Design and Implementation of a Manufacturing Business Collaboration Platform based ESB Technology

Sui Xin¹, ²,a, Zhu Yunlong¹,b, Nan Lin¹,c, Wu Junwei¹, ²,d

¹ Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, 110016, LiaoNing, Shenyang, China;
² Graduate School of the Chinese Academy of Sciences, 100039, Beijing, China
*a suixin@sia.cn, b ylzhu@sia.cn, c nl@sia.cn, d wujunwei@sia.cn

Keywords: ESB, SOA, Manufacturing Business Collaboration

Abstract. Enterprise Service Bus (ESB) technology, which is one of the implementation of the infrastructure of Service-Oriented Architecture (SOA), is put forward to solve various heterogeneous problems in enterprise information sharing and the management system integration. First, this paper introduces the definition and technical features of ESB and then, shows the design and implementation of an ESB-based Manufacturing Business Collaboration Platform (MBCP), which is used to solve the problems of the business integration in manufacturing industry area. Finally, we presented the application of MBCP in a steel plant as an example to illustrate the effectiveness of MBCP.

Introduction

Enterprise Application Integration (EAI) is meant to integrate application software which is distributed in various departments, so as to form a system of data sharing, business process unity and collaboration work [1]. In the process of integrating enterprise applications in manufacturing industry area, we encountered complicated, heterogeneous environment which was composed of diversified system environments, various types of communication protocols, and different kinds of programming languages. This incompatibility goes against the need for flexible integration of various applications, does no good to system extensibility, and thus results in expensive integration and maintenance cost [2].

So far, Service-Oriented Architecture (SOA) is the best answer to solve this problem. SOA defines an open-standard-based and loosely coupled architecture for integrating various types of applications. To enable transparent service integration, Enterprise Service Bus (ESB) was introduced as the infrastructure for SOA service connection and message exchange [3, 4, and 5]. By means of the key features like message transformation, message transmission and dynamic message routing, ESB can invocate and mediate the services to facilitate the interaction of disparate distributed information technology resources (applications, services, information, platforms) in a reliable manner [6].

The rest of this paper is organized as follows. Section 2 gives a short view of ESB technology. Section 3 presents the design and implementation of the ESB-based MBCP. Section 4 gives the application of MBCP in a steel plant to illustrate effectiveness of MBCP. Section 5 summarizes the advantages of MBCP-ESB and gives the possible directions of our future research.

ESB Technology

SOA and ESB. The major use of SOA is business activity reuse and enterprise-level interoperability. Its realization depends on how to solve two key issues:

1. How to align the applications of enterprises with the real business activity? Once this goal is achieved, we will no longer need to reprogram the applications when business reconstructing occurs;
instead, we only need to rearrange the sequence of applications just like we need to rearrange business activities;

2. How to integrate and reuse applications in legacy information management system? The architecture of SOA can adapt to the changes in business environment more quickly and effectively, and save the expenses of development and maintenance of software as well.

To solve the two problems mentioned above, ESB occurs into being as needed. ESB learns the concept in hardware-bus and proposes “software-bus” to manage and simplify the topology of enterprise application integration. ESB takes the responsibility of message exchanging between applications on the base of standards-based message routing. With the ability of message transformation and dynamic message routing, ESB can support message-level, event-level and service-level interoperation between heterogeneous software. As one word, ESB can orchestrate existing modules to construct more powerful and more flexible enterprise application instead of programming everything from the very beginning.

**Definition of ESB.** So far, there are various definitions of ESB in IT industry domain. This paper considers ESB as a standards-based integration platform that combines messaging, web services, data transformation, and intelligent routing to reliably connect and coordinate the interaction of significant numbers of diverse applications across the extended enterprises with transactional integrity [3]. ESB supports message-based interactions between stand-alone business applications in heterogeneous software environment. Besides the ability of connecting apps within an enterprise or cross enterprises, ESB also provides various kinds of management and monitoring on software interactions.

**Key Features of ESB.** ESB’s main purpose is to realize seamless integration of the functions (services) within enterprise information systems. In this process, ESB has to at least play two types of roles:

Message Adapter: ESB has to convert message formats for any specified service that is programmed on the base of certain type of message format and transmission protocol;

Message Router: as a message middleware, ESB needs to route message between services. When several candidates arising, ESB has to choose the most proper one as service provider according to service consumers’ needs which is contained in message itself.

Further, the two abilities mentioned above are transparent to the services which use ESB as message middleware to interact with other services in SOA ecosystem. Each individual service is developed based on its own programming or message protocol and does not need to consider the special circumstances of other services. Therefore, the change or replacement of one or more of the integrated services will not affect other services, which is called loose-coupled service integration.

Specifically, ESB provides a XML-based message high-way for web services, J2EE applications, .Net apps and many other standard-based applications, which makes it possible for information system to synchronously or asynchronously change message through the whole information chain. Precisely based on the connectivity it provides, ESB can construct complex, distributed, highly extensible integration across multiple enterprise applications, systems or even firewalls; Besides, ESB assembles existing services (rather than developing new services) to construct enterprise-level applications, which shortens the application development time and saves a lot of cost of software development and maintenance.
Manufacturing Business Collaboration Platform

Aiming at the integration demands in manufacturing industry area, this paper designs a Manufacturing Business Collaboration Platform which agrees with the principle of SOA. The architecture of this platform is shown in Fig. 1.

The architecture of the MBCP is divided into four layers:

**User Portal Layer.** The enterprise application integration in manufacturing industry area has to integrate various information management systems, like ERP, CRM and WMS. Further, these integrated systems have their own user interfaces, identity authentication system and permission management system. In order to simplify the user interaction with the MBCP and lower the cost of shifting operations in different systems, the MBCP designs a browser-based user portal which is developed under Ajax and Springs. Through this web-based user portal, users of the MBCP can monitor, orchestrate business process in manufacturing and manipulate business data in a unified way.

**Business Collaboration Layer.** In order to realize automated business integration and execution within an enterprise or across enterprises, the main function of business collaboration layer is to provide the ability to deploy, execute and monitor the business collaboration processes which across applications and enterprises. This business collaboration layer is designed according to the WS_BPEL 2.0 standard and is compatible with the advanced features of BPEL like BPEL4People and etc. With the assistance of ESB, users of the MBCP can conveniently orchestrate business activities and manage the applications in information management system.

**ESB Layer.** ESB layer (called MBCP-ESB) provides the connectivity for integrating heterogeneous enterprise applications. Generally speaking, the architecture of MBCP-ESB is a pluggable message middleware. We design and implement several plug-in binding components which meeting the integration demands in manufacturing industry. These components contain data conversion BC, Soap BC, JMX BC and some other components. More details of the architecture of the MBCP-ESB will be introduced later.

**IT Resources Layer.** The IT resources layer covers the application resources and data resources which are integrated by MBCP. In order to lower the difficulty of integrating legacy systems, we do not change the old application program interfaces of legacy system and leave the work like operating heterogeneous applications and data conversion to the ESB layer; Towards those newly developed...
systems, we directly design, implement and deploy the components in these systems under the principle of SOA.

The Architecture of MBCP-ESB. The design of the architecture of MBCP-ESB, which is shown in Fig. 2, will directly determine the efficiency, stability and scalability of the MBCP.

![Figure 2 Architecture of MBCP-ESB](image)

As an open, standard-based and event-driven lightweight SOA infrastructure, the MBCP-ESB is divided into several modules:

Component and component framework: there are two types of components named as service engine (SE) and binding component (BC). The SE realizes the control and orchestration of business process; The BC realizes message conversion and transformation based on particular type of message protocol. The pluggable component framework is a management framework that serves to manage components.

Service unit and service unit framework: service unit (SU) is the smallest function unit in MBCP-ESB, which encapsulates deployment info of the services that running in IT resources layer of MBCP. Service unit framework realizes the management of service units. The management contains the SU deployment, SU undeployment, and other runtime management.

Normalized message router (NMR): is a fast, reliable and in-memory message bus which can route normalized message between components in MBCP-ESB. The message routing of NMR is based on the service address info like endpoint-name, service-name and interface-name in the routed normalized message.

Management console: is made up of management facade and manage kernel. The browser-based management façade receives commands from clients and sends them to the management kernel. Management kernel receives, parses and executes these commands with the helps of corresponding modules like component framework and service unit framework.

Application of MBCP

Taking the actual demand of a steel plant as the background, this paper applies MBCP on the steel plant’s actual business collaboration process. An example of business collaboration process is shown in Figure 3.
As is shown in Fig. 3, this example involves several dispersed business activities and business data which exist in independent management system like browser-based user portal of MBCP, business management system, manufacturing center ERP and warehouse management system. Further, there are three absent business activities (dashed box in Fig.3) and correlative, absent business process (dotted line with empty arrow in Fig.3). Before the implementation of MBCP, this business collaboration process is separated into several independent activities which are connected through manual operation. It is obvious that we have to realize application integration and data integration to improve the efficiency of whole business collaboration process in steel factory.

In order to achieve the objective proposed above, we design and implement 19 service models, 19 data models and 10 business collaboration models, which supplement and expand the business process. With these models, we realize the business process integrations and data integrations including transactions, manufacturing, storage, and distribution. Fig.4 (a) shows the order plan business collaboration model as an example; Fig. 4 (b) shows the user portal of system.
The advantages of applying MBCP are most manifest in the following aspects:

MBCP constructed the unified service specification with outside systems and defined data models of purchase order, transport order. Building on the base of several open standards, MBCP lowered the difficulty of integrating 3rd-Party management systems.

MBCP realized the seamless information integration between steel factories, steel process centers, steel warehouses, 3rd-party logistics and clients.

MBCP achieved the unified, process-oriented information management in steel trading system and cross-system, automated business process.

MBCP improved the efficiency of steel transportation in steel trading system and increased business extensibility of steel plant’s information management system.

Conclusion

The success, that applying MBCP-ESB in the steel plant’s actual business collaboration, proves the validity and reasonableness of the design of MBCP and forcefully supports the promotion of SOA and ESB technology.

In the process of applying MBCP, we found the inefficiency and limitation of the keyword-based message routing algorithm that was using in MBCP-ESB. Therefore, the research of semantic-based message routing algorithm and correlative technology are our next research objectives.

Acknowledgment

The authors are grateful for the valuable comments and suggestions from the referees that significantly enhanced the paper. This research is supported by the National High Technology Research and Development Program (863 Program) of China under grant No. 2008AA04A105 and the Technical Program of Shenyang, China (No.F10-205-1-51)
References

[1] Jun Gui, He-biao Yang. Realization of EAI based on service-oriented architecture. vol. 2, ICEIT 2010, pp. v2-424-v2-428.

[2] S. Vinoski, “Integration with Web Services” IEEE Internet Computing, vol. 7, no. 6, IEEE, 2003, pp. 75–77.

[3] D. Chappell, “Enterprise Service Bus”, O'Reilly, 2004.

[4] Mike Gilpin, Ken Vollmer. The Forrester WaveTM, Enterprise Service Bus, In: Tech Choices, November 2005.

[5] Jean-Louis Maréchaux, ”Combining Service-Oriented Architecture and Event-Driven Architecture using an Enterprise Service Bus”, IBM Developworks, March 2006.