

DESIGNING AND DELIVERING PUBLIC SERVICES ON THE CLOUD

Yehia Taher¹, Rafiqul Haque², Dinh Khoa Nguyen¹ and Willem-Jan van den Heuvel¹

¹European Research Institute in Service Science (ERISS), Tilburg University, Tilburg, The Netherlands
{Y.Taher, D.K.Nguyen, wjheuvel}@TilburgUniversity.edu

²Irish Software Engineering Institute (Lero), University of Limerick, Limerick, Ireland

Keywords: e-Government, Public services, Reusability, Customization, Cloud computing, T-Shaped platform.

Abstract: Cost and complexity are currently the most substantial obstacles for designing and delivering services in the public sector. The traditional in-house development and maintenance landscape of public services require experts from diverse domains, various technologies and complex on-premise infrastructure, etc. The high upfront cost and complexity impede the proliferation of Information Technology (IT) within the domain of public sector. It is the aim of this research project to deliver a cloud based platform that allows non IT-experts to customize prefabricated and reusable public services by parameterizing them. This customization revolves around reference guidelines that accommodate a methodology in a consistent manner.

1 INTRODUCTION

Recently, IT has undergone vigorous development to facilitate organizations to develop and deliver services in more efficient and effective fashion. In this setting, public service organizations are harnessing the power of IT and distributed technologies (e.g., Internet) to automate their desk-based human provided services and also to make services ubiquitous. The ubiquity ensures that public services are accessible by the users (e.g., citizen) via the Internet. Despite enormous amelioration of technologies, public service organizations have failed to exploit the *full* potentials of IT. Our analysis reveals three salient reasons behind this including economical, technological, and societal. In this article, we focus on economical and technological. To be precise, this research deals with two primary issues embodying *cost* and *complexity* to design and deliver public services.

Traditionally, the in-house development landscape includes a group of service development experts from multiple domains (e.g., business, technology), on-premise infrastructure, technical specialists, and experts for maintenance of applications and infrastructure. Evidently, such a landscape augments cost and complexity enormously that is not affordable and manageable by a large number of public service organizations.

The above considerations bring the widely known concept called *reusability* into the light of this research. Reusability enables to reuse existing elements that could fit into specific requirements. It is a promising concept for developing public services that do not require to be developed from scratch. This helps to diminish the development cost for public services. However, the only shortcoming of reusable services is their *fitness to specific requirements* because they are still generic (*i.e., independent of any specific context*). Hence, it is highly unlikely that generic services can completely satisfy the requirements of a specific context. This implies that reusable services require customizations. In order to customize context-independent services to context specific ones, organizations require experts from various specific contexts (*e.g. service usage context, service runtime environment, etc.*) and technologies (*e.g., WS-* standards, REST, etc.*). In this setting, the reusability concept turns out to be an effective means to reduce the service development cost *partially* but not *completely*. In addition, the licensing cost and the cost of infrastructure increase the upfront cost heavily that is beyond the ability of many public service organizations to afford. Therefore, there is a strong requirement for a solution approach that will serve as a means of developing public services in cost effective manner.

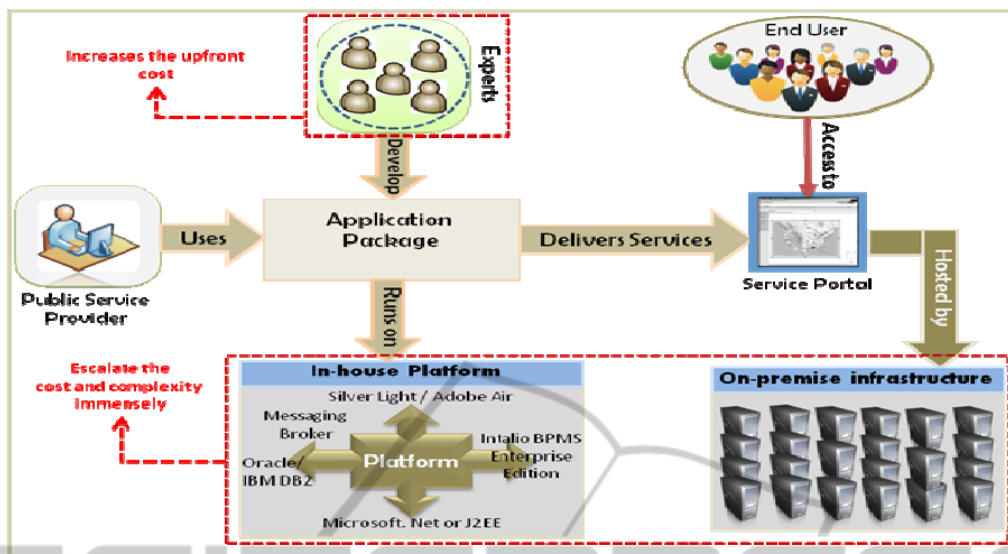


Figure 1: The traditional landscape of developing and delivering services.

Complexity is another grand challenge for public service organizations to exploit the potentialities of IT. Large numbers of organizations today are striving to remove the excessive complexity from the service development landscape. Many organizations consume more time and effort on maintaining their infrastructure instead of focusing on adding value on services. Complexity may underlie the entire infrastructure. The reason behind this complexity is merely the integration of various technologies. It turns the service development landscape into a space only for experts who possess sound knowledge on development platforms. In addition, maintaining complex infrastructure is a non-trivial task which requires a highly proficient team of experts. These complexities preclude public service organization to adopt IT for delivering services to the end-users.

This paper aims at developing a cloud based platform that will bolster the public service organizations to develop and deliver public services in efficient and (cost-) effective manner. We named the platform *T-Shaped* platform. The T-Shaped is grounded on the concept of reusability, a methodology for customizing reusable processes, and most importantly the cloud computing paradigm. This cloud based solution will facilitate migrating the excessive complexity that arises on account of in-house service development infrastructure. This will encourage much wider adoption of IT within public service domain, promote the development of innovative public services, and reduce the time to deliver services to customer (e.g., citizen). The proposed platform

includes guideline that will enable the public service organizations to develop and deliver services without requiring experts. This implies, public service organizations will be able to reduce the service development cost significantly since they will be able to lower their budget on experts. Additionally, the pay-per-use economic model will help reducing the cost of infrastructure and licensing fees for technologies. In a word, the cloud based T-Shaped platform will lessen the overhead cost of designing and delivering public services.

This paper is organized as follows, a motivating scenario is presented in section 2 to illustrate the cost and complexity underlie in today's solution architecture; section 3 describes the core concept of our proposed cloud-based platform for public service organizations; section 4 describes the methodology that facilitates the public service customization using the T-Shaped platform. We explain the related literatures in section 5 and the future extensions of this research in section 6.

2 MOTIVATING SCENARIO

We present a motivating scenario in this section. Figure 1 shows the scenario. The scenario demonstrates various development costs that exist in traditional development landscape. It also reflects the complexities that are underlying the traditional application development ecosystem.

Figure 1 demonstrates several areas of traditional in-house service development ecosystem, which

increases the cost massively. The red-dotted rectangles in figure 1 spot these areas. In the given scenario, a public service provider develops services that run on *in-house platform*. This requires (at least) a team of developers lead by a specialist, analysts, solution designer, and solution architect. Hiring these experts increases the upfront development cost exponentially. Besides, the platform is integration of various technologies that also raise the cost immensely. The reason is merely the licensing cost of enterprise editions of these technologies.

Additionally, the public service providers host the services on *on-premise infrastructure* that causes a considerable escalation of the total cost. The infrastructure entails various equipments in particular, processing equipments (e.g., CPU, memory), storage equipments (e.g., database storage), and networking equipments (e.g., switch). In addition, a group of technical specialists need to assemble these equipments in an infrastructure. The costs of these resources (equipments and human) increases the total cost heavily that is far-affordable to many public service organizations.

Furthermore, in-house service development platform can be enormously complex. The platforms today integrate several technologies that raise complexity. The classic examples are Eclipse and Netbeans platforms that are largely complex to install and operate for a non-expert user. Developing applications or services using these platforms are far beyond their ability. Figure 1 shows that a traditional development platform integrates different technologies including Silverlight, Intalio BPMS, and so on. These technologies make the platform infrastructure highly complex that is not easy to use for users who are not adequately expert. Developing services using this platform by a non-expert user is merely impossible. In addition, there are no clear guidelines that can ease the service development complexity.

Now, the above issues foster one important question: *what is the most suitable approach that optimizes the total cost and reduces the complexity of developing and maintaining public services?* We initiate this research to find answers of these questions.

3 T-SHAPED PLATFORM AS A SERVICE: CONCEPTUAL OVERVIEW

Adopting the cloud paradigm, we offer the T-Shaped

platform as a service for public service organization to design and deliver services in a cost-effective manner. The *T-Shaped* solution aims at providing a virtual development platform for the public service administrators to streamline the public service design, delivery and management processes on the cloud.

In this section we describe the fundamental concept of the T-Shaped platform. The T-Shaped platform consists of two different views: *Horizontal View* and *Vertical View*. For service design, the horizontal view of the *T-Shaped* platform proposes that public service administrators can reduce the transparent cost by exploiting a number of generic *Reusable Services* in the public service domain together with a *Reference Guideline* for customizing these services. The *connector* of T-Shaped platform connects a provider to a public service repository where providers can query and find reusable services. The T-Shaped platform embeds a reference guideline that underpins the customization of reusable services.

Following the vertical view of the *T-Shaped* platform, the users may use the reference guideline to customize the generic public services as they want. The reference guideline serves as the guiding principle for public administrators or service providers accommodating the customization of generic services without the need of having intense knowledge on processes as well as its related technologies. It contains a large set of parameters derived from diverse domains that are important to the customization of public services. T-shaped allows public administrators to use these parameters to customize the process-based services. The key idea of the reference guideline is to bolster *stratified customization* which allows fine-tuning a context independent service to context specific one that represents the unique interests or characteristics of an organization. Figure 2 depicts the stratified customization approach with an example.

Stratified customization is an approach that promotes customization of context independent services to context specific ones in a multiplicity of 1 to n.

The meta-reference model in figure 2 is a reusable process model that associates services containing the elements from global perspective that is independent of usage and context. The customization produces the reference model containing the elements and characteristics for specific usage but not context and finally the solution model that is the concrete solution to a specific context. The example in figure 2

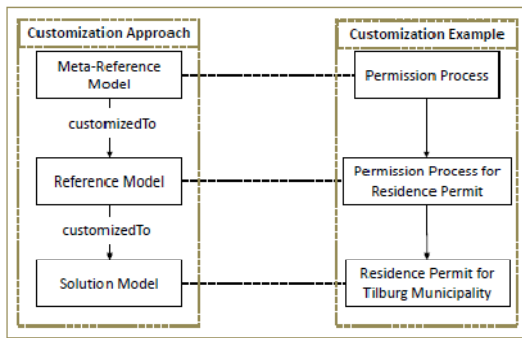


Figure 2: The stratified customization approach with example.

demonstrates the customization process. We provide the example linked with customization approach using dotted lines. The meta-reference model is the permission process associating permission services. The process is customized to permission process for residence permit which is not yet a concrete solution. The final customization produces the concrete solution which is residence permit for Tilburg Municipality. We discuss the methodology for customizing services in section 4.

Developing services in-house may require much work in addition to the actual service logic development. This work may include the procurement, installation, configuration, and operation of platforms, servers, storage, and networks as well as the provision of database services and user-authentication mechanisms. Providing these portions of the development process as cloud platform services is expected not only to reduce the development costs but also to ease the work load of the public administrations while improving the service value they provide. Inspired by this idea, our *T-shaped* solution will be delivered to the public administrations as a virtual development and hosting cloud platform service.

Conventional on-premise service development in the public service domain, as explained in section 1, is unacceptable due to high complexity and development cost. Moving to our virtual platform helps to delegate all the complexity of the development lifecycle of public services to our platform. In particular, this enables public administrations to better concentrate their efforts on understanding the citizen requirements and preferences which - without any doubt - leads to more adaptive public services to the citizen ever-increasing dynamic needs. We believe that this approach helps much for creating public services that continue to have a significant positive impact on the citizen's life. Furthermore, public

administrations, when using our T-shaped solution platform, may be charged based on the pay-per-use or subscription schemes with a variety of predefined pricing models and agreements. That gives them the possibility for considering the most economic way of developing their public services.

In the next section, we describe the methodology of our cloud-based T-shaped platform for public services.

4 METHODOLOGY

The T-Shaped underlies a methodology for customizing reusable public services. We propose this methodology to facilitate stratified customization of processes. The methodology is influenced by the research works from (Ma & Leymann, 2008), (Heuvel & Jeusfeld, 2007). It includes three phases that are described in the followings:

(i) Discover

The public service provider finds the required services from the service market or service cloud. The T-Shaped solution connects a provider with the service cloud.

To support finding and loading of services from a public service cloud, the following primitives need to be supported:

- *Find*: The provider finds the service from the service cloud using this primitive.
- *Load*: This primitive is used to import a service in T-Shaped platform to customize to be used in specific context.

Figure 3 shows that a public service provider find and load public services from the service cloud.

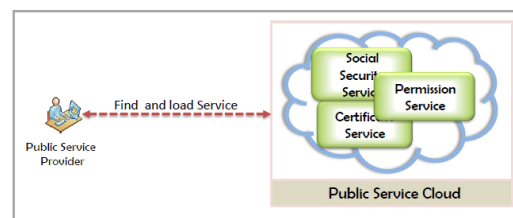


Figure 3: Discovering public services from the cloud service.

(ii) (Re)-Design

After the service is loaded on T-Shaped platform, a service provider analyzes the loaded service. The

analysis fosters re-designing the service to satisfy specific interests. Comparing with specific requirements, the provider selects the list of activities that should be *refined*, *pruned*, and *aggregated*. The service may need extra functionality to be *extended* to satisfy the context specificity. To support public service providers, we integrate the following primitives with the T-Shaped platform:

- **Refine:** Refine allows refining an activity into sub-activities. For instance, *prepare permission application* is an activity may be refined to *prepare document* and *fill application*.
- **Prune:** An activity can be removed from the process using this primitive.
- **Aggregate:** Aggregation allows combining two or more activities into single activity.
- **Extend:** This primitive used to extend the required functionality.

Additionally, activities can be *renamed* to fit into a specific context. Renaming is labelling the activities, actors, events of the generic process.

Figure 4 shows how T-Shaped solution facilitates a service provider redesigning a generic process using the operators.

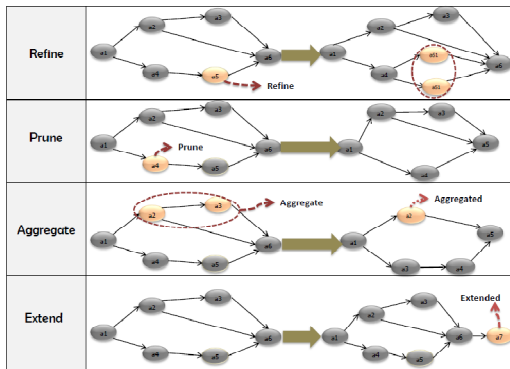


Figure 4: Redesigning a Generic Process using T-Shaped Solution Operators.

(iii) Parameterize

After the service is scoped to a specific interest through redesigning the underlying process, this phase supports parameterization of the service. Parameterization allows specifying the quality of services (QoS), policies, and security parameters. It also allows specifying the values of parameters. Thus, parameterization is highly critical to reach the goal of service and also to ensure the user satisfaction.

In this regard, T-Shaped provides an idealsolution, a reference guideline that delivers a

large set of parameters from diverse domains (e.g., regulatory policy) to support service provider to parameterize services. The providers will be able to select and specify the parameters without having much expertise. Figure 5 shows an example of parameterization of a service.

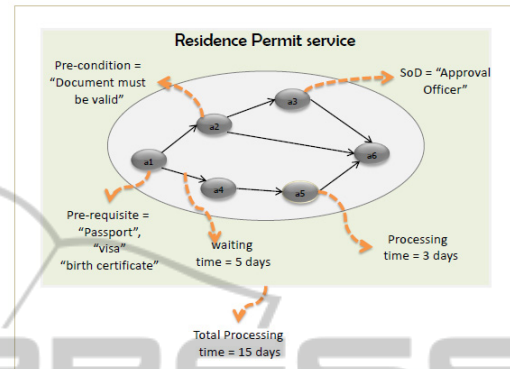


Figure 5: Parameterization of Public Services.

5 RELATED WORK

This research revolves around three key concepts cloud computing, reusability, and customization. In recent days, cloud based solution for developing applications is gaining enormous popularity with a high number of market-available platforms. (Heroku, 2010) was one of the first cloud platform providers, which delivers the whole lifecycle management solution for Ruby & Rails application over the Internet. Another prominent example in cloud-based development platform is the Force.com (Salesforce.com, 2010) that allows designers to build their CRM applications faster and with lower cost using a very simplified programming model. Instead of providing a whole virtual development platform like Force.com, (Google App Engine, 2010) and (Microsoft-Windows Azure, 2010) provide SDKs that support a simulated development and testing environment.

The existing cloud-based solutions available in the market may largely contribute to reducing the upfront cost of acquiring in-house development platforms and the complexities that underlie the traditional service development ecosystem. However, these solutions are only provided as generic, domain-independent solutions as they mainly target a large number of clients. Yet in the public service domain, expert cost regarding developing services is still a question since public service organizations may require a high number of experts that have adequate domain knowledge. Our

approach is certainly exceptional in this regard since it particularly focuses on the public service domain.

Furthermore, the concept of reusability of public service processes is heavily documented throughout various bodies of literature. (Ma & Leymann, 2008) proposed a research that highlights fragmenting a complex business process into shards that are intended to be flexible and reusable for future business process modelling. This research work is enormously interesting especially the life-cycle model for business process modelling using reusable fragments. However, the scope of the work does not solve our problem entirely since we focus not only on facilitating reusability but also a robust guideline for process customization, which add value on the top of reusable fragments.

(Curran et al., 1997) initially proposed reusable business processes as an approach for large-scale enterprises. Their work has been cited in an extensive number of research works, yet was criticized by (Mendling et al., 2006) with counterarguments on SAP reference model. (Heuvel & Jausfeld, 2007) proposed a framework with guidelines to transform a model with reference models in particular, the SAP reference model. Strictly speaking, these researches are only conceptually related to our approach but different in terms of applications. In this paper, we narrow down our scope to public service reusability that has not been considered yet. The closest work related to reusable public service has been proposed by (Koussouris et al., 2008), in which the authors presented a modelling view of generic processes. Their main contribution was to support the Public Administrations to achieve resolution of organizational interoperability and systematically address the Homogeneous Service Composition.

Fairly speaking, there is no solution approach available that entails the cross-domain parameters to support multi-modal customization.

6 CONCLUSIONS

The lack of expertise, capital expenditures, and high cost of experts are the predominant factors that severely preclude wider adoption of eGovernance among public service organizations.

In the paper, we have presented a T-Shaped cloud based solution which aims primarily at reducing the costs and enhance the efficiency and effectiveness related to public service design and delivery process. The T-Shaped is expected to influence the public administrations to embrace

electronic governance. The proposed solution is rendered as a cloud based platform focused to alleviate the involvement of public administrations with many aspects ranging from consultation, design, implementation, operation, to maintenance of applications and IT infrastructures. This enables public administrations to better concentrate their efforts on improving the value of the rendered services by relying on the T-Shaped customization facilities.

Our proposal reflects first attempt in defining a foundation for designing and delivering public services on the cloud. As part of our ongoing and future work, we will conduct empirical/experimental study focusing on the security and privacy including data confidentiality aspects of public services delivered on the cloud.

ACKNOWLEDGEMENTS

The work leading to this result is sponsored by the EU FP7 as part of the COCKPIT project.

REFERENCES

- Cabinet Office, (2005). eGovernment Unit: eGovernment Interoperability Framework. Retrieved from: <http://interim.cabinetoffice.gov.uk/govtalk.aspx>
- Curran, T., Keller, G., and Ladd, A., (1997). SAP R/3 Business Blueprint: Understanding the Business Process Reference Model. Enterprise Resource Planning Series. Prentice Hall PTR,
- Google App Engine (2010). Retrieved from <http://code.google.com/appengine/>
- Heuvel, W-J. V. D, Jausfeld, M., (2007): Model Transformations with Reference Models. IESA 2007: 63-75.
- Heroku (2010). Retrieved from <http://heroku.com/>
- Koussouris, S., Tsitsanis, A., Gionis, G. and Psarras, J, (2008). Designing Generic Municipal Services Process Models towards eGovernment Interoperability Infrastructures, eJETA 2008.
- Ma, Z., and Leymann, F., (2008). A Lifecycle Model for Using Process Fragment in Business Process Modelling. Proc. of the BPMDS'08.
- Mendling, J. et al., (2006).
- Mendling, J., Moser, M., Neumann, G., Verbeek, H.M.W., Dongen, B.F.V., and Aalst, W.M.P.V.D. (2006): *Faulty EPCs in the SAP Reference Model*, in Proc. of the BPM 2006, LNCS 4102, pp. 451-457, 2006. Springer-Verlag.
- Microsoft-WindowsAzure (2010). Retrieved from <http://www.microsoft.com/windowsazure/>.
- SalesForce.com (2009). Retrieved from <http://www.salesforce.com/platform/>