Analysis Service Engineering Development Lifecycle: an Object-Oriented Approach

Ade Hodijah

Computer Engineering and Informatics, Bandung State of Polytechnic (POLBAN) Jl. Geger Kalong Hilir, Ds. Ciwaruga, Parongpong, Kab. Bandung Barat, Jawa Barat 40012
e-mail: adehodijah@jtk.polban.ac.id

Abstract. Service Engineering (SE) is defined as a set of activities introducing a new business service. An object-oriented methodology is defined as principles and procedures to develop object-oriented software. This paper proposes an object-oriented methodology embedded SE framework in re-defining business service as a service-oriented application with detail implementation by using Activities and Artifacts. The proposed model for SE lifecycle consists of Process, Notation, and Tools modeled by utilizing Unified Process (UP), Business Process Model and Notation (BPMN), a programming language that supports the Business Process Management System (BPMS). In this research, the explanation is organized into three parts: (1) the brief review of the existing SE framework, (2) the elaboration of the proposed model, and (3) the detail of the SE lifecycle model with survey results from case studies. The framework is proposed to help practitioners and researchers to conduct service-oriented application by employing principles and methodology offered by the object-oriented approach.

1. Introduction
SE represents software engineering process at a higher level of abstraction. If the main purpose of software engineering is a software system, the main purpose of SE is business service system [1]. Business service in this research refers to a common definition of a service as business activity provided by a service provider to service consumer in order to create value for the customer [2]. There are three main factors to make an organization get more value than other competitors, which are a business organization, product or services produced, and ongoing business process organization [3].

There is a method to evaluate and improve business process, called Business Process Management (BPM) [3]. The use of BPM method in this research defines an SE framework into two workflows process of UP disciplines. Those workflows are Business Modeling of a current business process (As-Is) and implemented business process (To-Be). Based on the result of Requirement analysis for a solution of information system or application. BPM is applied on the stage of the SE framework.

The purpose of the SE framework is to provide service in a layer of information and technology. Particularly, in the form of web services can automate the interaction between services. Hence, the enterprise architecture has to be redefined using the approach of Service Oriented Architecture (SOA). The SOA approach can integrate business process and information technology infrastructure. Analysis and design of service are derived from the organization’s business process [4].

In SOA development requires a methodology, such as Service Oriented Unified Process (SOUP) from [5]. SOUP includes the complete stage of software engineering lifecycle for SOA development side (SOUP for Initial SOA Deployment) until the maintenance side (SOUP for Ongoing SOA...
Management). However, SOUP already explained the documentation, but it still leaves room for adaptation. Hence, this research gives a practical approach for SOA development by using a framework of SE [1].

A comprehensive SE framework for delivering business service as service-oriented application based on SOA methodology has been proposed by Suhardi et al. [6]. The SE framework consists of 8 activities grouped into three stages: (1) identification, (2) design, and (3) prototyping.

![SE framework](adapted from [6])

The main analysis tool proposed in Identification stage that is the Business Model Canvas (BMC), then in Design stage uses Service Blueprinting. The use of BMC and Service Blueprinting aim to replace the process of service analysis and service design from SOA methodologies.

There are several technical platforms introduced in Prototyping stage along with proposed technology that is Microsoft’s Windows Communication Foundation (WCF) [6]. To achieve the goal within the proposed lifecycle model in detailing Activities and Artifacts, Bizagi technology is recommended due to its rapid process automation. Bizagi support BPM and SOA with real-time analytics [7]. The BPM component consists of process modeling, process execution, and process monitoring [8]. Specifically, the SOA component is in the form of web service technology [1].

2. The Proposed Model

The idea of this research is to use True Requirement Investigation from Identification phase of SE framework [1] by using Business Modeling and Requirements phase of UP [9] as a prior workflows process for the design process. The Business Modeling tool uses flowchart as business process of As-Is. The Requirements uses basic analysis activities [10] to analyze the priority components which consists of: (1) list of problems, (2) list of directives, and (3) list of opportunity.

![Identification stage](adapted from [1], [9])

Furthermore, BPM is used in Identification Analysis by applying analysis value added to business service as service-oriented Idea [3].
Service Blueprinting can be used to describe service-oriented as business process of To-Be designed in SOA approached by using BPM notation (BPMN) model [8].

Bizagi as BPM system (BPMS) makes the implementation of immediate business requirements by providing monitor and continuously improve the business process [7]. In this research, for the Prototyping stage is only a Develop process which is performed.

Finally, an object-oriented methodology from UP [9] is used in detailing Activities and Artifacts with BPMN as a modeling language.

3. Results and Discussion
There are three phases of UP that are used in this research which are (1) inception, (2) elaboration, and (3) construction. Each phase consists of five workflows process: (1) business modeling, (2) requirements, (3) analysis & design, (4) implementation, (5) test.

| UP Phase   | Activity                | Artifact                                      |
|------------|-------------------------|-----------------------------------------------|
| Inception  | Analysis problem domain | Domain model (class diagram without method)   |
|            | Analysis software       | Use case diagram                              |
| Elaboration| Analysis problem domain | Revision of domain model (class diagram without method) |
|            | Analysis software       | Revision of use case diagram                  |
|            |                         | Use case scenario                             |
|            |                         | Sequence diagram                              |
|            |                         | Class diagram (analysis)                      |
|            | Design software         | User interface / UI                           |
|            |                         | Subsystem                                     |
|            |                         | Deployment diagram                            |
Details of activities and artifacts from the table 1 show that the stages of software development have not been prepared for web service composition [6]. The software based on SOA consists of two parts: (1) a set of services as service provider and (2) client applications as service consumer [12]. Hence, the additional activities required for defining the service that is service analysis and service design with reference to SOA lifecycle [4]. In this research, the SOA lifecycle consists of: (1) service-oriented analysis, (2) service-oriented design, (3) service development, (4) service testing.

| SOA Lifecycle       | Activities                                      | Artifacts                                                                 |
|---------------------|-------------------------------------------------|---------------------------------------------------------------------------|
| Service-oriented analysis [S1] | Determine scope (problem domain)                | - Business process of legacy system                                      |
|                     |                                                 | - Business requirement automation                                        |
| Service-oriented design [S2] | Determine technology implementation             | - Service candidate model (entity-centric, task-centric)                |
|                     |                                                 | - Service design for SOA layer (application, business, orchestration)     |
| Service development [S3] | Implementation of service                      | - Service source code for each SOA layer                                 |
|                     |                                                 | - Web service catalog                                                    |
| Service testing [S4]  | Test / evaluate the service                     | - Test plan for each web service in client side based on a design of the business process. Tools to test are: (1) unit test, (2) integration test, (3) system/service test, and (4) user acceptance test [1]. |
|                     |                                                 | - Test evaluation                                                        |

Notation for modeling Artifacts from table 2 uses BPMN with BPEL as a standard language for defining and executing BPM [7]. Technically, BPEL as a language that can be compiled by an application supporting orchestration of SOA layer [4]. BPMS role in this research is to support the technology needs of BPMN and BPEL. The following figure is the proposed lifecycle model for SE based on UP methodology.
Table 3. The proposed lifecycle model in detail

| UP  | SOA | Activities                        | Artifacts                                                                 |
|-----|-----|-----------------------------------|----------------------------------------------------------------------------|
| U1  | S1  | Analysis problem domain [A1]      | - Domain model (class diagram without method)                             |
|     |     | Analysis business process (As-Is) [A2] | - Legacy system with glossary - Business process As-Is - Business specification: scope, software, hardware - Service automation candidate |
|     |     | Analysis application requirements [A3] | - Use case diagram - Use case scenario / supplementary specification - Vision |
|     |     | Analysis problem domain [A1]      | - Revision of domain model (class diagram without method) - Revision of legacy system with glossary - Revision of business specification: scope, software, hardware - Business process To-Be with value added |
|     |     | Analysis service model (To-Be) [A4] | - Revision of service automation candidate - Service candidate model (entity centric, task centric) |
|     |     | Analysis client software [A5]     | - Revision of use case diagram - Revision of use case scenario - Revision of vision - Sequence diagram - Class diagram (Process Data ini Bizagi) |
| U2  | S2  | Design service [A6]              | - Service candidate and technology - Service operation - Deployment diagram for service - Business process To-Be in BPMN for service (Model the Process in Bizagi) - User interface for service (Forms Creation in Bizagi) |
|     |     | Design client software [A7]      | - Subsystem - Deployment diagram for client software - Business process To-Be in BPMN for client software (Model the Process in Bizagi) - Revision of class diagram (Process Data ini Bizagi) - User interface for client software (Forms Creation in Bizagi) |
|     |     | Design service architecture [A8] | - Infrastructure design for desktop/web/mobile application |
| U2  | S3  | Implementation [A9]              | - Technology specification: software, hardware - Source code: model, controller, UI (Business Rules in Bizagi) |
| U2  | S4  | Testing integration [A10]        | - Test plan: unit test, integration test, system/service test.              |
| U3  | S2  | Analysis service model (To-Be) [A4] | - Revision of service candidate model (entity centric, task centric) |
From table 3, the proposed lifecycle model is below (I-Inception; E-Elaboration; C-Construction; s-start; r-refine):

| Activity | Artifact | I | E | C |
|----------|----------|---|---|---|
| A1       | Domain model (class diagram without method) | s | r |   |
|          | Business process To-Be with value added |   |   |   |
| A2       | Legacy system with glossary | s | r |   |
|          | Business process As-Is | r |   |   |
|          | Business specification: scope, software, hardware | s | r |   |
|          | Service automation candidate |   |   | r |
| A3, A5   | Use case diagram | s | r |   |
|          | Use case scenario / supplementary specification | s | r |   |
|          | Vision | s | r |   |
| A4       | Service candidate model (entity centric, task centric) | s | r |   |
| A5       | Sequence diagram | s | r |   |
| A5, A6, A7 | Class diagram (Process Data in Bizagi) | s | r |   |
| A6       | Service candidate and technology | s | r |   |
|          | Service operation | s | r |   |
|          | Deployment diagram for service | s | r |   |

Table 4. Iteration of the proposed lifecycle model
### 4. Case Studies

The first case study was performed before a definitive formal framework was devised. The goal was to provide an integrated e-government implementation [13]. The case study was started with a strategic analysis of the EKPPD (Evaluasi Kinerja Penyelenggaraan Pemerintahan Daerah) with BMC. Service blueprinting tool was only used lightly to specify the data dissemination service based on a request from SKPD (Satuan Kinerja Penyelenggaraan Pemerintahan Daerah). SOA method was employed to produce three layers of service: task-centric, entity-centric, and application utility. The service interface is described by using WSDL.

The second case study was done after the first draft of the framework was formalized. The case study employed SE based on an object-oriented methodology of room usage permit in POLBAN [14]. The work produced two BPM diagrams: as-is business process to describe existing situation and to-be business process to propose the new service based on Porter’s value chain analysis in performing true requirement investigation. Modeling notation was complemented with basic UML diagrams: Use Case, Class Diagram, and Sequence Diagram.

The third case study also select the process of room usage permit in POLBAN, but focusing on the design of proposed external data service for the agency as an external user. The work was also done by following the completed and formalized framework. BPMN diagram and Bizagi were used as BPMS in supporting the simplicity of service analysis as SE Identification, service design as SE Design, and service implementation as Prototyping of SE.

The results from the proposed framework utilization in the third case study were taught in a Proyek 3 Diploma degree class for a semester. At the end of the semester, there are three questions of a survey that was conducted to 32 students of the class to assess the simplicity of using the proposed framework. The response are given under 5-scale likert respond, from “strongly agree” to “strongly disagree”. The questions and their results are presented in the following charts.

|   | Business process To-Be in BPMN for service (Model the Process in Bizagi) |   |
|---|-------------------------------------------------------------------------|---|
|   | User interface for service (Forms Creation in Bizagi)                    |   |
| A6| Subsystem                                                                | s |
|   | Deploymen diagram for client software                                    | s |
| A7| Business process To-Be in BPMN for client software (Model the Process in Bizagi) | s |
|   | User interface for client software (Forms Creation in Bizagi)            | s |
| A8| Infrastructure design for desktop/web/mobile application                 | s |
| A9| Technology specification: software, hardware                             | s |
|   | Source code: model, controller, UI (Business Rules in Bizagi)            | s |
| A10| Test plan: unit, system                                                  | s |
The results from figure 6 show that more than 83% of the respondents agree that they able to adopt the proposed framework regarding the explicitness of activities and artifacts in implementing service-orientation application.

5. Conclusion
Alternative software development for adopting SOA lifecycle within SE framework is proposed here along with several trial case studies. The proposed framework is emphasized on the simplicity of the object-oriented approach by using specific tools, notation, and process for each activity and artifact of introducing a new business service.

There is technology to support business service as a service-oriented application, Bizagi as Tools. The technology is used to ensure the proposed service analysis is implementable by using BPMN as Notation for service design process. The analysis and design of service will continue to be improved implementation in each UP phase as Process.

Acknowledgments
Alhamdulillah, acknowledgment is delivered to all lecturers majoring in computer engineering and to all staff of UPPM Polytechnic State of Bandung who has given the opportunity to carry out “Penelitian Dosen Pemula 2016”.

Reference
[1] Suhardi, Robin Doss, and Purnomo Yustianto, Service Engineering Based on Service Oriented Architecture Methodology, TELKOMNIKA, vol. 13, no. 4, December 2015.
[2] J Cardoso, K Voigt, and M Winkler, Service engineering for the internet of services, Enterprise Information System, pp. 15-27, 2009.
[3] Harmon, Paul. Business Process Change, A Guide for Business Managers and BPM and Six Sigma Professional. 2nd Edition. Burlington, USA. Morgan Kaufmann Publishers is an imprint of Elsevier. 2007. [Online].
[4] Erl, Thomas. Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, 2005.
[5] Mittal, Kunal. *Service Oriented Unified Process (SOUP)*. 2016. [Online].

[6] Suhardi, Pathya Budhiputra, and Purnomo Yustianto, *Service Engineering Framework: A Simple Approach*, in International Conference of Information Technology System and Innovation (ICITSI), Bandung-Bali. 2014.

[7] Bizagi Web Services (SOA Layer). [Online].

[8] Bajwa, Imran S. *SOA Embedded in BPM: A High-Level View of Object Oriented Paradigm*. 2011.

[9] Larman, Craig. *Applying UML And Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, 2nd Edition*. 2004.

[10] Jeffrey Whitten and Lonnie Bentley. *System Analysis and Design Methods*. New York, USA: McGraw-Hill, Inc. 2007.

[11] Desenta, Salma. *Development of a Configuration and Monitoring Tools for Software Project with Unified Process Methodology*. Undergraduate thesis. STEI-ITB. 2006.

[12] MATEI, Gheorghe. *SOA and BPM, a Partnership for Successful Organizations*. Informatica Economica. 2011.

[13] Hodijah, Ade. *Perancangan Sistem Portal Evaluasi Kinerja Penyelenggaraan Pemerintahan Daerah (EKPPD), Studi Kasus: Pemerintahan Kota Cimahi*. Thesis STEI-ITB. 2012.

[14] Hodijah, Ade, Joe Lian Min, dan Nurjannah Syakrani. *Rancang Bangun Aplikasi REST Web Service Berbasis Service Oriented Architecture Menggunakan Metodologi Unified Process*. Prosiding 7th IRWNS Politeknik Negeri Bandung. 2016.