Lampiran I :
Full Text Jurnal yang sudah dipublikasi
di jurnal internasional
Modeling of Business Process Management of Academic Affairs Information System

Wiranto Harry Utomo
Department of Information System
Sanja Wacana Christian University
Diponegoro 52-60, Salatiga, Indonesia

Theophilus Welfirm
Department of Information System
Sanja Wacana Christian University
Onorogono 52-60, Salatiga, Indonesia

ABSTRACT
Misalignment between business process and information system is common in many organizations. Typically, the efforts to correct this misalignment do not result in positive outcomes because of two factors: (1) the complexity of the information technology architecture derived from heterogeneous applications built on different architectural programming languages, web platforms, and (2) the degree to which applications should be kept running when updated. In order to link business process and information system, both should be integrated. The objective of this paper is to present the business process of a university's information System for Academic Affairs. 2.2 analyze and design an information system of business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method. The purpose of this paper is to present the business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method. The purpose of this paper is to present the business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method. The purpose of this paper is to present the business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method. The purpose of this paper is to present the business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method. The purpose of this paper is to present the business process of a university's information System for Academic Affairs. In order to identify the services needed for the automation and integration of business process and information system. The next step is the integration method to be used in this research is a combination of Service-oriented Architecture (SOA) and Model-view Controller (MVC) using the commercial method.
1. Point-to-point Integration Architecture [7]

were not aligned with the overall business objectives. Because the information systems were built using a mix of software tools and were not connected to any common business functions or applications.

In this way, there were many integration points that were not able to support the business functions that were captured by the application.

One of the key reasons for this was the lack of a common integration platform that could connect the various systems. This lack of standardization made it difficult for the business to effectively manage and maintain the various systems.

3. SERVICE-ORIENTED ARCHITECTURE

A Service-Oriented Architecture (SOA) is a design approach that focuses on the creation of software services that can be composed into applications. This approach is based on the idea that software services should be developed as independent, reusable components that can be combined to create new applications.

The key benefits of a SOA include:

- Lower development costs
- Faster time to market
- Increased flexibility and scalability
- Improved software quality
- Reduced maintenance costs
- Enhanced interoperability

The main components of a SOA include:

1. Services: These are the basic building blocks of a SOA. Services are typically small, self-contained units of functionality that can be easily integrated into applications.
2. Service Orchestration: This is the process of connecting and coordinating services to create a complete solution.
3. Service Orchestration Language (SOL): This is a language used to describe how services should be orchestrated.

The main advantage of a SOA is that it allows organizations to build applications that are more flexible and scalable. This approach also makes it easier to integrate new services and to change the way that existing services work.

In summary, a SOA is a powerful approach for building modern applications that can be easily scaled and adapted to new business needs. By focusing on the development of services, organizations can build applications that are more robust and easier to maintain.

References:


While the hybrid-and-pole integration model makes use of lightweight connectors to integrate applications through a middleware infrastructure, it is not the only alternative to the SOA-MA method. Some vendors offer proprietary integration models that deviate from the SOA-MA standard, which may lead to compatibility issues and vendor lock-in. However, SOA-MA has gained significant traction in the industry due to its flexibility and potential for innovation.
This research uses the combination of SIA and SMDA methods [3]. The combination of these methods is based on the thinking that each method in itself is not the optimum in integrating services. SIA must continue to provide the service analysis and identification, but a lack of specific guidelines to implement services found. SMDA method requires complexity in the transformation of the required process model to the level model (model) which is in an additional, but without guidelines to ensure of finding the best for each individual SIA and SMDA methods two combining with other advantages will be gained, SMDA provides an environment that reduces complexity in the services stage and integrates all kinds of technology, protocol, and application between SIA in a model in high level business. Proposed model transformation visualization independently, and use in programming code. The combination of SIA and SMDA methods will be a complete method for enterprise integration.

5. RESULTS AND DISCUSSION

An experimental study is presented in section 4. The SIA-SMDA method consists of various tasks: Phase 1: namely Smaniac software model, the Core Model, Model, Information Layer Model, and User Model. Smaniac Model. Conceptual Model. ANS Model. NAI Model. NAII.

5.1 Identification of current Business Processes


5.2 Business Process Automation for Academic Affairs

Using SMDA is a method in which academic affairs businesses are implemented in the Smaniax Software model. SMDA's can be defined as an automating these 13 business processes can reduce the number of business processes to only two processes for the first important stage 11. New model registration 5. Academic registration 6. Business Study.

5.3. New Student Registration

New student registration model can be described in a case diagram (see Figure 6). This diagram will result in the output diagram in each case 1. Smaniax system processing to the business process for new students registration, 2. Despite the business process, diagram will be added to the function of the model system. New student case diagram consists of four cases: New Student Form, New Student Information System, and Business Study.
5.3.3 Identification of Service candidate

In this study, the identification of service candidates was performed at the registration stage, in the business service layer, and at the service layer. The identification process is based on the needs that have been obtained from the application model used. The Service candidate in Business Service layer identifies the service candidate based on the existing business process. In the service layer, the service candidate is identified based on the business process and service candidate. The identification of service candidates is based on the needs that have been obtained from the application model used.
Fig 9: Final Service Catalogue

3.2.1 Service candidate in Application Service Layer
Service candidate is the application service that is the source need to be the basis to execute the service in the new SOA-based integration business process. This study is known that there is one service to support new strategy, new information management in database. This service can be included in the business process as is a technical business related to the database applications. This service or service layer can be separated from the service layer in application layer at the domain management level.

3.2.2 Service candidate in Orchestrating Service Layer
Service candidates of orchestrating layer are the services that performed the application of service that exist at the lower layer (Business service layer and application service layer). From the existing business process it was found that the most business process is expanded and applied to the new business process which then to service in the orchestrating layer is needed to develop the services from business service layer and application service layer. This service is involved in the application layer.

3.4 Final Service catalogue
The service candidates are combined to produce service candidates as shown in Figure 9. The 14 services that have been found will be implemented as service layer in application layer in the domain management level. Services from these services will be used in the next SOA Model (Service Model, Composite Model, WSDL Model, BPEL Model, and Composite Model of the SOA Model).

6. CONCLUSION
Using SOA-MSM method, it is a system architecture that illustrates processes that currently implemented in Swiss Weather Christian University (SICU) can be adapted. Automating some 13 business processes can reduce the amount of the business processes in only five processes. The service oriented services can be made to support Web services in aggregate of 35 unstructured information systems in SICU such as reasonably identified.

7. REFERENCES
Scenarios on Electronic Commerce and Security. Published by the IEEE Computer Society.


Lampiran 2:
Draft Artikel Jurnal yang akan dipublish di jurnal internasional (Sudah mendapat Acceptance Letter)
Journal of Theoretical and Applied Information Technology <editor@jatit.com>

to: wranto.uksw@gmail.com, wranto.utomo@staff.uksw.edu, ermanwillem@gmail.com

Dear Corresponding Author Wiranto Herry Utomo

We are pleased to inform you that your paper titled "THE IMPLEMENTATION OF ENTERPRISE SERVICE BUS (ESB) IN GRADUATION BUSINESS PROCESS INTEGRATION" Paper ID: 20262 - JATIT has been accepted for publication in peer-reviewed and indexed (SCOPUS, EBSCO, IDEAL, ULPI, DRJ, IAEG, JPI, JIPN, INAR, Capp, CERES, IJCC, NIT, MCI, SIAI, MOJAC, AVS, JATIT, JIPRI, ISPT) JOURNAL OF THEORETICAL AND APPLIED INFORMATION TECHNOLOGY (ISSN: 1817-3195, ISBN: 1993-6465). The acceptance decision was based on the internal and external reviewers' evaluation and editorial board's approval.

You have to proceed with submission of publication paper registration fee ($250) via credit card transaction through our online payment system (Use any valid credit card of Yourself/Friend/Family etc.). Please submit the dues on our basis submission system at:

http://store.kpi.edu.pk/jatit/index.php

so that your paper may be published in upcoming issues. From this information, please make the receipt number generated after completing the payment process so that we can easily track your payment. The billing invoice will be posted on your own statement as kpi etc., panchayat california usa. (Any Authentic Credit Card of Yourself/Friend/Family etc. can be submitted voluntarily use).

You can submit the publication registration fee via wire transfer directly to the following bank account (US $250 publication payment + $10 bank charges):

- Bank Account Title: Journal Of Theoretical And Applied Information Technology
- IBAN: PKH4CSCH00001144565801
- SWIFT Code: SCBLPKXL
- Bank Name: Standard Chartered Bank Ltd - Market Branch - Islamabad, Pakistan.
- JATIT Address: Flat No. 5 Block No. 17 Capital Sector I-1, Islamabad, Pakistan.

Forward the receipt id after making the payments so that the transaction be traced. You may also contact otherwise if you want to opt for payment from western-union or moneypngn etc.

Kindly also submit a camera ready copy (CRC) with updates as per evaluation in Ms Word document and journal (two columns) format (attach with this acceptance letter) to address publisher@jatit.org "Mr. Shazia" after registration fee submission for limited available slots allocation in the upcoming Vol 60 issues of JATIT to be assigned on first paper registration payment basis. JRC copies can be submitted at a later time after slot reservation. A certificate of publication can also be provided on demand after submission publication if required earlier than publication time for use in official purposes.

12/11
We congratulate you on acceptance of paper and shall encourage more quality submissions from you and your colleagues in future.

Regards,

Shahtadi Ghayyar
Co-Chief Editor

Editorial office
Journal of Theoretical and Applied Information Technology
The enclosed manuscript is under consideration for the above-mentioned journal. Please provide feedback on the following criteria so that further process may be initiated:

<table>
<thead>
<tr>
<th>Mark where appropriate</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it a research or review paper?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is it within the scope of the journal?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is it a full paper submission?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is the language of paper English?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Are the English and syntax of the submission good?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Will the paper be of interest to the audience?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Has the paper or part of it already been published elsewhere? [Based on Google Search on the paper abstract]</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Recommendations: Mark where appropriate.

| Accepted After Internal Review and Recommended for External Technical Review | X   |
The enclosed manuscript is under consideration for the above-mentioned journal. Please provide a comment on the following criteria. Please be advised that you should provide comments within a month of receiving the manuscript. Reviews should be returned to editor@JATIT@gmail.com / editor@JATIT.org as an attached file.

<table>
<thead>
<tr>
<th>Mark (X) where appropriate</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the title accurately reflect the content?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is the abstract sufficiently concise and informative?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Do the keywords provide adequate index entries for the paper?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Is the purpose of the paper clearly stated in the introduction?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Does the paper achieve its declared purpose?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Does the paper show clarity of presentation?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Do the figures and tables aid the clarity of the paper?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Are the English and syntax of the paper satisfactory?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Is the paper concise? (If not, please indicate which parts might be cut?)</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Does the paper develop a logical argument or a theme?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Do the conclusions sensibly follow from the work that is reported?</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Question</td>
<td>Yes/No</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------------------------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Are the references authoritative and representative?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is the paper interesting or relevant for an international audience?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is there a valuable connection to previously published research in this area?</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Is the overall quality suitable for inclusion in this journal?</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Recommendations: Mark where appropriate.

<table>
<thead>
<tr>
<th>Recommendation</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Definitely publishable. Accept without correction or minor corrections</td>
<td>X</td>
</tr>
<tr>
<td>Publishable, however accept subject to changes</td>
<td></td>
</tr>
<tr>
<td>Reject due to major changes but encourage resubmitting</td>
<td></td>
</tr>
<tr>
<td>Reject due to unpublished material</td>
<td></td>
</tr>
</tbody>
</table>

Additional Comments for author:

Confidential Comments to Editor:

Paper has my approval as it is fine by review standard. You may forward acceptance letter.
THE IMPLEMENTATION OF ENTERPRISE SERVICE BUS (ESB) IN GRADUATION BUSINESS PROCESS INTEGRATION

WIRANTO HERRY UTOMO, THEOPHILUS ERIAN

Information System Department, Satawa Wacana Christian University
Information System Department, Satawa Wacana Christian University
E-mail: wirantoherryutomo@gmail.com, eriantheophilus@gmail.com

ABSTRACT

ESB is one of SOA pillars beside WS and BPEL. ESB is an infrastructure for SOA service connection and message exchange. The main function of ESB is routing, protocol and messaging of data transformation, protocol and message transformation function in ESB help to deal with protocol and data discrepancy. ESB uses connection mediation and the reuse of service components. It also simplifies integration that will then improve integration scalability. This research has successfully integrated graduation business process that involves more than 80 services including faculty, library, dormitory, administration and academic bureau. Student Affairs Business Finance, and laboratory. This integration of graduation business process uses ESB as integration middleware. Web services integration reveal the role of ESB in routing and secure and protocol transformation.

Keywords: SOA, EAI, BPEL, integration, and service

1. INTRODUCTION

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

The inharmonious 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: the complexity of technology information architecture, from a heterogenous application build from different architecture, programming language, and platform 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 (Joric, 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’s only uses traditional approach. The first architectural approach to help solving the issue that is related to this system integration is SOA. According to Shihabuddin (2014), SOA is a solution to harmonize information technology and business processes. Adopting SOA will lead to uniformity in information technology system. According to Shihabuddin (2014), SOA can also lead to the improvement in the use of resources outside the company.

In addition, according to Joric 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 intra-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 business process 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 been 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 involves some units, such as Facilities, Library, Dormitories, Administration, and Academic Library. Model in Altair-Berlin, Finance Department, and Laboratories will be developed. This graduation business process uses ENS as integration middleware (See Figure 1).

![Figure 1: Graduation Business Process Integration](image)

**2. INTEGRATION ARCHITECTURE**

According to Figure 1 (2011), there are four integration architectures: 1) point-to-point, 2) hub-and-spoke, 3) enterprise service bus (ESB) and 4) ESB/SAO. 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 service bus architecture, independent systems are integrated using a message bus. SOA-based integration architecture uses service 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 need N(N-1)/2 interfaces. A 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 it is to modify and maintain the application.

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 mediation 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 a front hub. It will reduce the chaos in the integration setup. The weakness in hub-and-spoke architecture lies in hub's centralized character. If hub experiences failure, the whole integration will fall as well. Besides, hub and spoke's ownership and integration technology is locked by vendor.

At 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 more flexible 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 do this 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 data transformation. Through the function of protocol transformation and message in ESB, protocol discrepancy can be overcome. ESB also increases 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 attraction level within business process.
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 solvency, reduce time to market, and provide support for global expansion. SOA substantially impacts the whole architecture of enterprises. Business services provided by SOA forms the foundation of business and process architecture. SOA forms the infrastructure that can be exposed as services that can be divided and used. Business processes, services, and events are converted into appropriate applications services. Business and support service architecture. Services used application architecture, as an information infrastructure, is achieved through data standardization and availability through a service provider (Woo-Matching, 2011)

SOA is a software architecture that supports service oriented principles (Fuji, 2005). Service: 2006; Karchevsky et al. 2008; Mciak, 2008; Reay et al. 2009; Jansen et al. 2010). Service-oriented architecture is a concept in software engineering representing different approaches to separate interests.

It means that system functionality is divided into smaller logical and called service. The services are independent, but they have the ability to interact with each other through a particular communication mechanism. Therefore, E.1. (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 requestor and service provider. Service requestor is a service that calls other services, whereas the requested service is called service provider.

In addition to SOA definition according to E.1., there are more similar SOA definitions from other sources. Sometimes this SOA definition is called WS framework that is general can be seen in Figure 2.

This WS framework is a technology framework standardized and mapped into SOA model as follows: a) Services realized as WS, b) Message is described by SOAP protocol, c) Description is prescribed by WSDL, d) This in 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 (E.1., 2005). The service oriented concept make sense 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 participate in an orchestration. Therefore, there are some problems that must be processed by services, they are how services are connected, how services communicate, how services are designed, and how message services message is defined by Edi(2015).

As has been stated in background of the problem, SOA Delivery concepts according to Edi(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. Juni (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 SOA. ESB is required to connect to some information technology (IT) services. ESB must be flexible to integrate, and remit components according to the changes of business needs. ESB creates a technology that improves the bond that it provides the ability to integrate system into SOA and deploys gradually (Juni, 2007; Addy-Jaye, 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 based 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 (Juni, 2007; Addy-Jaye, 2010).

ESB general architecture with connected components can be seen in Figure 3. Components can play a role as other service producer or service consumer. Services can be in any form of virtual 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 (Juni, 2007; Addy-Jaye, 2010).

According to Juni (2007) and Addy-Jaye (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 plugable architecture for connector 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).

![ESB Architecture in General](image)

ESB provides strong, reliable, and safe interface communication infrastructure that can be exploited. ESB also provides communication control over the use of services that include (Juni, 2007): 1) The ability 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, role, 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 image, distribution/deployment, on-the-fly reconfiguration, and etc.

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

5. RESEARCH FINDINGS & DISCUSSION

5.1 Use Case Diagram

This Use Case diagram describes 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 specifications to 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 commerce system, as well as sales. Figure 4 shows the need of graduation information system.

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.

![Image](image-url)

5.3 Implementation with Glassfish ESB

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 360 GB. The specification of the software used is: Microsoft Winndows XP SP3 operation system, Java SDK Standard Edition 1.6.0, update 26 version, Netbeans IDE 6.7.1, and Glassfish ESB 3.1 (Open ESB)

![Image](image-url)
The component diagram will be implemented using Glassfish ESB in the form of Composite Application stored in file using Appliastic's framework Apinthat can be seen in Figure 7.

Figure 7 Implementation Result in the form of Composite Application

Figure 7 is the implementation of Composite Diagram into Composite Application run on 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 of applications. This problem can be solved using indirect connections through application traffic using ESB that provides facilities to route content or context based message.

Besides, there are two heterogeneity problems. The first is the discrepancy of using 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 heterogeneity problem is related to the discrepancies 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 another XML based format.

The three cases of ESB in terms of routing, protocol transformation, and message/data transformation has been performed successfully in the research. JBI module developed 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 converts HTTP inter-protocol communication using JDIC, between HTTP and SMTP, or between HTTP and FILE. Message transformation can be done through XML data format.

7. CONCLUSION

Graduating business process involving facilities, Library, Dormitory, Administration and Academic Bureau, Student Affairs Bureau, Finance Department, and Laboratory has been performed successfully. This graduating 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.
REFERENCES


Ludwig, H. and Rabi, F., 2009, Service Oriented Computing in Practice – An Agenda for Researchers into the Features Influencing the Organizational Adoption of Service Oriented


ACKNOWLEDGEMENT

We thank DTFM DIKTI who has funded the research and allowed the publication of the this research finding journal.