Service Oriented Enterprise Application Integration and its Implementation Based on Open Source Software

Dongjin Yu1,2, and Guangming Wang3
1 School of Business Administration, Zhejiang Gongshang University, Hangzhou, China
2 Hangzhou Dianzi University, Hangzhou, China
3 Zhejiang Gongshang University, Hangzhou, China
Email: yudj@hdu.edu.cn
Email: gmwang@mail.zjgsu.edu.cn

Abstract—The technology of Enterprise Application Integration could be applied for the interoperability among distributed heterogeneous systems. This paper presents the application integration framework for data exchange and business interaction based on Service Component Architecture, Message Oriented Middleware and Enterprise Service Bus. Furthermore, it introduces the implementation of the framework using some leading open source software such as Apache Tuscany, Apache ServiceMix and Apache ActiveMQ. The case conducted in the integration of 38 regional labor management information systems shows the framework is reliable and also has good performance with reduced cost.

Index Terms—Enterprise Application Integration, Open Source, Service Oriented Architecture, Framework, Labor Management Information Systems

I. INTRODUCTION

Enterprise Application Integration (EAI) is the process of linking different applications together within a single organization or across organization boundaries in order to simplify and automate business processes to the greatest extent possible, while at the same time avoiding having to make changes to the existing applications or data structures. In the words of the Gartner Group, EAI is the unrestricted sharing of data and business processes among any connected application or data sources in the enterprise [1]. EAI usually involves the data exchange which achieves a uniform data view of participating systems, and the business interaction which accomplishes mutuality across the participating organizations.

Service-oriented computing promotes the idea of assembling application components into a network of services that can be loosely coupled to create flexible, dynamic business processes and agile applications that span organizations and computing platforms. Here, services are referred as autonomous, platform-independent entities that can be described, published, discovered, and loosely coupled in novel ways [2]. Service-oriented solution is becoming a new approach to EAI whose framework could be constructed through universally accepted standards such as SCA, BEPL and WSDL.

As for the construction of EAI framework based on service oriented technology, the academic circle has already conducted a great deal of research. Related topics include the integration methodology or patterns, semantic service model, and so on. For instance, Zhang GS et al. present a formal systematic analysis, verification and validation methodology called SOARM. Based on Petri nets and temporal logic, SOARM could be well suited in EAI [3]. The idea of executable EAI patterns, which Scheibler et al. introduce in [4], is also worth attention. Using workflows customized with these configurable EAI patterns in a software-as-a-service (SaaS) business model, companies could focus on the integration without the need for the setup of complex integration infrastructures.

Most of current service-based EAI solutions adopt WSDL as the description of web service. Therefore the retrieval of right services relied on keywords will probably lead to unsuitable results with formally match but semantically not. To manage semantic differences, many propose the semantically described services for EAI. In [5], Liyi Zhang and Si Zhou give a framework of semantic SOA for EAI, and use Web Service Modeling Ontology (WSMO) as its semantic service model. Similarly in [6], Martinek et al. give an example about how to increase the effectiveness of integration by applying semantically described services.

Meanwhile, many real cases of service-based EAI have been developed. O. R. Bagheri et al. present the elastic EAI framework which has a service-based architecture and could be developed from the bottom up by means of existing technology [7]. Frequent mentioned service-based EAI cases usually occur in the fields such as Enterprise Information Portal [8], dynamic integration of Supply Chain [9], simply because these fields usually involve disparate heterogeneous systems, and more importantly, the integration in those fields could bring about huge profits.

Different with above mentioned ones, this paper gives a novel approach to service-oriented EAI framework based on open source software. With the underlying Service Component Architecture (SCA) implemented by Apache Tuscany, Message Oriented Middleware (MOM) implemented by ActiveMQ, and the Enterprise Service Bus (ESB) implemented by Apache ServiceMix, the framework fulfills both business interaction and data exchange with a variety of binding mechanisms such as Web Services, JMS and JCA, but needs less Total Cost of
Ownership (TCO) compared with using commercial products.

Service Component Architecture (SCA) provides a programming model for building applications and solutions based on a Service Oriented Architecture (SOA). It aims to encompass a wide range of technologies for service components and for the access methods which are used to connect them. In SCA, a component consists of a configured instance of an implementation, which offers services used by other components. In other words, implementations may depend on services provided by other components – these dependencies are called references [10].

Enterprise service bus (ESB) consists of a software architecture construct which provides fundamental services for complex architectures via an event-driven and standards-based messaging-engine (the bus) [11]. Unlike the more classical EAI approach of a monolithic stack in a hub and spoke architecture, the participants in the ESB-involved EAI framework do not need to interact with each other directly. Instead, the bus is responsible to deliver the request to the specific qualified service provider.

Message-oriented middleware (MOM) is the traditional infrastructure focused on sending and receiving messages that increases the interoperability of an application by allowing the application to be distributed over heterogeneous platforms [12]. MOM typically supports asynchronous calls between the client and server by the message queues which provide temporary storage when the destination program is busy or not connected.

The rest of the paper is organized in the following manner. Section 2 presents the architectural design for the service-oriented EAI framework. Section 3 illustrates its implementation based on open source software. The successfully implemented case is illustrated in Section 4. Finally, Section 5 provides concluding remarks and offers future research directions.

II. ARCHITECTURE OF SERVICE-BASED EAI FRAMEWORK

The core of service-based EAI framework is the Enterprise Service Bus configured with two kinds of adaptors to provide the interfaces of Web Services and messaging services respectively. The framework could also be extended to implement interfaces of JCA components or EJB components when required. In addition, the MOM and SCA components are bound to the bus via adaptors. The framework’s architectural topology is illustrated in Fig. 1.

A. Design for business interaction

The framework is configured with the SCA-compliant process service engine above the Enterprise Service Bus to fulfill the business interaction across system boundaries. All existing shareable function units should be reconstructed as the standard service components. The interaction therefore could be realized between the coherent interfaces of services and references. Besides, the component implemented in BPEL is configured as the portal, which orchestrates the individual service components to achieve the business value. Finally, a dedicated data repository is indispensable, where the uniform master data and shareable business status data are kept.

![Figure 1. Service-based EAI framework](image-url)
The business interaction in real scenarios would happen in the following manner. Each independent system accomplishes its own business process. When interaction across system boundaries required, a service request is issued and dispatched to the service bus. After the necessary transformation of communication protocols and data formats, the bus then routes the request to the portal component implemented in BPEL. The latter invokes corresponding service components in different systems and returns the response to the bus. The bus eventually delivers the result to the original requestor after reversal transformations of communication protocols and data formats.

B. Design for Data Exchange

The data exchange among different systems is inevitable during the process of Enterprise Application Integration. For example, considerable quantities of data would probably be migrated from one system to another after a fixed period of time. Under other circumstances, data representing business status in one system are required to keep synchronized with that on other systems simultaneously [13].

The framework realizes the data exchange based on messages. The configured Message Oriented Middleware is coupled with service bus through the specific adaptor. It maps the heterogeneous data formats, interfaces and protocols into the uniform ones, and provides reliable message filtering, pre-processing and transferring.

The framework provides two data synchronizing modes, named as Publish-Subscribe and Request-Response respectively.

1) Publish-Subscribe Mode
   The data provider does not need to send the data directly to the receiver. Instead, the data are published under certain topics. The Message Oriented Middleware will propagate the data to the receiver subscribing it via the service bus. In this mode, the sending of data is initiated by the data provider.

2) Request-Response Mode
   The node demanding data requests the Message Oriented Middleware for the data access service. The latter then interacts with the corresponding data provider via the service bus. Once the data obtained, the Message Oriented Middleware sends it back to the demanding node. In this mode, the sending of data is initiated by the demand side.

III. IMPLEMENTATION BASED ON OPEN SOURCE SOFTWARE

To construct the service-based EAI framework presented in this paper, different supporting platforms, or containers, are needed. The Service Component Architecture engine could adopt IBM Websphere Process Server or AquaLogic Data Services Platform. The Message Oriented Middleware could adopt Microsoft MSMQ or IBM Websphere MQSeries. The Enterprise Service Bus could adopt Iona Artix. However, the implementation with the above commercial products significantly increases the Total Cost of Ownership (TCO) due to the expensive license fee.

Therefore, the framework here integrates some leading open source software such as Apache Tuscany, Apache ServiceMix and Apache ActiveMQ, given in Table 1.

IV. CASE STUDIES

The above service-oriented EAI framework has been successfully implemented in the field of labor and social security administration. For this case, there are altogether 38 independent heterogeneous systems across 12 cities and counties, covering 7 lines of business including labor supervision, labor contract management, rural labor force management, employment management, unemployment management, vocational introduction and social security management. Some systems are maintained respectively by each city or county, while others provide services for the whole area.

With the emergence of frequent migration of labor forces, the demands for cross regional affairs such as remote job application, transferring of social insurance accounts become increasingly urgent. Accordingly, the service-oriented Enterprise Application Integration framework was introduced to accomplish these cross regional affairs and to monitor the demand and the required of labor forces in the whole area.

<table>
<thead>
<tr>
<th>TABLE I. OPEN SOURCE SOFTWARE ADOPTED IN THE FRAMEWORK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td><strong>Main Features</strong></td>
</tr>
<tr>
<td>1) implements SCA Assembly Model V1.1</td>
</tr>
<tr>
<td>2) offers various component types including Java, C++, BPEL, Spring and scripting</td>
</tr>
<tr>
<td>3) provides a wide range of pluggable binding protocols such as RMI, Web Services, JSONRPC and EJB</td>
</tr>
<tr>
<td><strong>Web Site</strong></td>
</tr>
<tr>
<td><strong>Licensing</strong></td>
</tr>
<tr>
<td>Apache License</td>
</tr>
</tbody>
</table>
As the demo case, the labor contract verification process is illustrated in Fig. 2, where the Integration Platform is based on the above described service-oriented Enterprise Application Integration framework. In Fig. 2, the local labor contract management system first connects the corresponding Web service running in the Integration Platform, which then accesses the central database to check the status of both employers and employees. If the verification fails, failure reasons are returned and the process terminates. Otherwise, the Integration Platform checks the contents of contract. If successful, the Integration Platform backups the contract information in the local data repository and finally sends the messages back to the local system.

The Integrated Platform has already coupled 4 separate local labor contract systems in four cities after it started up in 2008. On average, 4000 till 5000 requests of labor contract verification are handled through the Integrated Platform each day. The response time generally does not exceed one second during the peak period.

V. CONCLUSIONS

Modeling the processes of data exchange and business interaction among loosely coupled heterogeneous systems based on services meets the demands of flexibility and reusability during the process of Enterprise Application Integration. Service Component Architecture, Message Oriented Middleware and Enterprise Service Bus are three cornerstones of service-based Enterprise Application Integration framework. The adoption of open source software such as Apache Tuscany, Apache ActiveMQ and Apache ServiceMix significantly reduces the total cost of ownership of service-based Enterprise Application Integration framework. Future work may include the development of its integrated management utilities which are expected to facilitate the end-users.

ACKNOWLEDGMENT

The work is supported by the Foundation of Zhejiang Provincial Key Science and Technology Projects (No. 2008C11099-1), and China Innovation Foundation for Technology-based Firms (No. 08C26213300677).

REFERENCES