IMPROVING SERVICE REUSABILITY USING ENTERPRISE SERVICE BUS AND BUSINESS PROCESS EXECUTION LANGUAGE

AKO ABUBAKR JAAFAR

A project report submitted in partial fulfillment of the
requirements for the award of the degree of
Master of Science (Computer Science)

Faculty of Computer Science and Information Systems
University of Technology Malaysia

JANUARY, 2013

TO MY PARENTS, WIFE AND FAMILY FOR THEIR LOVE AND SUPPORT

ACKNOWLEDGFEMENT

First of all, I would like to warmly express my utmost gratitude to Allah S.W.T for His blessing and given strength during the completion of this research.

I would like to express my sincere appreciation to my supervisor Associate Prof. Dr. Dayang Norhayati Abang Jawawi. She has been extremely helpful and offered great guidance throughout this research. Without her continual support and guidance, this research report would not have been the same as presented here.

Special thanks to my colleagues and friends for the collaboration in knowledge-sharing with each other. My sincere appreciation also extends to all of the lecturers and members in Faculty of Computer Science and Information System, UTM.

Last but not least, my heartfelt thanks go to all of my family members (my parents and my lovely wife), whose sacrifice, support, love, caring inspired me to overcome all the difficulties throughout my entire academic life.

ABSTRACT

Despite the ability of current technologies to integrate different applications together, it also makes integrating application and systems more complicated due to poor reusability and coupling in the present technologies. The integration solution of a traditional enterprise application usually focuses on Point to Point (P2P) integration. Generally, this type of integration creates tight coupling and complex integration. Service Oriented Architecture (SOA) based integration application is the most current solution which transforms IT systems into highly reusable and loosely coupled services. Web Services are the most prominent and experienced technology under the SOA's flag. But using Web Services in SOA-Based integration alone still provides P2P integration. The middleware Enterprise Service Bus (ESB) and orchestration language Business Process Execution Language (BPEL) together can provide loosely coupled and reusable integration. Web Services and ESB/BPEL have a different level of reusability. Consequently, this study attempts to investigate the reusability level of ESB/BPEL as compared to Web Services. The evaluation of ESB/BPEL and Point to Point Web Services (P2PWS) has been conducted using the Islamic Banking System integration case study. The implementation services were examined and evaluated using quality model metrics to exhibit the reusability level of ESB/BPEL and P2PWS. Besides the reusability level, the response time of the two approaches has been measured to illustrate architecture impact on performance. The result shows that the ESB/BPEL had a higher level of reusability but poor response time as compared to P2PWS.

ABSTRAK

Walaupun teknologi masa kini berupaya mengintegrasikan aplikasi yang berlainan, ia juga membuat integrasi aplikasi dan sistem lebih rumit disebabkan oleh kebolehgunaan semula yang kurang memuaskan dan gandingan dalam teknologi sekarang. Penyelesaian integrasi dalam aplikasi perusahaan biasa lazimnya tertumpu kepada integrasi titik ke titik (P2P). Secara amnya, integrasi sebegini membentuk gandingan yang ketat dan integrasi yang rumit. Aplikasi integrasi berdasarkan berorientasikan perkhidmatan seni bina (SOA) ialah penyelesaian yang paling terkini yang mengubah sistem IT menjadi perkhidmatan sistem guna semula yang tinggi dan hubungan gandingan yang longgar. Perkhidmatan Web adalah teknologi yang paling ketara dan paling lama digunakan dibawah bendera SOA. Tetapi menggunakan perkhidmatan Web di dalam integrasi berdasarkan SOA sahaja masih menyumbang kepada integrasi P2P. Perantaraan Bas Perkhidmatan Perusahaan (ESB) dan (BPEL) bersama-sama menyumbang kepada gandingan yang longgar dan integrasi penggunaan semula. Perkhidmatan Web dan ESB/BPEL mempunyai tahap kebolehgunaan semula yang berbeza. Oleh itu, kajian ini cuba untuk menyiasat tahap penggunaan semula ESB/BPEL berbanding dengan perkhidmatan Web. Penilaian ESB/BPEL dan (P2PWS) telah dikendalikan menggunakan kes kajian integrasi sistem perbankan Islam. Implementasi perkhidmatan-perkhidmatan telah dikaji dan dinilai menggunakan metrik model kualiti untuk menunjukkan tahap kebolehgunaan semula ESB/BPEL dan P2PWS. Selain daripada tahap kebolehgunaan semula, tempoh tindak balas kedua-dua pendekatan tersebut telah diukur untuk menggambarkan kesan reka bentuk ke atas pencapaian mereka. Keputusan menunjukkan bahawa ESB/BPEL mempunyai tahap tempoh tindak balas semula yang lebih tinggi tetapi , tempoh tindak balas yang kurang memuaskan berbanding dengan P2PWS.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABRIVATIONS

IB	-	Islamic Banking
SOA	-	Service Oriented Architecture
EAI	-	Enterprise Application Integration
ESB	-	Enterprise Service Bus
BPEL	-	Business Process Execution Language
WS	-	Web Services
WSDL	-	Web Service Definition Language
SOAP	-	Simple Object Access Protocol
RPC	-	Remote Protocol Call
CORBA	-	Common Object Request Broker Architecture
DCOM	-	Distributed Component Object Model
CB	-	Core Bank
ATM	-	Auto Teller Machine
IF	-	Islamic Finance
XML	-	Extensible Markup Language
UDDI	-	Universal Description, Discovery, and Integration
IFI	-	Islamic Financing Institutions
IT	-	Information Technology
SBB	-	Shariah Supervisory Board

P2P	-	Point-to-Point
A2A	-	Application-to-Application
WSCI	-	Web Service Choreography Interface
UML	-	Unified Modeling Language
IIS	-	Internet Information Server
GUI	-	Graphical User Interface
WUI	-	Web User Interface
IBF	-	Islamic Banking Finance
CMS	-	Customer Management System
DFDI	-	Dijlah and Furat Bank for Development and Investment

LIST OF APPENDIX

CHAPTER 1

INTRODUCTION

1.1 Preview

Today, software and systems have become the keys for doing successful business. The efficiency of a system has a major role to determine the success of a business. Therefore, a distribute system environment totally depends on the speed and agility of IT implementation to provide services and products. The necessity of the complex systems was felt with the demand from the large scale company or requirement. For the last decades, software systems have became more complex and larger such as banking sector. The traditional approaches which is developed a system from scratch eventually resulted the problems like failure to meet customer requirements, budget constraint and extend deadline. Therefore the industries cannot afford to write and develop a new system from scratch and replace it with the legacy system because this operation needs huge effort and huge cost. Then, in order to overcome those problems, it was realized that the software reusability is the essential factors that contributing of minimizing the existing problems. Software reusability allows parts of software component to be renewed or replaced by existing components. Thus, software reusability helps software developers to concentrate on adding more complex system functionalities rather than focusing on developing basic components.

The competitive business enterprises need to manage internal and external information, so the needs of enterprise application integration has always existed

(Center, 2006) because current business solution innovations require integration of various business units, applications, business systems, and enterprise data. Integrated systems have a great advantage to improve its competitiveness with unified and efficient access to the information. Hence, it is easy to see the importance of integration (Matjaz *et al.*, 2007). The current using Service Oriented Architecture (SOA) with loosely coupled nature can eliminate the barriers in application integration. It can follow a modular approach to add new services or update existing services without changing the original system and can make the new application systems easily to integrate into other types heterogeneous systems, so that enterprises can make changes according to the demand changes of market (Pan *et al.*). Enterprise architects believe that SOA can help businesses respond cost-effectively and more quickly with changing of market conditions. SOA promotes reuses at the service level rather than class level. It can also integrate simply with the existing (legacy) assets and use them (Center, 2006).

SOA defines a new and evolving architecture method for building distributed applications. Distributed component exposed as a service that provide well-defined interfaces to deliver and process XML messages. This approach makes sense for establishing and building solutions that cross corporate domain boundaries, departmental, and organizational. SOA used by business with multiple applications and systems on different platforms to build a loosely coupled integration solution that implements unified workflows (Jeffrey and Mauricio, 2006). Gartner Research in 2005 predicted that by 2008, SOA will be the basis for 80 percent of development projects (Hayward, 2005).

SOA is not obliged to any particular technology; In fact, the designer can implement SOA combining a wide range of technologies like Web Services (WS), Web Service Definition Language (WSDL), Simple Object Access Protocol (SOAP), Remote Protocol Call (RPC), Common Object Request Broker Architecture (CORBA), Distributed Component Object Model (DCOM), and Enterprise Service Bus (ESB).

Web services are popular approach to implement SOA, which services can be accessed over the Internet independent of platforms and programming languages (Rosenberg, 2010). Also ESB is another famous approach to implement SOA. ESB is a software infrastructure that simplifies the integration and flexible reuse of business components using a SOA (Matjaz *et al.*, 2007). On the other hand using XML based orchestration Business Process Execution Language (BPEL) enables task sharing across enterprises using a combination of Web Services (Aarti *et al*., 2011). ESB as middleware resolve the integration issue and BPEL support system integration and collaboration (Koufi *et al.*, 2010). BPEL uses to orchestrate complex flows, for example when it contains calling of external services and processing their results to create result own result.

1.2 Problem Background

Up to date, quite number of studies and project has been done on SOA and application integration. A traditional enterprise application usually appears in the form of various independent application systems. Accordingly, the integration solution of a traditional enterprise application usually focuses on Point to Point (P2P) integration. P2P were the first type of integration. Generally, this type of integration creates tight coupling (Votis *et al.*, 2008). On the one hand, is complicated because in this way every system needs its own application interface program. On the other hand, it cannot provide necessary flexibility, adaptability and reusability and thus impose great burden on the related technical staff in the enterprise (He *et al.*, 2009). The next drawback of this style of integration is that integration logic is spread over all of integrated systems and there was no central control, administration and management (Kusak, 2010). This was accomplished through Remote Procedure Call (RPC) which resulting to complex mesh network. To resolve complexity the Hub/Spoke architecture was introduced. In this architecture, a middleware (hub) serves as a central point of integration through which all business applications were connected using various application communication protocols like Internet Inter Object Protocol (IIOP) (Gu and Zhang, 2010). This approach was successful at

reducing complexity; however, it maintains a tight coupling between the integrated applications which made it difficult to response to changes in business processes. Service Oriented Architecture can be defined as an Architectural design style and principles that enable the integration of enterprise applications as linked loosely coupled services (Eric and Hugh, 2006).

The services can be existing legacy applications modules that are wrapped and exposed as web services or newly developed J2EE or .NET Web services components. Web service and SOA can realize not only business function integration but also data integration. And it can expediently encapsulate legacy system to integrate with SOA system, having lower cost and realizing software reusing (Votis *et al.*). A Web service, however, greatly simplifies the development of the enterprise applications by dividing them into various modules which can be easily integrated and reused. In this way, information sharing and data interaction based on the same services framework can be realized within the enterprise and even among enterprises (He *et al.*, 2009).

Web Service has a great flexibility by relaying to standards such as XML, WSDL, SOAP, and the ability of reusing functionality of SOA. According to this Web Service enabled SOA offers a solution to the standards problem by avoiding the central point of integration, which is often a bottleneck in previous EAI solutions. Also it reduces the number of P2P adapters because each interface is based on WSDL and it can communicate with each other WSDL enabled interface (Lu and Zhang, 2009). But still using the Web Service alone provides the P2P integration (Riad *et al.*, 2010).

The implementation of SOA solves the problem of interoperation during service module calling. To better serve enterprise applications, ESB is introduced (He *et al.*, 2009). ESB is a collection of middleware services which provides integration capabilities. The integration logic is placed inside the ESB and integrated systems are connected to the ESB via entry points. ESB came to the rescue. An ESB provides many functions, including protocol and message transformation, message

routing based on content and context, location transparency, Quality of Service (QoS), data enrichment, and other functions. Also, developing applications that support web services interfaces will not be enough to provide complete and coordinated business processes. Therefore, another approach is needed to compose and organize these web services together in order to form processes definition (Abdaladhem *et al.*, 2009). Using XML based orchestration BPEL with ESB enables task sharing across enterprises using a combination and Orchestration of Web Services (Aarti *et al.*, 2011). Hence each of these technologies and approaches has a set of capabilities that make it a suitable and compatible option of realization for specific demands and cases (Raid *et al.*, 2010).

Nowadays quit number studies have been done to evaluate each of these approaches to show there capability and suitability. Fraz (2008) evaluate ESB and P2PWS to E-Learning system for some criteria such as Scalability, Reliability, and Security which shows that P2PWS is better than ESB to be applied in small and medium system. Graaff (2009) illustrate the comparison of P2P, traditional EAI and ESB based on some criteria such as messaging, reliability, flexibility, scalability, and security which show the advantage of ESB over traditional EAI. Sabooniha *et al.* (2012) evaluate Traditional EAI techniques such as CORBA, and DICOM for the Hospital Information System (HIS) in term of flexibility, reliability, reusability, complexity, and maintainability that show the CORBA and other technique of traditional EAI increase reusability of applications. Kruessmann *et al.* (2009) evaluate availability of open source ESB tools. Ahuja *et al.* (2011) evaluate ESB performance for some open source ESB tools.

One application which can benefit from reusability advantages of SOA is Islamic Banking (IB). Banking system is the most complicated parts in software development and software market, because this kind of system needs to develop various systems and advance technologies and techniques to integrate these systems together (Raid *et al.*, 2008). In addition, these systems become more complex and heterogeneous due the increasing business requirements with growing new customer's needs, therefore, banking system should be developed on the basis of market demands. Islamic banking (IB) is a one of those requirements that emerged in

the Middle East countries during the last decades and European countries in recent years. The IB is a financial system which is based on Islamic law and guided by Islamic economics which made the system even more complex than the conventional banking system. Therefore software reusability helps these kinds of system to decrease the complexity and the cost.

This research use IB as a case study, so as the perspective of banking system there are many attempts to use SOA based integration in the banking system. Raid *et al.* (2008) proposed banking integration based on SOA and web services to solve point-to-point integration but this kind of integration using web services still in point-to-point integration and its lack of fully loosely coupled, flexible reusability. Many banking systems deploy Enterprise Application Integration (EAI) to integrate front ends and core banking systems which are suffering in lack of scalability, loosely coupled, and heterogeneity. So Jenny *et al.* (2006) proposed ESB to integrate front-end application and back-end applications in traditional banking system with chorography approach to composite application which is still not a flexible approach and lack of loosely coupled between presentation layer and process layer.

Furthermore some studies have been done on SOA and Islamic banking. Halimah *et al.* (2008,2009,2010) has develop Islamic banking system (HiCORE) which address limitation problem to create a new product for the customer based on Islamic principles that cover both deposits and financing. Also, an Enterprise Application Integration subsystem based on bus technology is used to integrate all applications together without considering detail of integration issue in IB. So until now many studies have been done on the SOA based integration to a specific domain such as banking system, healthcare system, and supply chain management system as well as in general domains. Also, many studies have been done to illustrate the advantages and drawbacks for each approach. But still there is not quit studies that show ESB/BPEL effect in term of reusability as well as the advantages of ESB/BPEL over the P2PWS. Also there are not many studies that evaluate ESB/BPEL in the banking sector especially the IB.

1.3 Problem Statement

In relation to the issues mentioned, there are some problems and requirement such as 1) The most of current SOA or non SOA based integration, such as traditional EAI hub-spoke to integrate existing applications increase reusability but still suffer to provide high reusability, complexity, and loosely couple to achieve flexible integration. 2) Also studies show that a Web Service and ESB promote the high reusability which does not show the real reusability ratio of these approaches. 3) The most of studies and evaluations on ESB does not consider the orchestration service and its effect on reusability.

However, still there is not much research dealing with improving reusability of application integration based on ESB with consideration to orchestration approach like BPEL. Even so, a previous research does not show the real service reusability with difference approach to integration such as Web Service and ESB/BPEL. Thus, this research will answer the following questions:

- How the ESB and BPEL improve the reusability of service?
- How difference approach of application integration like (P2PWS and ESB/BPEL) affect service reusability and loosely coupled?
- What is the draw back of ESB/BPEL in order to increase the reusability and loosely coupled.

In order to answer this question, a set of research questions that will aid to solve this problem are defined as below:

1. What is SOA and SOA approaches to integration?

    - What are the services in SOA and what is SOA feature in the integration?
    - What are the SOA-based application integrations and service composition approaches?

- How SOA with integration technology helps specific domain such as IB system in the integration issue to achieve loosely couple, and reusability?

1.4 Research Aim

The aim of this research is to apply the Enterprise Service Bus (ESB) with Business Process Execution Language (BPEL) together to integrate application in the IB system and evaluate reusability of ESB/BPEL. Also compare the evaluation with P2PWS to show the differences between them and show the performance drawbacks of ESB/BPEL in term of response time.

1.5 Research Objectives

According to the problems descriptions above the objectives of this research are:

1. To identify and study of current integration approaches based on SOA.
2. To propose and apply ESB/BPEL with SOA integration application framework model.
3. To implement the chosen approach using a case study.
4. To evaluate reusability of ESB/BPEL approach using quality model metrics and compare it with P2PWS approach.

1.6 Research Scopes

In this research a sample of Islamic banking services will be taken for implementing and demonstration of integration process, and some of the Islamic principles for IB will be defined. Overall, the scope of this research is on the following area:

1. This research was inspired by the concepts of the Web Service and its initiatives proposed both in academic and in industry. This concept is XML based and uses standard protocols such as Universal Description, Discovery, and Integration (UDDI), Web Services Description Language (WSDL), and Simple Object Access Protocol (SOAP), Business Process Executing Language (BPEL), and Enterprise Service Bus (ESB).
2. This research focuses on application integration and process integration in IB system as well as proposes ESB/BPEL to integrate applications and processes.
3. Dot Net technology (C#) was used to develop services.
4. This research uses as a case study (DIJLAH & FURAT BANK FOR INVESTMENT AND DEVELOPMENT) in Iraq to demonstrate Islamic bank structure and integration approach of IB system.

1.7 Research Significant

The study focuses on integration of SOA, and IB service architecture and integration of IB with corresponding banks. This study provides some idea which can be used as a benchmark for researches and practitioners who are interested in system integration and architecture. This research could be a part of research work on system reusability, and loosely coupled. The advantages that can be derived from this study are to motivate the use of new technologies to improve the reusability and loosely coupled for application and services.

REFERENCES

Aarti, M. K., Vaibhav, N. C., and Meshram, B. B. (2011). Working of Web Services Using BPEL Workflow in SOA. *ICAC3 2011, CCIS 125*, 7.

Abdaladhem, A. S., Partrik, F., and Jacque, P. (2009). Web Services Orchestration and Composition. Retrieved Aug 12, 2012, from http://diuf.unifr.ch/drupal/softeng/sites/diuf.unifr.ch.drupal.softeng/files/file/publications/albreshne/download/WP09-03.pdf

Ahuja, S. P., and Patel, A. (2011). Enterprise Service Bus: a performance evaluation. *Communications and Network*, *3*(3), 133-140.

al-Roubaie, A., Barhrain, and Alvi, S. (2009). *Islamic Banking and Financ CRITICAL CONCEPTS IN ECONOMIC* (Vol. 1): Routledg.

Cambronero, M. E., Diaz, G., Martinez, E., and Valero, V. (2009). *A Comparative Study between WSCI, WS-CDL, and OWL-S.* Paper presented at the e-Business Engineering, 2009. ICEBE '09. IEEE International Conference on.

Center, K. (2006, Jan 7 2005). Service-Oriented Architecture (SOA). Retrieved Jul 31, 2012, from http://www.itcsoftware.com/code/soa.htm

Choi, S. W., and Kim, S. D. (2008, July). A quality model for evaluating reusability of services in soa. In *E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services, 2008 10th IEEE Conference on* (pp. 293-298). IEEE.

Corporation, I. Service-oriented architecture. Retrieved Jun 13, 2012, from http://pic.dhe.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Fcwbs_soa.html

Dijlah and Furat bank. (2010). Dijalh & Furat Overview. Retrieved 1 Oct, 2012, from http://www.dfdi-bank.com/english_index.html

El Hog, C., Ben Djemaa, R., and Amous, I. (2012, March). Profile annotation for adaptable Web Service description. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 1935-1940). ACM.

Eric P., and Hugh *T. (2006). Understanding Enterprise SOA. USA: Manning Publications.*

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. NEW YORK: PRENTICE HALL.

Fraz, U. (2008). Evaluation and comparison of SOA implementations to enhance E-Learning systems.

Fuhr, A. (2011). *Identifying Legacy Code for Service Implementation using Dynamic Analysis and Data Mining.* University of Koblenz-Landau.

Gao, T. (2007). *The Complete Reference To Professional SOA With Visual Studio 2005 (C# & VB 2005) .NET 3.0*: LULU Press.

García-Jiménez, F. J., Martínez-Carreras, M. A., and Gómez-Skarmeta, A. F. (2010). Evaluating Open Source Enterprise Service Bus. In *e-Business Engineering (ICEBE), 2010 IEEE 7th International Conference on* (pp. 284-291). IEEE.

Graaff, V. (2009). *Towards Distributed Information Access Possibilities and Implementation.* University of Twente.

Greg, L, and Eric, N. (2005). Introduction to SOA with Web Services *Understanding SOA with Web Services*: Addison-Wesley Professional.

Gu, C., and Zhang, X. (2010, July). An SOA Based Enterprise Application Integration Approach. In Electronic *Commerce and Security (ISECS), 2010 Third International Symposium on* (pp. 324-327). IEEE

Huynh, Q. T., Pham, T. Q., and Tran, Q. V. (2007). The Reusability and Coupling Metrics for Service Oriented Softwares.

Halimah, B. Z., Sembok, T. M., Azlina, A. S., Sharul, A., M.N., Azuraliza, A. B., Zulaiha, A. O., Nazlia, O., Salwani, A., Sanep, A., Hailani, M. T., Zaher, M. Z., Azizah, J., Nor Faezah, M. Y., Choo., W. O., Abdullah, C., and Sopian, B. (2008). Holistic Islamic Virtual Banking System: SOA parameter- based semantic approach (HiCORE). Paper presented at the Information Technology, 2008.

Halimah, B. Z., Azlina, A., Sembok, T. M., Suffian, I., Shahrul Azman, M. N., Azuraliza, A. B., Zulaiha, A. O., Nazlia, O., Salwani, A., Sanep, A., Hailani, M. T., Zaher, M. Z., Azizah, J., Norfaezah, M. Y., Choo., W. O., Abdullah,

C., and Sopian, B. (2010). Evaluation of HiCORE: Multi-tiered Holistic Islamic Banking System based on User Acceptance Test. Paper presented at the Information Technology (ITSim), 2010.

Hanif, M. (2011). Differences and Similarities in Islamic and Conventional Banking. *International Journal of Business and Social Science, 2*(2), 10.

Hansen, D. (2008). Blueprint for Successful SOA Integration. Retrieved Aug 3, 2012, from http://www.oracle.com/technetwork/articles/entarch/blueprint-soa-integration-087583.html

Hayward, S. (2005). *Positions 2005: Service-Oriented Architecture Adds Flexibility to Business Processes,* Gartner, Inc.

He, X., Li, H., Ding, Q., & Wu, Z. (2009, December). The SOA-Based Solution for Distributed Enterprise Application Integration. In *Computer Science-Technology and Applications, 2009. IFCSTA'09. International Forum on* (Vol. 3, pp. 330-336). IEEE.

Holdings, T. (2007). *Islamic Banking – Product Summary*.

Hunt, R. (2007). *Islamic Banking: Core Vendors Fill Growing Demand for Shari'ah-Compliant Banking*: TowerGroup.

Huynh, Q. T., Pham, T. Q., and Tran, Q. V. (2007). The Reusability and Coupling Metrics for Service Oriented Softwares.

Jeffrey, H., and Mauricio D. (2006). *Expert Service-Oriented Architecture in C# 2005: Defining Web services development with ASP.NET and WSE 3.0* (2 ed.). NewYork: apress.

Jenny, Chen, A., and Hsu. (2006, Oct. 2006). *Implementing a Banking Front End Processor in Taiwan Using SOA.* Paper presented at the e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on.

Jr, S. (2007). Getting on Board the Enterprise Service Bus. *I E E E C o m p u t e r S o c i e t y*.

Kabir, H., and Mervyn, L. (2007). Islamic banking: an introduction and overview. Kabir, H., and Mervyn, L. (Eds.), *Handbook of Islamic Banking*. Cheltenham, UK: Edward Elgar.

Koufi, V. M., Papakonstantinou, F., and Vassilacopoulos, G. (2010). Using ESB and BPEL for Evolving Healthcare Systems Towards Pervasive, Grid-Enabled SOA Information Systems Development. In G. A. W. Papadopoulos, Wita

Wojtkowski, Gregory Wrycza, Stanislaw Zupancic, and Jože (Ed.), (pp. 167-175): Springer US.

Kruessmann, T., Koschel, A., Murphy, M., Trenaman, A., and Astrova, I. (2009, June). High availability: Evaluating open source enterprise service buses. In*Information Technology Interfaces, 2009.* ITI'09*. Proceedings of the ITI 2009 31st International Conference on* (pp. 615-620). IEEE.

Kusak, D. (2010). Comparison of Enterprise Application Integration Platforms. Charles University in Prague Faculty of Mathematics and Physics.

Li, Q. (2009). *Bank Application System Integration based on SOA and EAI*. Paper presented at the International Conference on Industrial and Information Systems.

Li, Z., Yan, W., and Fujiang L. (2011). *The Banking System Reference Implementation Based on SOA*. Paper presented at the Fourth International Joint Conference on Computational Sciences and Optimization.

Lu, W., and Zhang, D. (2009, December). Research of Enterprise Application Integration Base on Service Oriented Architecture. In *Computational Intelligence and Software Engineering, 2009. CiSE 2009. International Conference on* (pp. 1-9). IEEE.

Manes, A. T. (2007). *Enterprise Service Bus: A Definition*: Burton Group.

Mannava, V., and Ramesh, T. (2012, October). A Service Configuration and Composition Design Pattern for autonomic computing systems using Service Oriented Architecture. *In Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology* (pp. 401-407). ACM.

Mark, R., Richard, M., and David, C. (2009). *Java Message Service* (2nd ed.). United States of America: O'Reilly Media.

Matjaz, J., Ramesh, L., Poornachandra, S., and Frank, J. (2007). *SOA Approach to Integration*. BIRMINGHAM - MUMBAI: packt.

Meenu, R., Amit, K. C., and Shalini, B. Web Service Choreography Description Language (WS-CDL): Goals and Benefits Retrieved Oct 7, 2012, from http://www.rimtengg.com/coit2007/proceedings/pdfs/49.pdf

Milanovic, N., and Malek, M. (2004). Current solutions for Web service composition. *Internet Computing, IEEE,* ***8***(6), 51-59.

Mubarak, M. (2011). *A Glimpse into Islamic Banking Software Systems*: islamic finance news.

Pan, X., Pan, W., and Cong, X. (2010, April). SOA-based enterprise application integration. In *Computer Engineering and Technology (ICCET)*, *2010 2nd International Conference on* (Vol. 7, pp. V7-564). IEEE.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *Computer*,*40*(11), 38-45.

Pervez, S., Imran, A., and Fatima, J. (2008). *Handbook of Islamic Banking Products & Services*: Islamic Banking Department State Bank of Pakistan

PolarLake. *Understanding the ESB*: PolarLake.

Raid, A., Hassan, A., and Hassan, Q. (2008). Leveraging SOA in Banking Systems Integration. *Journal of Applied Economic Sciences, 3*(2), 4.

Raid, A., Hassan, A., and Hassan, Q. (2010). Design of SOA-based Grid Computing with Enterprise Service Bus. *International Journal on Advances in Information Sciences and Service Sciences, 2*(1), 11.

Raymond, K., and TechniCom Group. (2007). Service-oriented Architecture (SOA) defined. Retrieved Aug 2, 2012, from http://www.cad-portal.com/articles/article_images/82/SOA%20description.pdf

Ricky, E. S. (2007). Using Ada in a Service-Oriented Architecture. Paper presented at the SIGAda

Sabooniha, N., Toohey, D., and Lee, K. (2012, August). An evaluation of hospital information systems integration approaches. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 498-504). ACM.

Rosenberg, D. (2010). Modeling Service-Oriented Architectures

Roshen, W. (2009). *SOA-Based Enterprise Integration*: McGraw-Hill.

Sarev, V. G. (2007). *Process and realization of SOA centralized system.* University of Sofia.

Schaik, D. V. (2001). Islamic Banking. *The Arab Bank Review, 3*(1), 8.

Sindhgatta, R., Sengupta, B., and Ponnalagu, K. (2009). Measuring the quality of service oriented design. *Service-Oriented Computing*, 485-499.

Solms, F., and Gruner, S. (2012). What is Software Architecture?.

Shen, W., Hao, Q., and Xue, Y. (2012). A loosely coupled system integration approach for decision support in facility management and maintenance. [doi: 10.1016/j.autcon.2012.04.003]. *Automation in Construction,* 25(0), 41-48.

Steve. (2012, May 18 2012). WHY SOA MATTERS. Retrieved Jul 31, 2012, from http://accelerance.com/why-soa-matters/

Tijs, R., and Jos, D. (2009). *Open Source ESBs in Action*. United States of America: Manning.

Votis, K., Alexakos, C., Vassiliadis, B., and Likothanassis, S. (2008). An ontologically principled service-oriented architecture for managing distributed e-government nodes. *Journal of Network and Computer Applications*, *31*(2), 131-148.

W3C. (2007a). SOAP Version 1.2. Retrieved Aug 3, 2012, from http://www.w3.org/TR/soap12-part1/

W3C. (2007b). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Retrieved Aug 3, 2012, from http://www.w3.org/TR/wsdl20/

W3C Group. (2004). Web Services Architecture. Retrieved Aug 3, 2012, from http://www.w3.org/TR/ws-arch/#introduction