Journal of Theoretical and Applied Information Technology
20th April 2014. Vol. 62 No.2
© 2005 - 2014 JATIT & LLS. All rights reserved·

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE IMPLEMENTATION OF ENTERPRISE SERVICE BUS (ESB) IN GRADUATION BUSINESS PROCESS INTEGRATION

[1]WIRANTO HERRY UTOMO,[2] THEOPHILUS WELLEM
[1]Information System Department, Satya Wacana Christian University
[2] Information System Department, Satya Wacana Christian University
E-mail: [1]wiranto.utomo@staff.uksw.edu, [2] theophilus.wellem@staff.uksw.edu

ABSTRACT

ESB is one of SOA pillars besides WS and BPEL. ESB is an infrastructure for SOA service connection and message exchange. The main function of ESB is routing, protocol and message or data transformation. Protocol and message transformation function in ESB help to deal with protocol and data discrepancy. ESB eases connection, mediation and the re-use of service components. It also simplifies integration that will later improve integration scalability. This research has successfully integrated graduation business process that involves some units including Faculty, Library, Dormitory, Administration and Academic Bureau, Student Affair bureau, Finance Department, and Laboratory. This integration of graduation business process uses ESB as integration middleware. Web services integration reveal the role of ESB in routing and message and protocol transformation.

Keywords: *SOA, ESB, BPEL, integration, web service*

1. INTRODUCTION

Business system usually develops in a different speed from information system. This will result in the harmony problem of business system and information system. The problem will also result in applications that do not fully support business tasks. Sometimes there is a department in a company that is apart of the main business and does not support business' needs. As a result, organization becomes less flexible and difficult to adapt to market changes. Only companies whose applications can quickly and efficiently adapt to the changes of business needs will remain competitive in global market.

The inharmonic business and information system is common in almost all company or organization. The effort to deal with the problem is usually unsuccessful because of two main reasons 1) complexity of technology information architecture from a heterogenic application built from different architecture, programming language, and platforms and 2) the existing applications have to keep running when it is improved. To harmonize business and information system, an integration that also serves as mediation between the two layers is needed (Juric, 2010).

Some methods have been proposed to solve the problem of harmony in business and information systems. However, it is difficult to do if it only uses traditional approach. The latest architectural approach to help dealing with the issue that is related to this system integration is SOA. According to Shahzadam (2008), SOA is a solution to harmonize information technology and business purposes. Adopting SOA will lead to uniformity in information technology/system information department that can also lead to the improvement in the use of resources outside the companies.

In addition, according to Juric et al (2010), SOA is not a new architecture that suddenly appears. It is the evolution of integration method and distributed architecture. Before SOA, an inter-application integration method referred to as EAI (Enterprise Application Integration) had developed. In the beginning, EAI focused on application integration in companies (intra-EAI). The development of intercompany integration (B2B, business-to-business) has expanded EAI's focus into inter-EAI.

Intra-EAI integration integrates applications in a company by creating services as functionalities of the existing applications. B2B integration or inter-EAI is related to message exchange from services outside the company. SOA has improved and expanded the flexibility of the previous integration method (EAI) and distributed architecture. It has also focused on the use of existing application and system, interoperability and application integration, as well as business process composition of services or functionalities provided by application.

As a proof of concept of integration using ESB, integration of graduation business process involving some units such as Faculties, Library, Dormitories, Administration and Academic Bureau, Student Affair Bureau, Finance Department, and Laboratory will be developed. This graduation business process uses ESB as integration middleware. (See Figure1).

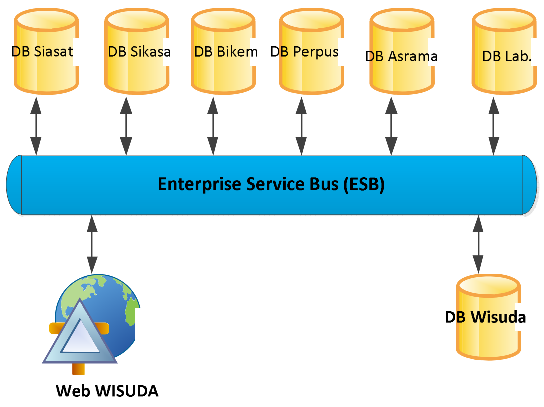

Figure 1. Graduation Business Process Integration

2. INTEGRATION ARCHITECTURE

According to Juric (2007), there are four integration architectures, 1) point-to-point, 2) hub-and-spoke, 3) enterprise message bus (EMS) and 4) ESB/SOA. Point-to-point architecture is a set of independent systems connected through a network. Hub-and-spoke architecture represents the next stage in system integration evolution by using central hub for inter-network communication. In enterprise message bus architecture, independent system is integrated using a message bus. SOA based integration architecture uses services passed through a middleware known as ESB.

Point-to-point integration model cannot be expanded and is difficult to maintain. It is related to complexity in point-to-point integration. In this point-to-point integration model, integration between N applications and N of other applications needs N(N-1)/2 interfaces. It will need fifteen interfaces if there are six application integrations and to integrate 150 applications, 11.175 interfaces are needed. The more applications integrated point-to-point, the more difficult to modify and maintain the applications.

Hub-and-spoke integration model is similar to point-to-point integration model. The distinction is in the additional hub that connects the whole applications. Message transformation and routing take place in hub. This integration model is the development of point-to-point solution by reducing the number of connections needed for integration. Because applications are not directly connected to other applications, the applications can be removed from integration topology by removing it from hub. It will reduce the chaos in the integration setup. The weakness in hub-and-spoke architecture lies on hub's centralized character. If hub experiences failure, the whole integration will fail as well. Besides, hub and spoke's ownership integration technology is locked by vendor.

In the beginning, SOA based application integration concept was a solution of the complexity of point-to-point and hub-and-spoke integration. SOA is the architectural development of service based software application. Therefore, there will be a loose bond in services integration. It allows the reuse of the existing services and creates applications that can be easily and quickly built and changed.

Orchestration using WS has two weaknesses, in scalability and the fact that it has not been able to overcome protocol and data discrepancy. To deal with the problem, WS orchestration using ESB is built. ESB is an infrastructure for SOA service connection and message exchange. The main functionalities of ESB are for routing, protocol transformation, and message or data transformation. Through the function of protocol transformation and message in ESB, protocol discrepancy can be overcome. ESB also eases connection and mediation, simplifies integration, and eases the reuse of service components so that it improves integration scalability.

Another strength of WS orchestration using ESB is that it allows business layer and information system to have a closer relation because WS orchestration is presented in high abstraction level known as business process, by hiding traditional middleware object used to support business to business interaction. Aside from that, business needs can be directly translated into business process application through WS composition.

3. SERVICE ORIENTED ARCHITECTURE

SOA is a framework in enterprise architecture and is aimed at achieving the same business goals: minimize ownership cost, create flexible business solutions that improve business solidity, reduce time to go to markets, and provide a support for global expansion. SOA substantially impacts the whole key aspects of enterprise architecture. Business service proposed by SOA forms the foundation of business and process architecture. SOA forms business architecture because business' function is exposed as services that can be divided and reused. Business processes, services, and events are converted into appropriate application services that create and support service architecture. Services form application architecture, whereas information architecture is achieved through data standardization and availability through interface service (Vos-Matthee, 2011)

SOA is a software architecture built using service oriented design principles (Erl, 2005; Sterff, 2006; Kanchanavipu, 2008; Nikayin, 2009; Reddy et al, 2009; Li et al, 2010), whereas service orientation is a concept in software engineering representing different approaches to separate interests.

It means that system's functionality is divided into smaller logical unit called service. The services are independent, but they have the ability to interact with each other through a particular communication mechanism. Therefore, Erl (2005) defines SOA component as a service, description, and message. Services communicate with others through message that allows inter-services interactions prescribed by description. Two services communicate with each other, referred to as service requester and service provider. Service requester is a service that calls other services, whereas the requested service is called service provider.

In addition to SOA definition according to Erl, there are more similar SOA definitions from other sources. Sometimes this SOA definition is called WS framework that in general can be seen in Figure 2.

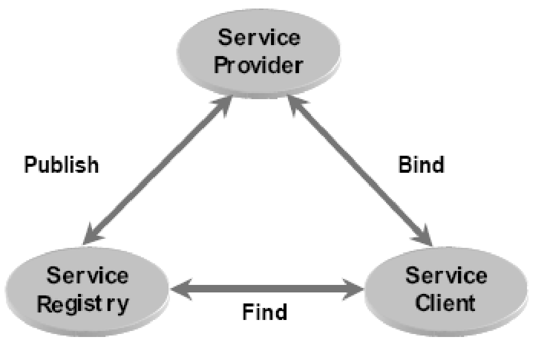

Figure 2 Service Oriented Architecture (Erl, 2005; Luthria et al, 2009; Andary-Sage, 2010)

Services provide services for public, play a role as a service provider that provides the description of its service to service registry. Service requester asks service delivery from service provider using the service registry. The two services bind each other for data exchange and other functionality usage. This fits with Erl's definition that is extended with service registry. In this service registry, services can be registered along with their service descriptions, so that the services can be found by service requesters. In this case, Erl extends his basic definition of SOA by using WS framework, which is the concrete definition of SOA based on WS.

This WS framework is a technology framework standardized and mapped into SOA model as follow: a) Services realized as WS, b) Message is described by SOAP protocol, c) Description is prescribed by WSDL, d) In this model, service registry uses UDDI

It basically fits the correlation shown in Figure 2. The common definition of SOA is used in many articles. However, this SOA definition only deals with SOA's technological aspect. It is closely related to WS based solution and its needs, but the concept can be abstracted to build SOA foundation in general.

SOA is a form of architecture technology following service oriented principles (Erl, 2005). The service oriented concept make some approaches by dividing big problems into sets of small services aimed at solving particular problems. When all problems can be divided into some services, the problems must be able to solve by allowing all services to participate in an orchestration. Therefore, there are some problems that must be possessed by services, they are how services are connected, how services communicate, how services are designed, and how inter service message is defined by Erl (2005).

As has been stated in background of the problem, SOA Delivery concept according to Erl (2005) is not enough to conduct SOA based integration. Therefore, the research does not only refer to SOA Delivery but also other concepts that have used ESB as integration middleware. Juric (2006) has used ESB as SOA based integration architecture.

4. ENTERPRISE SERVICE BUS (ESB)

ESB is an infrastructure to integrate applications and services. ESB strengthens SOA through reduction of number, size, and interface complexity between applications and services. ESB is used to connect existing and new software components to build an SOA. ESB is required to connect to some information technology resources. ESB must be flexible to integrate and reinstall components according to the changes of business needs. ESB connects the components with loose bonds that it provides the ability to integrate system into SOA and deploys gradually (Juric, 2007; Andary-Sage, 2010).

Services bus approach for integration uses technology that provides bus for application integration. Different applications do not directly communicate with each other but they communicate through SOA backbone middleware. A distinctive ESB architecture feature is the distributed character of integration topology. ESB is a set of middleware services that provide integration ability. Middleware services are the heart of ESB architecture that places message to be routed and transformed (Juric, 2007; Andary-Sage, 2010).

ESB general architecture with connected components can be seen in Figure 3. Components can play a role as either service producer or service consumer. Services can be in the form of special components such as orchestration machine, adapter for data resources or adapter for external system with message transformation or protocol transport conversion. ESB mediates inter-component message, decides location for message route, and transforms message. ESB needs persistent memory such as being connected to database (Juric, 2007; Andary-Sage, 2010).

According to Juric (2007) and Andary-Sage (2010), one approach in defining ESB general architecture is Java Business Integration specification. JBI is a standard for ESB, whereas ESB is an architectural pattern for SOA. JBI specifications describe pluggable architecture for container to service provider and component user. Services connect through Binding Component (BC) or can be hosted into the container as a part of Service Engine (SE). The services are described using WSDL. Message is always translated into general message format and routed by Normalized Message Router (NMR).

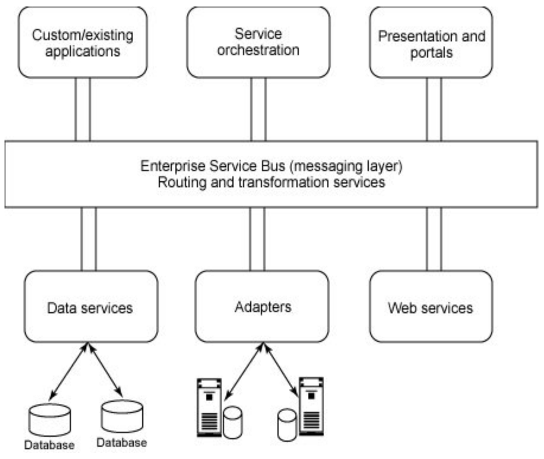
Custom/existing applications
Service orchestration
Presentation and portals
Enterprise Service Bus (messaging layer)
Routing and transformation services
Data services
Adapters
Web services
Database
Database

Figure 3 ESB architecture in general (Juric, 2007; Andary-Sage, 2010)

ESB provides strong, reliable, and safe interface communication infrastructure that can be expanded. ESB also provides communication control over the use of services that include (Juric, 2007): 1) The ability to receive message that enables it to receive request message for services and response message from services, as well as gives additional process. Through this way, ESB can play a role as an intermediary; 2) Routing ability that allows ESB to do message routing to different services that is based on content, root, or other attributes; 3) Transformation ability that enables message transformation before it is transmitted to services. For XML message format, such transformation is done using XSLT (Extensible Stylesheet Language for Transformations) or XQuery machine; 4) Control over deployment, service use and maintenance. This allows logging, profiling, load balancing, performance tuning, cost of service usage, distributed deployment, on-the-fly reconfiguration, and etc.

Other management feature involves inter-message correlation definition, reliable communication path definition, security constraints definition related to message and services, and etc.

5. REASEARCH FINDINGS & DISCUSSION

5.1 Use Case Diagram

This Use Case diagram described system's needs. Use Case diagram is used to describe what the system will do, system functional needs, and system functionality expected by the environment. Supplementary specification is the need that has never been mapped to Use Case specification that contains non functional needs such as code maintenance, reliability, performance and system support or constraint system, as well as safety. Figure 4 shows the need of graduation information system.

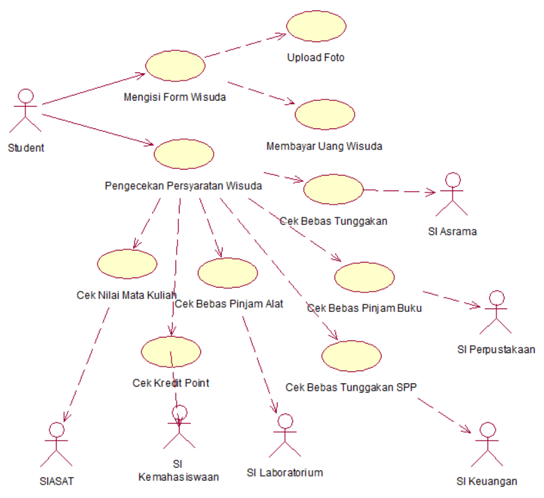

Figure 4 Use Case diagram of Graduation business process

5.2 System Architecture

In this research, the platform used to realize the integration is Java EE platform with the support of OpenESB tool as middleware ESB infrastructure. The system architecture used to realize this integration model can be seen in Figure 5.

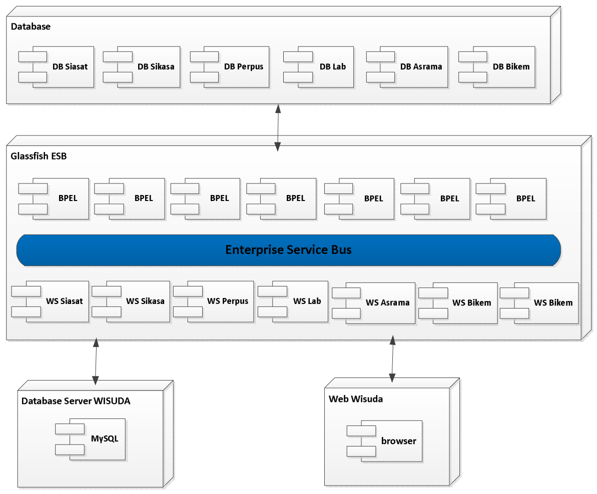

Figure 5 Graduation business process system Architecture

The specification of the hardware used is: Intel Pentium Core 2 Duo 2.0 Ghz processor, Memory of 3.0 GB RAM DDR2, and Hardisk 300 GB. The specification of the software used is Microsoft WindoWS XP SP3 operation system, Java SDK Standard Edition 1.6.0. update 20 version, Netbeans IDE 6.7.1, and Glassfish ESB 2..1 (Open ESB).

5.5. Implementation with Glassfish ESB

Business Process Implementation using BPEL 2.0 in Glassfish ESB can be seen in figure 8. The business process begins with service consumers, college students, who request to graduation business process. Service consumers are implemented with WSDL transmitted using SOAP BC, whereas service provider is in the form of BC Database wrapped by WSDL. Next, the students receive SOAP BC response. This case reveals the different protocol and message format used by service consumers and service provider. Service consumers use SOAP, whereas service provider uses database.

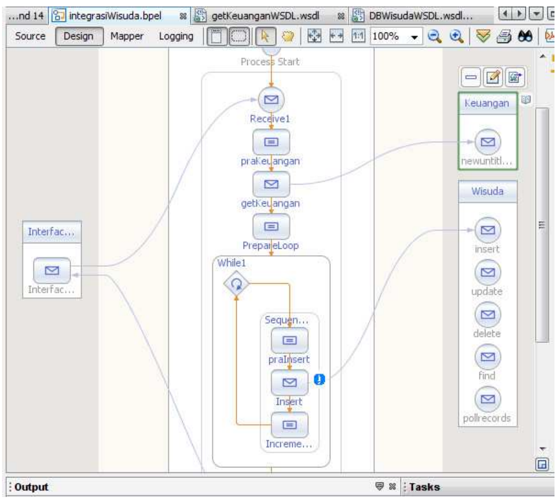

Figure 6 Graduation business process implemented using BPEL

The component diagram will be implemented using Glassfish ESB in the form of Composite Application stored in file using AplikasiIntegrasiCA.zip that can be seen in Figure 7.

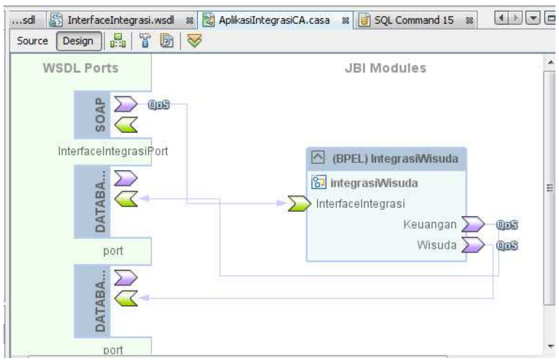

Figure 7 Implementation Result in the form of Composite Application

Figure 7 is the implementation of Component Diagram into Composite Application, deployed to Glassfish v2.1.application server.

6. ESB AS INTEGRATION MIDDLEWARE

ESB is one of SOA pillars besides WS and BPEL. WS provides only point-to-point integration, which is no longer appropriate to integrate large quantities applications. This problem can be solved using indirect connections inter-application through ESB that provides facilities to route content or context based message.

Besides, there are two heterogeneity problems. The first is the discrepancy of among communication protocols used among service consumers and service provider. The discrepancy does not allow users to request the services provided by service provider. ESB can solve the problem by providing facilities to convert a communication/transport protocol into other needed protocols. For example, this facility will transform HTTP protocol into SMTP protocol. Through the facility, application can communicate with each other even though the protocol of service consumers is different from service provider.

The second homogeneity problem is related to the discrepancy between message format used by service consumers and service provider. The problem is solved by ESB that provides facilities to transform message format used by service provider and service consumers. For example, the facility can transform SOAP message into other XML based format.

The three roles of ESB in term of routing, protocol transformation, and message/data transformation has been performed successfully in the research. JBI module produced by the research has proven the use of ESB to route service users to local and external service provider.

Protocol transformation has also been performed successfully through the JBI module. JBI module reveals HTTP inter-protocol communication using JDBC, between HTTP and SMTP, or between HTTP and FILE. Message transformation can be done through XML data format.

7. CONCLUSION

Graduation business process involving Faculties, Library, Dormitory, Administration and Academic Bureau, Student Affair Bureau, Finance Department, and Laboratory has been performed successfully. This graduation business process uses ESB as integration middleware. Through web service integration, it can be known the role of ESB in routing and message and protocol transformation.

The use of ESB in this integration method can also solve the complexity of information technology architecture from heterogenic application built from different programming architecture and language as well as different platform.

REFRENCES:

[1]. Andary, J.F. and Sage, A.P., 2010, The role of service oriented architectures in systems engineering, Information Knowledge Systems Management 9 (2010), IOS Press

[2]. Erl, T., 2005, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, Upper Saddle River, New Jersey 07458

[3]. Erl, T., 2004, Service Oriented Architecture: A Field Guide to Integrating XML and Web services, Prentice Hall PTR, Upper Saddle River, New Jersey 07458

[4]. Erl, T., Karmarkar, A., Walmsley, P., Haas, H., Yalcina, U., Liu, C.K., Orchard, D., Tost, A., dan Pasley, J., 2008, Web service Contract Design and Versioning for SOA, Prentice Hall PTR, Upper Saddle River, New Jersey 07458

[5]. Juric, M.B., Loganathan, R., Sarang, P., dan Jennings, F., 2007, SOA Approach to Integration, Packt Publishing, Birmingham, B27 6PA, UK.

[6]. Juric, M.B., Mathew, B., dan Sarang, P. 2006, Business Process Execution Language for Web services, Packt Publishing, Birmingham, B27 6PA, UK.

[7]. Juric, M.B., Chandrasekaran, S., Frece, A., Hertis, M., dan Srdic, G., 2010, WS-BPEL 2.0 for SOA Composite Applications with IBM WebSphere 7, Packt Publishing Ltd. 32 Lincoln Road Olton Birmingham, B27 6PA, UK.

[8]. Kanchanavipu, K., 2008, An Integrated Model for SOA Governance An Enterprise Perspective, Master Thesis, IT University of Göteborg Chalmers University of Technology and University of Gothenburg, Göteborg, Sweden

[9]. Li, G., Muthusamy,V. and Jacobsen, H., 2010, A Distributed Service-Oriented Architecture for Business Process Execution, ACM Transactions on The Web, Vol. 4, No. 1, Article 2, Publication date: January 2010.

[10]. Luthria, H. And Rabhi, F., 2009, Service Oriented Computing in Practice – An Agenda for Research into the Factors Influencing the Organizational Adoption of Service Oriented Architectures, Journal of Theoretical and Applied Electronic Commerce Research, ISSN 0718–1876 Electronic Version VOL 4 / ISSUE 1 / APRIL 2009 / 39-56, © 2009 Universidad de Talca – Chile

[11]. Nikayin, F.A., 2009, Adopting A Theoretical Method For The Development Of A Service-Oriented Information System, Dissertation, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur

[12]. Reddy, V.K., Dubey, A., Lakshmanan, S., Sukumaran, S. and Sisodia, R., 2009, Evaluating legacy assets in the context of migration to SOA, Software Qual Journal (2009) 17:51–63, Springer Science+Business Media

[13]. Shahzadam B.M., Jan, J., Wim, G., and Herwig, M., 2008, Aligning Technology With Business An Analysis Of The Impact Of Soa On Outsourcing, Journal of Theoretical and Applied Information Technology, published by www.jatit.org.

[14]. Sterff, A., 2006, Analysis of Service-Oriented Architectures from a business and an IT perspective, Master Thesis, Technische Universität München, Fakultät für Informatik

[15]. Vos, W., and Matthee, M.C., 2011, Towards A Service-Oriented Architecture: A Framework For The Design Of Financial Trading Applications In The South African Investment Banking Environment, South African Journal of Industrial Engineering May 2011 Vol 22(1)

[16]. [1] T.S. Bhatti, R.C. Bansal, and D.P. Kothari, "Reactive Power Control of Isolated Hybrid Power Systems", *Proceedings of International Conference on Computer Application in Electrical Engineering Recent Advances (CERA)*, Indian Institute of Technology Roorkee (India), February 21-23, 2002, pp. 626-632.

[17]. [2] B.N. Singh, Bhim Singh, Ambrish Chandra, and Kamal Al-Haddad, "Digital Implementation of an Advanced Static VAR Compensator for Voltage Profile Improvement, Power Factor Correction and Balancing of Unbalanced Reactive Loads", *Electric Power Energy Research*, Vol. 54, No. 2, 2000, pp. 101-111.