, even if the broker is wrong), the broker sends the control information back to the client runtime as default. The broker validates that the information has been sent to the target, and saves information in the broker's database. The send () function of generator is prevented until the validation is received. This validation is transparent for the target client of persistence information.

In the case that passing mode was set as no persistence mode. The broker does not send the control information back to the client runtime as default. But, if one knows whether broker received the no persistence information, the sender can enable the broker validation. In fact, the broker validation must be enabled for the broker to slowdown the information generating speed, if the targets memory bound is reached.

2) Information Processing and Routing

Step2: Broker reads information from the connection and sends it to the targets.

Step3: Broker saves the persistence information into data store.

This step assures that broker will not lose the persistence information before it was received by the targets. It means that the broker has to save the information in persistence data store, when the information arrived at its physical target. If the broker is wrong for some reasons, the information can be recovery and sent to the subscriber.

Step4: The broker sends a validation to client runtime of generator.

Step5: The broker determines the information routing.

Step6: Broker writes the information to the appropriate connection, and marks the information using the subscriber's unique identifier.

3) Using Information

Step7: The client runtime of subscriber pass the information from the connection to the subscriber.

Step8: The client runtime of the subscriber sends information used validation to the broker.

The broker is waiting for the validation from the clients, once gets it, deleting the information from targets. There are some different regulation validation modes respectively represented different grades of reliability. The clients' validations can apply to individual information, information groups and transactions.

The transaction is a method which is a working unit combines information generation and/or use of one or more information. In this case, the validation from the fore mentioned client runtime server and the broker run on the transaction level by default. The broker validation is sent automatically, as soon as the transaction is submitted.

When the information is generating or using in a transaction, each sending and receiving process is tracked and completes them until the client sent a submission request. If failure of the given sending or receiving happens, exception handling will be toggled. The client may ignore it, retry it and roll it back, to deal with the abnormity. All operations are finished as soon as transaction is submitted successfully, and all operations are canceled if there is a rolling back.

4) Terminate Lifecycle of Information

Step9: The broker answers the client validation, and deletes the information in the targets and data store database, after it can confirm that all the validations have been received.

If there is something wrongs with the brokers or the connection, the broker perhaps cannot get the client validations, and retransfer all information which was transferred before but didn't received the client validations, and mark it with a retransfer tag.

When the information cannot send out, the system will give up it or put it into the dead information queue. This method is adopted according to the actionable prevented factors.

The information was given up in case of can't be sent out successfully and not in use:

a. Purge one or more information in targets using management tools.

b. Had been deleted or redefined a long-term subscription, so it caused the information can't be sent out.

However, in the following situations, the information will be treating as dead information and put it into the dead information queue according to its behaviors configured:

a. Information timeout, that's means the passing section takes longer time than timeout value setting in the information title.

b. The information consumed the memory more than the threshold; a delete restricted behavior has been called.

c. The client abnormity lead the failed passing information that cannot be used.

It can hold this kind of information and put them into dead information queue. At the same time, it set the special information queue attribute value to sign when and why put them here.

Step10: Information server sent a validation to the client runtime that the committed validation was treated, and needless to send it again.

IV. CONCLUSION

An open application integration framework based on the J2EE platform, not only can meet the needs of heterogeneous application integration, but also can minimize the complexity of enterprise application integration. Because that it introduces web services support, it can also improve the flexibility of the system. Through the implementation of integration, MISP not only largely eliminates information discontinuous between different information systems communication, such that they can achieve very good seamless integration, but also truly reflects its functions in the enterprise management. In the framework, each component must be able to manage the data and a variety of business rules with the correct method, and implement business constraints.

It has a better effect to use of heterogeneous application integration solutions based on IOM. Because it can pass information without connect the sender and receiver synchronously, and has very strong transmission protection by request and answer, even the communication of sender and receiver was happened in different times, and can provide defense communications and logs. IOM guarantee the information transmission without lost by means of powerful technique. IOM has a log files, checking and error recovery. In conclusion, functions of IOM were mainly reflects on two aspects: one is application integration, the other is to insure reliable communication cross applications. IOM provides reliable, asynchronies and loose coupling communication service to guarantee data passing, and provides a reliable information service for distributed applications.

ACKNOWLEDGMENT

This work is supported by the National Science and Technology Support Program, China (2006BAF01A21).

REFERENCES

[1] Serhani, Mohamed Adel; Benharref, Abdelghani. Enforcing Quality of Service within Web Services communities. *Journal of Software*, vol.6, no.4, pp.554-563, 2011.

[2] M. Kimura, K. Yasui, T. Nakagawa, N. Ishii. Reliability consideration of a mobile communication system with network congestion. *Computers & Mathematics with Applications*, vol.61, no.2, pp.377-386, 2006.

[3] Takoua Abdellatif, Jakub Kornas, PJean-Bernard Stefani. Reengineering J2EE servers for automated management in distributed environments. *IEEE Distributed Systems Online*, vol.8, no.11, pp.1, 2007.

[4] Muhammad Mustafa Kamal. The case of EAI facilitating knowledge management integration in local government domain. *International Journal of Information Management*, vol.31, pp.294-300, 2011.

[5] Zhang Xiaoshuan, Chen Peijun, Zhao Ming. Data expression methods in enterprise application using J2EE architecture. *WSEAS Transactions on Communications*, vol.7, no.6, pp.541-550, 2008.

[6] Hu, Min; Pan, Ding-Ding; Zhou, Pei-En. Research and application of J2EE and AJAX technologies in industry report. *Journal of Computers*, vol.6, no.9, pp.1847-1851, 2011.

[7] Matthias Merz. Enabling declarative security through the use of Java Data Objects. *Science of Computer Programming*, vol.70, no.2, pp.208-220, 2008.

[8] PChris Richardson. Untangling Enterprise Java. *Component Technologies*, vol.4, no.5, pp.36-44, 2006.

[9] G. Costa, F. Martinelli, P. Mori, C. Schaefer, T. Walter. Runtime monitoring for next generation Java ME platform. *Computers & Security*, vol.29, no.1, pp.74-87, 2010.

[10] MB Juric, I. Rozman, B. Brumen, M. Colnaric, and M. Hericko. Comparison of performance of Web services, WS-Security, RMI, and RMI–SSL. *Journal of Systems and Software*, vol.79, no.5, pp.689-700, 2006.

[11] Marco Crasso, Alejandro Zunino, Leonardo Moreno, Marcelo Campo. JEETuningExpert: A software assistant for improving Java Enterprise Edition application performance. *Expert Systems with Applications*, vol.36, no.9, pp. 11718-11729, 2009.

[12] Yanfang Wang, Chunyan Guo, Lei Song. Architecture of E-Commerce Systems Based on J2EE and MVC Pattern. *In the Proceedings of the ICMECG'09*, pp.284-287, 2009.

[13] Islam, Shareeful; Houmb, Siv Hilde. Towards a framework for offshore outsource software development risk management model. *Journal of Software*, vol.6, no.1, pp.38-47, 2011.

[14] Yu, Su; Brewster, Jon J. Formal specification and impementation of RBAC model with SOD. *Journal of Software*, vol.7, no.4, pp.870-877, 2012.

[15] Li, Xianjun; Ye, Gang; Li, Zhongwen; Ma, Shilong. Study and implementation of spacecraft integration test platform based on component technology. *Journal of Computers*, vol.6, no.5, pp.963-968, 2011.

[16] D. I. Gusev. Cost estimating technique for information technology in enterprise information system. *Automation and Remote Control*, vol.71, no.9, pp.1964-1968, 2010.

[17] Michael Mortensen, Sudipto Ghosh, James M. Bieman. A test driven approach for aspectualizing legacy software using mock systems. *Information and Software Technology*, vol.50, no.7, pp.621-640, 2008.

[18] Askar S. Boranbayev. Defining methodologies for developing J2EE web-based information systems. *Nonlinear Analysis: Theory, Methods & Applications*, vol.71, no.12, pp.e1633-e1637, 2009.

[19] Songwen Pei, Baifeng Wu, Kun Zhu, Qiang Yu. Novel Software Automated Testing System Based on J2EE. *Tsinghua Science & Technology*, vol.12, pp.51-56, 2007.

[20] A. Sanchez, MS. Perez, K. Karasavvas, P. Herrero, A. Perez. MAPFS-DAI, an extension of OGSA-DAI based on a parallel file system. *Future Generation Computer Systems*, vol.23, no.1, pp.138-145, 2007.

[21] Stefan Hrastinski. The potential of synchronous communication to enhance participation in online discussions: A case study of two e-learning courses. *Information and Management*, vol.45, no.7, pp.499-506, 2008.

[22] Filip Maly, Antonin Slaby. Web client server systems with advanced asynchronous communication, their architecture and applications. *WSEAS Transactions on Computers*, vol.9, no.7, pp.716-725, 2010.

[23] Wolfgang Emmerich, Mikio Aoyama, PJoe Sventek. The impact of research on middleware technology. *SIGOPS Operating Systems Review*, vol.41, no.1, pp.89-112, 2007.

[24] S. Oh, JH. Kim, G. Fox. Real-time performance analysis for publish/subscribe systems. *Future Generation Computer Systems*, vol.26, no.3, pp.318-323, 2010.

[25] Wei Xiong, T. Altiok. Queueing analysis of a server node in transaction processing middleware systems. *Computers & Operations Research*, vol.35, no.8, pp.2561-2578, 2008.

Pengshou Xie was born in Tianshui city, Gansu province, China, in Jan. 1972. He received B.E. degree in the mechanical design and manufacturing from Gansu University of Technology in 1995, and M.E. degree in mechanical design and theory from Lanzhou University of Technology in 2011. His major research fields include manufacturing informatization, parallel and distributed software, computer aided technologies, software engineering and integration technology.

He engaged in education and scientific research at Gansu University of Technology since 1995. Up to now, he has accomplished 29 scientific research projects in charge or in participation, and received 3 awards. He published 30 research papers as the first author, among which 10 were included in EI or ISTP.

Associate professor Xie is currently the director of the Information System Integration and Application Research Institute in Lanzhou University of Technology. He also is the director of the CAD research centre, and serves concurrently as senior member of China Computer Federation.

Zhiyuan Rui was born in Baiyin city, Gansu province, China, in Jan. 1962. He is vice-president, professor and supervisor of doctoral students of Lanzhou University of Technology. His major research fields include the digital manufacturing and automation equipment, the computer simulation and virtual technology, mechanical strength theory.

He engaged in education and scientific research at Gansu University of Technology in 1983. He has accomplished more than 60 scientific research projects in charge or in participation, owned 11 national patents, and received 5 awards. He has published more than 100 research papers, of which more than 20 have been indexed in the SCI or EI.