Enabling Reuse of Citizen Centric Government Processes through Service Oriented Architecture

by

Bhudeb Chakravarti, Vasudeva Varma

in


Report No: IIIT/TR/2008/45

Centre for Software Engineering Research Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2008
Enabling Reuse of Citizen Centric Government Processes through Service Oriented Architecture

Bhudeb Chakravarti
NISG, International Institute of Information Technology (IIIT), Gachibowli, Hyderabad
bhudebc@ieee.org

Vasudeva Varma
International Institute of Information Technology (IIIT), Gachibowli, Hyderabad
vv@iiit.ac.in

ABSTRACT
It is observed that information technology (IT) is being used to automate the traditional government processes and to achieve easy access of the information. They are developed to increase the efficiency of the government departments and represent their needs. As a result, they achieved a huge inventory of IT applications providing disparate processes implemented using different technologies and platforms. It thus became very important to use IT as an “integrator” allowing disparate processes running in different departments to interconnect. This helps the citizen to access different government services through a common single-point window and helps the government to transform from government-centric organization to citizen-centric organization.

Keywords
Reuse, Reusability, Service Oriented Architecture (SOA), Citizen-centricity.

1. REUSE OF GOVERNMENT SERVICES
It is observed that there is a great deal of commonality in the processes followed by various government departments in delivering the services to the citizen. It is thus possible to develop common processes that can be used across the departments. The services developed based on these common processes can be reused across departments and also across the states providing similar services. One of the primary tasks to enable reuse of government services would be identification of core processes and sub-processes that are repetitive in nature. We also need to develop a common infrastructure to provide these processes through a service oriented framework.

While identifying and designing the key processes for reuse, we should consider the following parameters:
- Simplification of the procedure of transactions by citizens
- Quality of the service provided to citizens
- Secured data transmission and storage
- Easy retrieval and access of information
- Increased efficiency and productivity of employees
- Ease of implementation and use
- Future relevance of the services being provided

To identify the reusable key processes, simple techniques such as root cause analysis, activity based analysis and management can be used. These techniques are very useful in terms of analyzing the important activities that add values to the service delivery.

Once the key processes for reuse are identified, the next step is to define these processes in terms of increasing levels of abstraction. While abstracting the processes, there should be a conscious effort to generalize as we simplify, resulting in the processes applicable to a broader scope and more suitable for reuse.

The evolution of reuse in software design and development is shown in figure 1. In the beginning, the reuse was achieved through sub-routines and functional reuse. Later object oriented programming enabled the reuse through classes and objects. In the mid-90s, component based design took the concept of abstraction into a new direction. In a component-based architecture, the components are organized into layers, most commonly:  presentation, business logic, data management, security and infrastructure.

Copyright is held by the author/owner(s).
ISEC’08, February 19–22, 2008, Hyderabad, India.
ACM 978-1-59593-917-3/08/0002.
To make services reusable, we must provide the services that are difficult to develop and easy to apply and install. So the “register, find, bind and execute” paradigm is a possible way to make more reusable components. With a service registry that is a directory on a distributed or networked environment, we can manage service contracts dynamically [1].

Krueger proposed a framework in 1992 to evaluate the software re-usability in terms of abstraction, selection, specialization and integration [2]. The business logics of the component providing the service can be published in the form of a service in such a way that they can be discovered, matched, composed, executed, verified, and monitored in real time and at runtime.

Successful reuse of services depends on the benefits they provide on improving productivity and quality. Integration of software reuse into each stage of the software development life cycle can provide long-term progress in solving productivity and quality problems and therefore, significantly reduce the cost of developing software [3].

2. **SOA AS AN ENABLER OF REUSE**

In service-oriented architecture (SOA), the services are developed as reusable building blocks for the citizen-centric application development. A service typically consists of a specification that mentions what the service does, an implementation that performs the task the service offers and an interface description that can be implemented through standard interface of the service requestor.

A service is a coarse-grained, discoverable, and self-contained entity that interacts with applications and other services through a loosely coupled, message-based communication model [4]. The loosely coupled architecture of SOA enables services to be accessed using any format. It overrides the syntax and transmits the physical address of the data requested. Service interface is different from the protocols used for data communication. The connection between service consumers and providers is stable, consistent and re-configurable according to the changing requirements. Even under different technical environments, service providers and consumers can be changed without disturbing the definition of the functionalities.

In SOA, identifying and defining meaningful services for external and internal users is essential. The services identified should have clear business concept, objective and requirements. Services are grained on the basis of business processes. Services that are fine-grained can be considered as the core activities like data access function or user interface component. The coarse-grained services are built on the basis of fine-grained services to address specific citizen requirements. Flexibility and reusability of business objects determines the level at which it is grained.

3. **SOA IMPLEMENTATION ROADMAP**

The first step of implementing SOA in government departments is to identify the needs of the citizen, convert the need into citizen-centric service requirements and map the requirements onto the SOA vision. This vision should be communicated to the citizen and the feedback from the citizen should be taken care of. This mechanism will establish cooperation and co-ownership between the government, IT department and the citizen.

The next step should be to create a long-term strategy and convey the same to all the stakeholders. SOA does not represent a quick-fix solution to long-standing challenges prevailing in the service delivery mechanism of government. In order to plan for and accommodate the long-term nature of SOA, we should start building the core infrastructure, skills and fundamental knowledge and change management. This ensures tight management of risks associated with SOA and enables the government departments to learn from the experience and to improve the approach over time.

Once the long-term strategy is planned, it is suggested to create a comprehensive SOA reference architecture. This architecture framework will behave as a standard reference for all the team members involved in the implementation of SOA. The department should create a long-term plan giving the holistic view of the SOA implementation strategy.

Reuse is the driving force behind the adoption of SOA. Most of the values and benefits of SOA manifest over time. Reuse does not happen overnight. But as the developers get more used to the notion of developing and using services, the reuse starts happening. One way to measure reuse is to define a metrics. An SOA management framework can be implemented to monitor real-time runtime behavior so that it can track which services are available, how they are being used and how often they are being called. As a result it will also track performance and quality of services available.

Finally, the government departments should define criteria and metrics that can be used to track and measure the impact of SOA implementation. The success should be measured in terms of improved processes and productivity, improved reuse level and cost savings and improved citizen satisfaction.

4. **CONCLUSION**

Duplication of IT applications providing same or similar services in various government departments is a major cost that can be reduced through reuse of government processes in a systematic manner. The grouping of similar functionality across multiple applications cuts down on overall complexity and provides greater flexibility to process reuse. Functional reuse will be the primary means of ensuring that government departments can respond rapidly and effectively to the citizen needs. SOA, being the main driving force for future functional reuse within the government departments, will enable functional components to achieve scalability through simple utility-style provisioning and resilience through the use of stateless services.

5. **REFERENCES**


