Research on Enterprise Application Integration Platform Based on SOA Architecture

Liu Pingping
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1341369601@qq.com

Lu Jiaxing
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1721653661@qq.com

***Abstract*—Tobacco industry is a relatively early industry in China' information industry, so there are many problems, such as the lack of overall planning, the wide application system, but the low degree of integration of information resources, and the serious problem of "information island". In order to solve how to establish an efficient and flexible way of information interaction within the enterprise, an enterprise application integration platform based on SOA architecture is proposed. The platform takes ESB as the core, transforms enterprise information integration into a new way conforming to SOA architecture, and establishes the idea of basic data management, so as to achieve the purpose of optimizing the overall information resources of the enterprise**

Keywords-SOA Framework; Information Interaction; ESB; Information Integration

I. INTRODUCTION

SOA is widely used in the IT industry in the 21st century. SOA is service oriented architecture, Service oriented architecture. It is architecture, not a technology or a method. We can also say that SOA is an idea. In China, many enterprises begin to build enterprise integration platform based on SOA. For example, Kingdee Apusic SOA. Fmqm exotica, developed by Almaden Laboratory of IBM company, is a distributed workflow management system based on persistent message queue, which can save the execution information of workflow through persistent message queue and complete the complete independence of all nodes in the execution process

SOA architecture has three significant advantages: loose coupling, coarse granularity, and location protocol transparency. Through the encapsulation of services to achieve a comprehensive loose coupling, loose coupling can reduce the dependency between services, so that the flexibility of the service itself can be improved, and it will not be forced to adjust because of other services adjustment, thus greatly improving the reusability of services. Coarse granularity means that the interface of services defined in SOA is close to the actual user operation. Location protocol transparency means that when accessing the service defined by SOA, you do not need to know the specific location and transport protocol of the service. Even if the location and transport protocol of the service change, the client that invokes the service does not need to change. Based on the investigation and analysis of the problems of information island, high coupling degree and poor integration expansibility of each system in BJ cigarette factory based on SOA architecture, an enterprise application integration platform design scheme suitable for the actual situation of the enterprise is proposed to solve the current problems.

Therefore, through the research on the practical application of the enterprise application integration platform based on SOA architecture in cigarette enterprises, through the research on SOA architecture and ESB technology, combined with the analysis of the actual problems of information integration in cigarette manufacturing enterprises, this paper puts forward the design scheme of the enterprise application integration platform that adapts to the actual situation of enterprises.

II. THE CURRENT SITUATION

Tobacco industry is an industry with an early start of information construction in China. At present, the level of information is generally high. BJ cigarette factory, as the main cigarette manufacturing enterprise in Shaanxi Province, has many application systems after years of information construction, covering all aspects of the factory from production to management. The main information systems include manufacturing execution (MES) system and enterprise Industrial

resource planning (ERP) system, logistics system, data acquisition system in the car room, centralized control system in the silk making workshop, power and energy management system, human resource management system, enterprise card system, etc.

The more application systems there are, the problem is not only the disunity of basic data, but also the complexity of system integration. The traditional integration method generally adopts the point-to-point mode. Each system needs a special channel to integrate Chengdu. As shown in Figure 1, the integration of N application systems will generate N * (n-1) integration channels with high complexity. When new application systems need to be integrated, the complexity improvement will also be exponential.

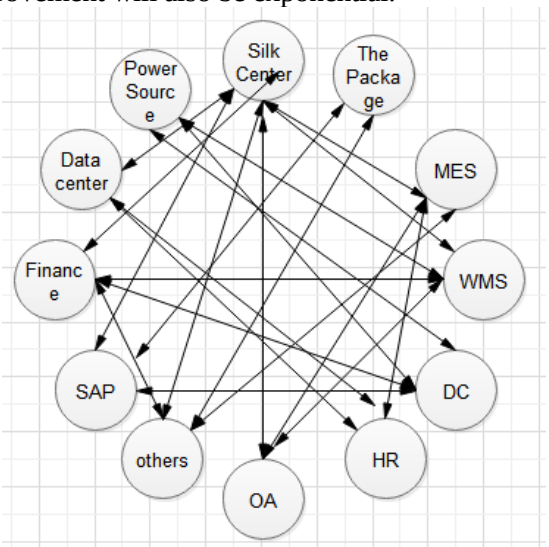

Figure 1. Integration complexity

In summary, the current informatization problems of BJ cigarette factory mainly include the following aspects:

1) There are some isolated information islands and some application systems are in information closed state due to lack of external integration means.

2) The basic data in the enterprise is scattered in different application systems, which need to be maintained separately, so it is difficult to ensure the unity of the basic data of the whole enterprise, so as to "count out one place". The lack of a unified basic data code system makes information interaction difficult.

3) Basic data depends on business system and has high coupling. At present, the basic data in the enterprise mainly depends on ERP system, and the purpose of the basic data is to provide the most basic data information for all application systems of the whole enterprise, so relying on a single application system will cause unnecessary impact on the users of other basic data.

4) Poor integration scalability. At present, the information system of the whole enterprise adopts the point-to-point integration mode. If the new application system wants to join the integration system, it needs the cooperation of each application system. The upgrading or transformation of the existing application system also needs to involve a lot of external interface changes, because of this poor scalability.

5) The lack of management and monitoring of data interaction process makes it difficult to find and deal with problems in the process of data interaction in time. Some data have high requirements for timeliness. If it can not be communicated in time, it will have a significant impact on the actual business. Therefore, effective management and monitoring measures are needed for the data interaction process.

6) The integration of point-to-point results in the aggravation of network burden, and many data interaction contents are repeated, but the data can not be reused, resulting in the waste of resources.

Through the analysis and optimization of the current problems of the enterprise, the core of which is to establish a reasonable and efficient way of information integration. In recent years, with the continuous development of information integration technology and the formulation of a series of standards and specifications, a new solution is gradually being paid attention to, which is based on service oriented architecture the enterprise application integration (EAI) of Architecture (SOA) regards each application system in the enterprise as the service unit of SOA architecture, and establishes the enterprise application integration platform to realize the information integration between each application system. In this enterprise application integration platform, an enterprise service bus (ESB) is needed to provide standardized services. Enterprise service bus is the service operation support platform in SOA architecture, and the services encapsulated by other application systems run on this service bus, as shown in Figure 2, its establishment can effectively optimize the current enterprise ' s disordered and meshed integration mode. Secondly, we need to establish a data exchange management platform to manage all services running in the enterprise service bus and monitor the data interaction process in the integration. Finally, we need to establish a basic data management platform, as a service provider in the SOA architecture, to provide basic data management functions for other application systems. The basic data

management platform will integrate the basic data of other application systems and manage them uniformly. Other systems do not need to be managed separately.

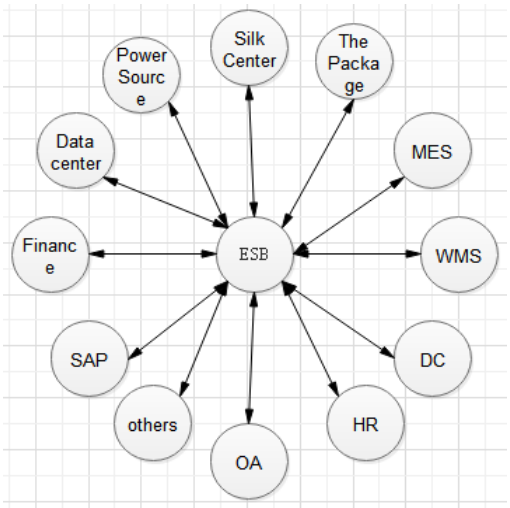

Figure 2. Schematic diagram of optimized enterprise integration channel

III. Design and Implementation

Because IBM WMB is used as the enterprise service bus, IBM DB2 will be used as the database and IBM was (WebSphere Application Server) will be used as the application server for better overall stability.

According to the demand analysis, the data exchange platform as the enterprise service bus will provide a unified entry service WS? MB. After other application systems call the service, the ESB will parse and route the called messages, and find the corresponding registered business processing WebService to call.

The data exchange management platform is responsible for the management and monitoring of IBM WMB enterprise service bus. In the data exchange management platform, it is necessary to register, modify, disable, reuse and other management functions for the service processing business. At the same time, the data exchange management platform should also realize the log recording of data sending, so as to complete the monitoring of data exchange process.

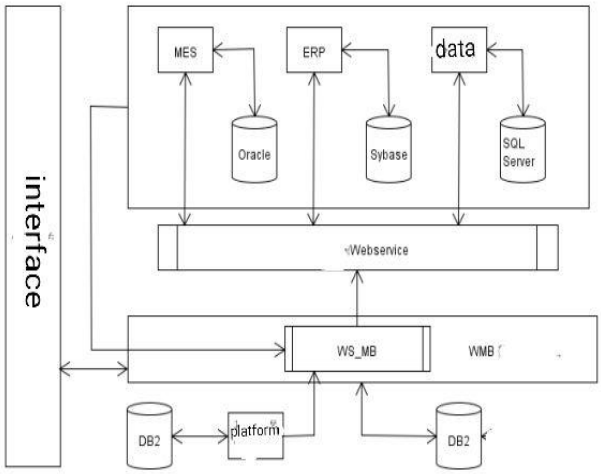

Figure 3. Technical framework of the platform

IV. Design of Enterprise Service Bus Date Exchange Platform

As the core module of enterprise application integration platform, data exchange platform needs to undertake the important work of message transmission. As shown in Figure 4, it is a basic data exchange process. The data exchange platform publishes a unified entry service WS? MB through the web service. The service caller calls the service first, and sends the call request to the data exchange platform in the form of XML message. The data exchange platform will analyze the message content, find the actual service to call, and send the message to the actual, the actual service provider will return the processed results to the data exchange platform in the form of XML message, and then the data exchange platform will return to the original caller.

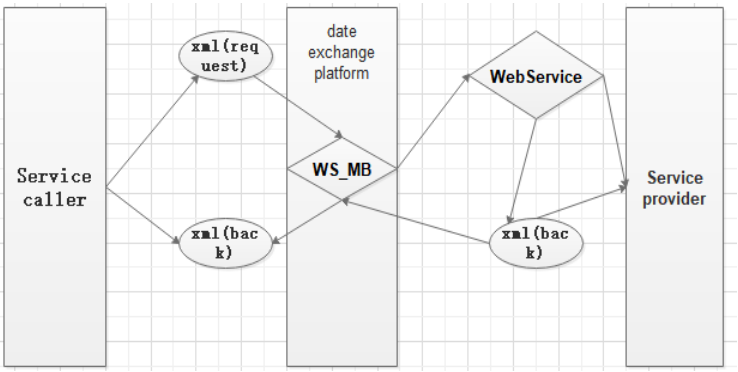

Figure 4. Flow chart of data exchange

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

A. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable the request message sent by the service caller should contain complete routing information, so how to define the routing information? In fact, from the process of data exchange, it can be seen that the three elements of data sender, data receiver and the service to be called can constitute a unique data process, so the routing information should also contain three elements.

Unified format definition of service call request XML message:

<?xml version="1.0" encoding="GB2312"?>

<Msg>

<Head>

< ID > < ID > / / message ID or serial number

< name > < name > / / message description

< source > < source > / / data source

< target > < target > / / data destination

< sername > < / sername > / / call service ID

< msgtype > < / msgtype > / / type of message (0: normal 1: request 2: answer)

< rtcode > < / rtcode > / / the return value of the corresponding request (1: success 0: failure)

< rtdesc > < / rtdesc > / / return description of the corresponding request

< backup1 > < backup1 > / / standby information

<Backup2></Backup2>

<Backup3></Backup3>

< date > XXXX / XX / XX XX: XX: XX < / date > / / message sending time

</Head>

<DATA>

<table tablename = 'table name 1' fieldname1 = 'column 1 description' fieldname2 = 'column 2 description' fieldname3 = 'column 3 description'... >

<row action = insert 'id = ' primary key value 'fieldname1 = ' AAA ' fieldname2 = ' BBB '... >

<table tablename = ' table name 2' fieldname1 = ' column 1 description' fieldname2 = ' column 2 description' fieldname3 = ' column 3 description'... >

<ROW ACTION = 'INSERT' ID = '' FIELDNAME1 = 'AAA' FIELDNAME2='BBB' .../>

</ROW>

</TABLE>

</ROW>

< / Table > / / the main body of the data sent. Table represents a data table. Row is the specific data. Table can be nested in table to represent the data of the main sub table structure.

</DATA>

</Msg>

In the XML definition, the head part describes the basic information of the data, and the three attributes of source, target and sername are the most important routing information. Through these three attributes, you can uniquely determine the service that the data needs to call, that is, the user of the service, the provider of the service and the name of the service. These attributes are registered by the service management module and are in the unified portal after receiving the XML data, the service calls the corresponding service according to the three attributes and the service registration information in the management module to send the data.

V. IMPLEMENTATION

The main functions of data exchange management platform are service management and data exchange process monitoring. As shown in Figure 5, it is the main interface of the data exchange management platform. The frequency of data exchange can be calculated through the log, and the reception and transmission volume of each system and data accessing the platform can be displayed intuitively.

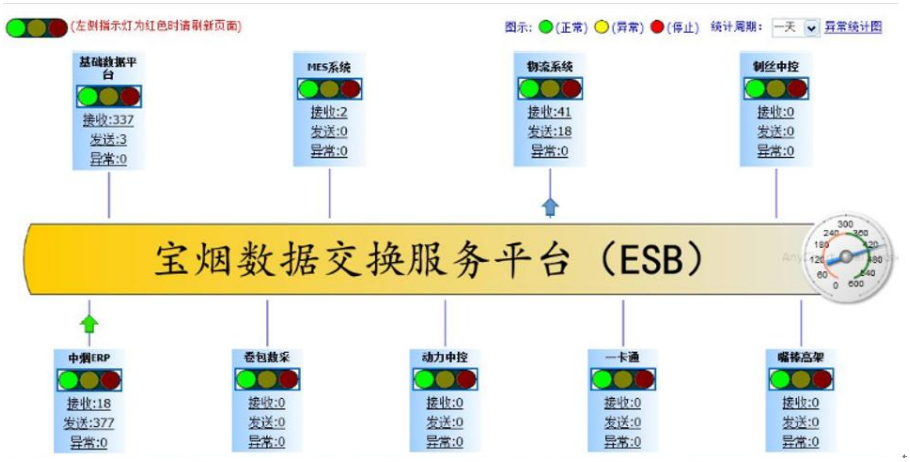

Figure 5. Main interface of data exchange management platform

a) The function of service governance is realized by registering and managing the web service published by each application system. As shown in Figure 6, the contents to be registered include: sequence number, system name, interface name, enabling tag, source, target, interface service name and WebServiceURL, namespace, calling method input object, input parameter name, output parameter name, calling method output object, extended input parameter, extended input parameter value, authentication information, WebService technology, remarks, etc. (due to confidentiality reasons, the figure is not complete).

b) Figure 7 is the implementation interface of the authority management function of the management platform for basic data. The maintenance of basic data is usually carried out by the personnel in charge of specific business. Different business personnel are usually responsible for different data. The authority management module can configure the addition, modification, deletion and query authority of various basic data according to different roles, and can also configure whether specific attributes are visible The RBAC mode is implemented.

Figure 6. Data operation module

First, configure different roles in role management, then configure the permissions of roles through role and function relationship, and finally configure the roles of different users through role and user relationship or user and role relationship. A user can have multiple roles, and a role can be played by multiple users at the same time.

Figure 8 is the implementation of the data synchronization module of the basic data management platform. By customizing the interface content and configuring different sending interfaces for different systems, you can configure whether the attribute column of the basic data is sent, the name of the sent column, etc., and you can also filter and group the sent content through SQL statements.

Figure 7. Authority management

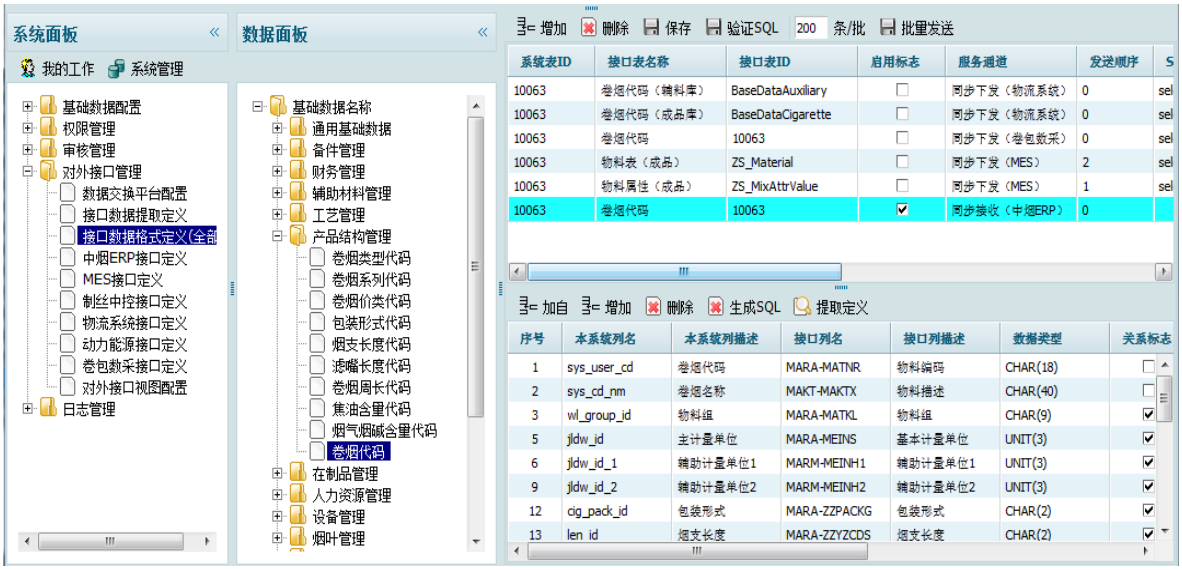

Figure 8. Definition of interface service

Other application systems publish and register the service of receiving basic data in the data exchange management platform. After the basic data management platform configures the interface, when the new basic data maintenance is completed, the platform will automatically send the data to the corresponding system through the data exchange platform according to the interface configuration. All application systems adopt this mode, and the basic data is unified.

VI. CONCLUSION

In this paper, through understanding the application status of SOA architecture in BJ cigarette enterprise, the enterprise integrated information system based on SOA architecture is proposed to solve the problems of the enterprise's current information island, high coupling between various systems, and poor integration and expansion. The basic data management platform manages and synchronizes the basic data of the whole enterprise in a centralized way, so as to solve the integration problem of application systems caused by the data inconsistency.

REFERENCES

[1] Li Zi, Yang Bin. Application of Enterprise Service Bus Technology (ESB) in large enterprises [J]. Information technology, 2013, 2:146-147.

[2] Lu Hongwei. Enterprise application integration solution based on SOA and ESB [J]. Computer application and software, 2010, 4:215-216.

[3] Chen Lingping. Design and implementation of unified application service interface platform based on SOA [J]. Network security technology and application, 2009, 4:89K. Elissa, “Title of paper if known,” unpublished.

[4] Jin Baohua, he Zhenyuan, Zhang Liang, et al. Analysis and design of data sharing and exchange platform based on SOA [J]. Journal of Zhengzhou Institute of light industry (NATURAL SCIENCE EDITION), 2011, 2:102-103.

[5] Du Wanya. Design and implementation of SOA framework based on ESB [D]. Beijing: Beijing Jiaotong University, 2008.

[6] Zhang Jun. research and implementation of distributed ESB bus based on SOA Architecture [D]. Nanjing: Nanjing University of technology, 2009.

[7] Wang Li, ye Weiquan. Planning and design of Anyan tobacco data center based on SOA Architecture [J]. Computer knowledge and technology, 2013, 1:476-478.

[8] Electronic Publication: Digital Object Identifiers (DOIs):

[9] Lin Yuqin, Huang Chenhui. Research on ESB framework for enterprise application integration [J]. Computer application, 2010, 6:1658-1660.

[10] Ma Ruijuan, he Lili. Research on data exchange service system of tobacco industry based on ESB [J]. Computer programming skills and maintenance, 2013, 6:33-35.