

ZEF Framework for e-Government Applications: Featuring SOA and BPM Alignment

Eko K. Budiardjo, Ahmad Nurul Fajar, and Zainal A. Hasibuan

Abstract—A change is a challenge since it is unpredictable and can happen anytime. The enterprise systems have to be responsive to changes. The change will cause alteration in the application and/or existing database if it is not supported by flexible system and adaptable environment. Service Oriented Architecture (SOA) support a mechanism for reusability and seamless integration. In order to anticipate the changes and to improve information services, Business Process Management (BPM) and Service Oriented Architecture (SOA) need to be synergized. Some previous researches have synergized BPM and SOA to endorsed the creation of flexible enterprise system environment, but lack in detailing how to determine and manage commonality and variability features in application software. This Study is illustrated in the Indonesian e-government Enterprise System. Since Indonesia has central and local governments, there are some commonality and specificity of rules and policies. Beside that, there are similarities and dissimilarities of rules and policies among local governments. Hence, to optimize the possibility of intersection and interrelated among common and specific components are needed. Those intersection and interrelated components are integrated in Zuma's e-government Framework (ZEF). This framework consists of four layers: legacy layer, component layer, technical layer and business layer. The layers guaranteed the alignment of rules and policies among central and local governments in the e-government enterprise initiatives.

Index Terms—Component, SOA, BPM, e-Government framework

I. INTRODUCTION

Service Oriented Architecture (SOA) appears by offering something adaptive and reactive to the environment as well as offering solutions to the complexity in business process, system diversity and technology [1], [2]. SOA embraces bottom-up architectural approach while BPM embraces top-down process approach [3]-[5]. If SOA is applied in an organization without involving BPM, reusable and reliable services which can be created are still unable to increase the speed further and optimization [3]-[5].

BPM provides a high level of abstraction to define business processes as well as supervise and regulate the process [4], [5]. SOA enables many services to be combined and support the creation of flexible environment and agile organizations [3],[4]. BPM is a dynamic process of optimization and adaptation, while SOA is a mechanism to create an agile service and provide a mean to regulate the whole service [4],[5]. The critical is to identify and determined components, which are suitable to the change. Sometimes, there is no attention for the business environment

and the change of users need. The synergy between BPM and SOA is expected to adapt to the dynamic change of users need. This can minimize the gap between business experts and IT experts [5]. E-Government on cloud computing and SOA can be considered as such environment [6]. Business model for IT resources and can provide service using remotely-accessible over the internet called Cloud Computing [6],[7]. In next section, we describe related works on the alignment of SOA and BPM.

II. RELATED WORKS

Several related researches have studied the combination of BPM and SOA in achieving the realization of a flexible and dynamic organization [3]-[5], [8]-[10]. BPM-SOA combination leads to the continuity of the business domain and IT by providing more efficient and faster ways to develop IT-based organizations. SOA architecture embraces a *bottom-up* approach while BPM embraces the *top-down* process approach [4], [5], [7]. If the implementation of an enterprise SOA without involving BPM, *reusable* and *reliable* service can be created, but this framework cannot improve the speed further [4], [5]. The service will not be able to have the capability for continuous improvement and optimization. Whilst, BPM is basically cannot be measured and cannot adapt to the environment [4], [5]. The capacity for services monitoring and optimization will be reduced. Hence, if a framework is only based on BPM and lack of SOA characteristics, there will be a capacity reduction [4],[5]. The integration between SOA and BPM should start at least from the design phase to the deployment phase [5].

The transition from *design* to *deployment* phase is a crucial step to achieve a high level of flexibility in the process of the system making of business process models which have been designed [5]. To achieve this objective, automating the process of transition is a way to reduce the *effort* required for a system development model of existing business processes [5]. The results from the different perspectives of business and IT for a business process create a higher cost and time [5]. SOA is considered as a strong partner for BPM. It provides a mechanism so that an organization is able to be more flexible and responsive to change in business processes [4], [5], [7]. The business processes are likely to be dynamic due to environment (social, politics, economy, etc.) changes [4], [5], [7]. The consolidation-relation concept of BPM and SOA creates agile business. BPM and SOA offer a perfect combination for enterprise computing [8]. The combination of BPM and SOA will provide benefits for IT professionals and government officers, business users, etc., [4]-[6], [10]. Such combination can be depicted into a layered architecture.

The architecture combination SOA & BPM consists of a

business services layer, a business layer, an application services layer [6]. The architecture combination SOA & BPM consists of layering such as business process, service, application, technology layer [7]. The architecture combination SOA & BPM consists of layering such as legacy application, implementation service, service assembly, services wrapper, a business service, business process workflow and GUI component layer [4]. Multi model-driven collaborative development platform of service-oriented supports the business plan from the top down and a bottom-up development service [10]. This platform has 3 views: in order to support business and technical activities [10].

The time and the cost of the implementation and execution can be reduced by aligning SOA and BPM [4],[5]. For public sectors, such as government services, E-Government could be implemented based on SOA and Cloud Computing [6]. For academic, a Service Oriented Information System For Collaborative Research and Doctoral Education [11]. Since Indonesian government has some uniqueness in her administration, hence the combination of SOA and BPM is worth to explore. The Indonesian Government has central and local government. There are commonalities and specificity of rule and policies.

III. THE INDONESIAN E-GOVERNMENT SYSTEM AS AN EXPERIMENTAL OBJECT

Indonesia has central government and local government. Local government consists of 33 provinces and 497 districts, which has common as well as specific business processes, policies, and rules. The Indonesian e-Government system has some commonalities and variabilities features [12], [13]. There are much functionality that covered in functionality government group. Based on the Indonesian e-governments blueprint [14],[15], the hierarchy on e-government function are divide into (a) block of functions; (b) block of sub-functions; and block of modules, as depicted on Figure 1.

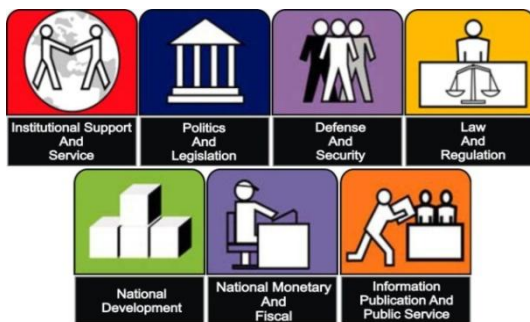


Fig. 1. The Applications Group [14]

Local government has 4 functional group areas: government, region, citizenship, and infrastructure. Local information system has generic and specific functional. The examples of generic applications are human resource information system, finance information system, budgeting information system, civil servant information system. There are developed based on rules and policies from central government. The examples of specific applications are: geographic information system, local tax and retribution information system, remuneration information system.

Besides that, each district in Indonesian government may have several different agencies. Specific applications are developed based on local government rules and policies, and may not necessarily refer to central government. On the other hand, the hierarchy in regulations and policies from the central to the local government is applied.

IV. THE ANALYSIS OF INDONESIAN GOVERNMENT FUNCTION GROUP

Based on the hierarchy policies, leveling of Indonesian E-Government System consists of Function Group, Sub Function Group and Module:

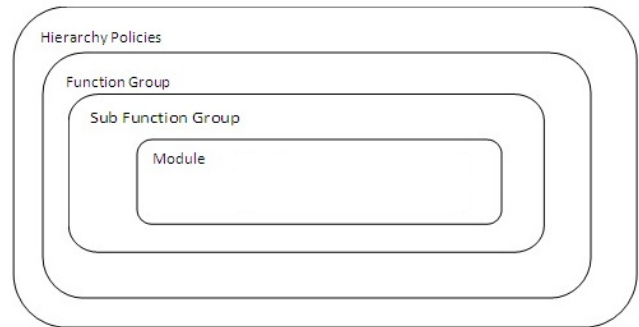


Fig. 2. Indonesian E-Government Function Group

Hollistic approach to the Indonesian government system, we define some related notation, as follows:

- $IGS = \{FG_1, \dots, FG_7\}$

IGS (Indonesian e-Government System) as an organization have set of 1...7 Function Groups (FG) area.

- $FG = \{SF_1, \dots, SF_{18}\}$

Function Group Area in IGS consists of 1...18 sub-functions (SF).

- $SF = \{Md_1, \dots, Md_{70}\}$

Sub Function in IGS consists of 1...70 modules (MD)

According to laws and policies, IGS has some number of function group in which breakdown into Sub Functions and Modules, as seen on Table I:

TABLE I: FUNCTION GROUP

| Function Group | Number of Sub Function | Number of Module |
|----------------------|------------------------|------------------|
| Support and Services | 4 | 15 |
| Politic and Legal | 2 | 5 |
| Defense and Safety | 2 | 6 |
| Law and Policies | 2 | 6 |
| Economic | 2 | 6 |
| National Development | 4 | 25 |
| Publication | 2 | 7 |

Looking down into those 70 modules, agility and reuseability become significantly concern. The need of an agile system in dynamic environment to respond functional requirement changes become crucial. Software are developed with high commonality between different users, however software are always customized for specific user needs [16]. Software Product Line Engineering (SPLE) consists of domain engineering and application engineering. It can

capture commonality and variability to improve traditional reusability. SPLE can exploit commonalities among related products in order to reduce cost and time of software development. The commonalities are used to create a product platform that can be used as a common baseline for all products within a product family [16]. Since, SPLE consists of domain engineering and application engineering. It can support to build a robust platform and build specific user applications [17]. The benefit of SPLE compare to traditional reuse is on maintenance phase [17].

The criteria to identify common and specific modules are user, feature, attribute, interface form, report, and business processes. Module in IGS has similarity among some users. The e-government application users consist of local government, central government, agencies and districts. Since each module related to its features, attributes, interface forms, reports, and business processes. We point out that among those modules, they are not only intersection but also union.

TABLE II: CRITERIA FOR IDENTIFYING COMMON AND SPECIFIC MODULES

| Criteria | Common | Specific |
|--------------------|--------|----------|
| User | ✓ | ✓ |
| Feature | ✓ | ✓ |
| Attribute | ✓ | ✓ |
| Interface Form | ✓ | ✓ |
| Report | ✓ | ✓ |
| Business Processes | ✓ | ✓ |

TABLE III : THE EXAMPLE OF COMMON AND SPECIFIC MODULE CLASSIFY BY USERS

| Module | Common | Specific | User |
|---------------------|--------|----------|---------|
| Civilization | ✓ | | Agency |
| Tax&Retribution | ✓ | | Local |
| Register&Permission | ✓ | | Local |
| Investment | | ✓ | Local |
| Local Election | | ✓ | Local |
| Catalogoe of law | ✓ | | Central |
| Planning&Developmen | | ✓ | Local |
| E-Procurement | ✓ | | Central |
| E-Budgeting | | ✓ | Local |
| Payroll | ✓ | | Central |
| Assesment | ✓ | | Central |
| Local Finance | | ✓ | Local |
| Tourism | | ✓ | Agency |
| Health | | ✓ | Agency |
| Education | | ✓ | Agency |

Those 70 Modules, as describe earlier, are used by ministry, non ministry, province, district and agency. According to the Indonesian law and policy, it has intersection and union. By considering central and local government roles, we have identify some common (C) and specific (C). As an example we present 15 modules out of those 70, where the user is local government, central government and agency, as seen on Table III.

The Case in our study is Budgeting Module. These module consists of 4 sub modules, include: planning, realization, monitoring, and evaluation (Table IV).

TABLE IV : COMMON AND SPECIFIC MODULE IN BUDGETING APPLICATION

| Feature | Common | Specific |
|--------------------|--------|----------|
| Manage Program | ✓ | |
| Submit | ✓ | |
| Approval | ✓ | |
| Reject | ✓ | |
| View & Report | | ✓ |
| Calculate | | ✓ |
| Manage TOR | | ✓ |
| Manage Transaction | | ✓ |
| Manage Indicator | ✓ | |
| Manage Input | ✓ | |
| Manage Output | ✓ | |
| Manage Outcome | ✓ | |
| Manage Indicator | ✓ | |
| Manage Impact | ✓ | |
| Formula | | ✓ |

Such vertical and horizontal relationships permit the uniformity and locality of e-government solutions to provide e-services. In order to integrate the vertical and horizontal relationship of e-government service components, we proposed Zuma's e-government Framework (ZEF). The name is inspired from the Zuma's game that can flexibly find new pattern and also Zuma means peace in Arabic language. As Indonesian's Government regularly and accidentally keep changing her policies, the capacity to maintain peace and adaptable to new formation are needed. The change will cause alteration in application and existing database if it is not supported with flexible system and environment. ZEF guaranteed to support such case.

V. ZUMA'S E-GOVERNMENT FRAMEWORK

There are several attempts to develop framework to combined SOA and BPM. According to Wojciech et al [9], two main trends related to e-government are constant development infrastructure computer and constant increase of user's skill and knowledge of internet. According to Imran et al [4], system architecture that aligned SOA and BPM has not been covered component layer. They split SOA and BPM at different layer. Sebastian et al [5] have combined SOA and BPM, but focus for transition between process design and deployment. Gopala et al [8] have synergized SOA and BPM, but it is not contribute to commonality of government administration. Haitham et al [9] focus for total quality management in production system based on alignment of SOA and BPM. Wojciech et al [9] have contributed for e-government solution based on cloud computing and SOA, but is not synergize SOA and BPM.

So far, there has not been any framework that synergize SOA and BPM for e-Government System. ZEF Framework could be solution for Indonesian government that has the commonality and specificity. It makes possible for intersection between reused, common and specific services

components. It manages reused, common and specific services components based on SOA and BPM alignment approach. ZEF framework also can provide database, business process, and application as a service. The ZEF framework consists of four layers: legacy layer, component layer, technical layer and business layer. The Legacy layer is a layer which consists of existing database, business process and application. The component layer is an intelligence layer which is responsible for clustering the components based on central and local government policies. This layer can provides component that can be complementary as well as substitution for other components based on existing environment. This layer contain reuse component, specific component, common component and substitute component. The technical layer is responsible for integration and orchestration. The business layer is responsible for monitoring and optimizing. In general, reusability is based on commonality. In case of central and local Indonesian governments, it possible the intersection among common and specific components that can be reused.

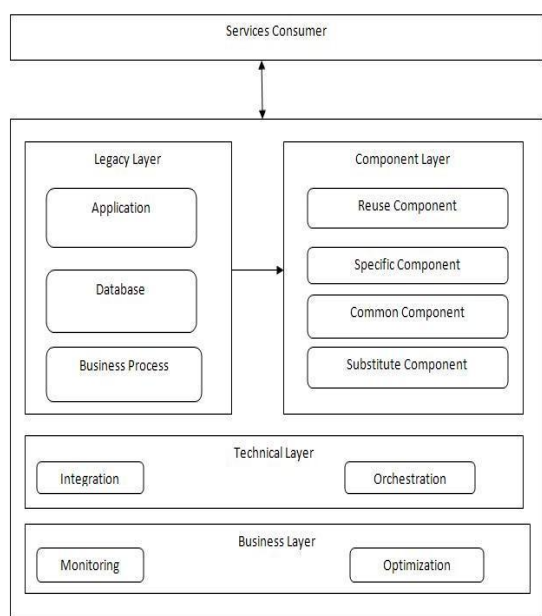


Fig. 3. ZEF Framework

Hierarchy of policies and rules from central government to local government has its commonality. Whilst, the local government may have specific characteristic which differentiate one local government to another. describes of how the central government related to the local government can determine reused, common, and specific components. The component identification from legacy environment is a base line for clustering. Clustering can be achieved from policies, rules and business processes commonality using SOA & BPM alignment approach. The ZEF framework is described in Figure 3.

ZEF framework supports alignment of SOA and BPM. It can also support dynamic changes in the central and local governments in which they have similar cases within the Indonesian government. Requirement is unpredictable and dynamic change is continuing. Thus, the challenge is to embed the components one to another. A component can be a complementary as well as substitution for other components.

The service provider should provide common, specific, and reused components which suit for 33 provinces and 497 districts. It has generic and specific functionalities based on policies and rules. The components here means are Feature.

Figure 4 describes how the citizens, the business, and the government are interrelated to access the services. The services are classified into three application groups, i.e. public-oriented, business-oriented, and government-oriented. The functions that are common and specific to government are the baseline for application group. Application group can be accessed by citizen, business and government. ZEF framework supports service-oriented application in e-Government System. Citizen, business and government may access the portal public as well as internal e-government. In order to anticipate the flaw, the system will provide suitable component without changing legacy application over all.

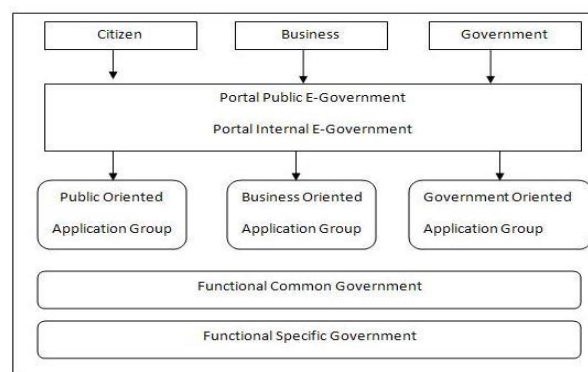


Fig. 4. Service Oriented Application Group

ZEF framework developed based on three stages (1) The stage of component identification. This stage aims to identify candidate components. It observes policies, rules, business processes, applications, data in the central and local government legacy system, (2) The stage of determining reused, common, and specific components. It aims to determine reused, common, and specific components using the family of business processes, policies, and rules, and also determine substituted and complemented components, (3) The stage of component composing and sharing. It aims to compose and share the components using orchestration and integration

VI. CONCLUSION

SOA and BPM synergize of both top-down and bottom-up approaches. The synergy can support dynamic changes in the central and local government. The rules and policies from the central to the local government have the commonality. Beside that, they also have their own specific characteristics which differentiate one local government to another. ZEF framework integrates all these components into four layers: legacy layer, component layer, technical and business layer. Each layer will be collaborated and integrated in providing data, business process and application of service. The component layer is an intelligence layer which is responsible for clustering the components based on central and local government policies.

Identifying reused, common, and specific components helps the decision makers and developers to develop the

flexible and adaptable e-government enterprise system environment. This will anticipate dynamic policies changing in the government so that information service will faster and better quality. For further research, we need to test the ZEF Framework with web based application.

ACKNOWLEDGMENT

This paper was fully supported by the Laboratory of Digital Library and Distance Learning (DL2) Faculty of Computer Science, University of Indonesia.

REFERENCES

- [1] M. P. Papazoglou, P. Traverso, Schahram, and Frank, *Service Oriented Computing Research Roadmap*, 2006
- [2] J. L. Zhao, M. Tanniru, and L.-J. Zhang, "Services computing as the foundation of enterprise agility: Overview of recent advances and introduction to the special issue Information Systems Frontiers," *Maret.*, vol. pp. 1-8, September 2007.
- [3] F. Kamoun, *A Roadmap towards the Convergence of Business Process Management and Service Oriented Architecture*, 2007
- [4] I. S. Bajwa, R. Kazmi, S. Mumtaz, M. A. Choudhary, and M. Shahid naweed, "SOA and BPM Partnership, A paradigm for Dynamic and Flexible Process and I.T. Management," *World Academy of Science, Engineering and Technology*, April 5, 2008
- [5] S. Adam and J. Doerr, "How to better align BPM & SOA – Ideas on improving the transition between process design and deployment," in *Proc. BPMDS*, 2008
- [6] W. Cellary and S. Strykowski, "e-Government Based on Cloud Computing and Service-Oriented Architecture," in *Proc. ICEGOV2009*, Bogota, Colombia, November 10-13, 2009
- [7] N. Leavitt, "Is cloud computing really ready for prime time?" *IEEE Computer Magazine*, vol. 42, no. 1, pp. 15-20, January 2009.
- [8] G. K. Behara, "BPM and SOA: A Strategic Alliance," *BP Trends*, 2008
- [9] H. A. M. El-Ghareeb, *Aligning Service Oriented Architecture and Business Process Management Systems to Achieve Business Agility*, 2008
- [10] X. Lu, Y. Li, C.-C. Lo, and K. M. Chao, *Service Oriented Development Platform For E-Business*, 2008
- [11] S. Klink, A. Oberwes, D. Ried, and R. Trunko, "A Service oriented information system for collaborative research and doctoral education," in *Proc. the IEEE International Conference On e-Business Engineering (ICEBE 2006)*, 2006
- [12] A. N. Fajar, E. K. Budiardjo, and Z. A. Hasibuan, "SOA and BPM Allignment on ZEF Framework, the Indonesian government case," in *Proc. ICCSIT*, Chengdu, China, 2011
- [13] A. N. Fajar, E. K. Budiardjo, and Z. A. Hasibuan, "System Architecture in Dyanamic Environmet based on Commonality and Variability Business Processes," in *Proc. ICCM*, Seoul, 2012
- [14] Ministry of Communication and information Indonesia, *Design Blue Print E-government Application*, 2006.
- [15] E. Zainal and A. Hasibuan, "e-Gov AMAN framework: a unified Indonesian e-government security," in *Proc. 3rd IEEE International*

Conference on Emergency Management and Management Science, 2012.

- [16] M. Eriksson, "An approach to software product line use case modeling," Thesis, Department of Computing Science, Sweden, 2006
- [17] F. van der linden *et al.*, *A Framework for software product line engineering*.



Dr. Eko K. Budiardjo was born in Jakarta on 1959. He has been the faculty member of the Faculty of Computer Science - University of Indonesia since 1985. Teaching, research, and practical services are aligned; give result in a full spectrum of academic achievement. Majoring in Software Engineering as professional track record, he has made some scientific contribution such as Software Requirement Specification (SRS) patterns representation method, ZEF Framework, and FrontCRM Framework. Graduated from Bandung Institute of Technology (ITB) in 1985, holds Master of Science in Computer Science from the University of New Brunswick – Canada in 1991, and awarded Philosophical Doctor in Computer Science from the University of Indonesia in 2007. Currently he is the Vice Chairman of ICT Technical Committee of The National Research Council (DRN), and Chairman of The Indonesian ICT Profession Society (IPKIN). He is reachable through eko@cs.ui.ac.id



Ahmad Nurul Fajar was born in Jakarta on 1978. Computer Science Doctorate Candidate in Faculty of Computer Science University of Indonesia (UI). Graduated from Gunadarma University major in Informatics in 2001, hold master of Informatics from Bandung Institute of Technology (ITB) in 2004. Currently, he is the Faculty member of Information System Department in UIN Syarif Hidayatullah Jakarta. Teaching, research, and practical services are aligned. Interesting research and teaching in domain are of Software Engineering and information system. He has made scientific contribution such as ZEF Framework and also Team Leader for Face Of Indonesia (FOI). His email is ahmad.nurul@ui.ac.id or nurul.fajar@uinjkt.ac.id



Prof. Zainal A. Hasibuan was born in Pekanbaru, Indonesia on 1959. He received BSc. degree in Statistic from Bogor Institute of Agriculture, Indonesia, 1986. MSc., and PhD., in Information Science, Indiana University, in 1989 and 1995 respectively. Currently, He is Lecturer and PhD supervisor at Faculty of Computer Science, University of Indonesia. He is also Head of Digital Library and Distance Learning. His research interest includes e-Learning, Digital Library, Information Retrieval, Information System, and software Engineering. His email is zhasibua@cs.ui.ac.id.