A study on a generic development process for SOA design and implementation

Toshimi MUNEHIRA†

†Director R&D Department
OGIS-RI Co., Ltd, Japan
Munehira_Toshimi@ogis-ri.co.jp

Abstract
In order to optimize annual IT spending and to reduce the complexity of entire system architecture, SOA trials have been started. It is common knowledge that to design an SOA system we have to adopt the top-down approach, but in reality silo systems are being made, so these companies cannot reuse newly designed services, and cannot enjoy SOA’s economic benefits. To prevent this situation, we designed a generic SOA development process referred to as the architecture of “mass customization.” To define the generic detail development processes, we did a case study on an imaginary company. Through the case study, we could define the practical development processes and found this could vastly reduce updating development costs.

Keywords: BPM, Generic Model, Mass Customization

1 Problem on a SOA development
It is common knowledge that to design SOA systems we have to adopt a top-down approach that begins with business modeling (Yoshida, Tanaka, and Une 2007[1], Dugan 2007[2], Yanagisawa and Mutoh 2008[3]). It might be only in Japan, Japanese companies’ SOA adapting cases accommodate this principle, but because of the limited business domain, they might be so-called “silo SOAs.” If these IT professionals continue building silo SOA systems, they will be sure to confront the problem that when interconnecting some SOA systems they find it very difficult because of different service definitions and master data, and as a result they cannot enjoy SOA’s economic benefits.

To use BPM suites to implement SOA systems is one of easy methods, but we are afraid we would be easily locked in by the BPM vendor. Originally, SOA allows us flexibility, but the SOA implementing tool, that is BPM suites, may decrease that flexibility.

To prevent such problems, we tried to develop a generic development process. In this report, we introduce the outline of this method and the results of validation of our SOA designing methodology.

2 Requirements for a SOA

2.1 Differences between the SOA-oriented system and the legacy system

Table 1 shows the differences between the traditional system development approach and the SOA-oriented system development approach (Schmelzer 2007[6]). The orthodox systems are basically designed to last, but the SOA-oriented systems (SOA) are designed to change. This is the fundamental difference. Other items shown in Table 1 arise from this point.

Table 1.Differences between SOA and Legacy

<table>
<thead>
<tr>
<th>Traditional Distributed Approach</th>
<th>Service Oriented Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Designed to last</td>
<td>Designed to change</td>
</tr>
<tr>
<td>TightlyCoupled</td>
<td>Loosely Coupled, Agile and Adaptive</td>
</tr>
<tr>
<td>Integrate Silos</td>
<td>Compose Services</td>
</tr>
<tr>
<td>Code Oriented</td>
<td>Metadata Oriented</td>
</tr>
<tr>
<td>Long development cycle</td>
<td>Interactive and iterative development</td>
</tr>
<tr>
<td>Middleware makes it work</td>
<td>Architecture makes it work</td>
</tr>
<tr>
<td>Favor Homogeneous Technology</td>
<td>Leverage Heterogeneous Technology</td>
</tr>
</tbody>
</table>
2.2 Requirements for SOA

2.2.1 Basic requirements

SOA was developed in response to users’ need to adapt agilely to a change of business circumstances and to reduce system maintenance costs that might become a fixed cost, as well as to fulfill software engineers’ desire to create flexible and stable system architecture (Nomura and Hara 2006[7]).

From the perspective of system development, we can find these common requirements for SOA.
1) Combine instead of coding
2) Immediately change systems according to a business change
3) Users do system modifications

2.3.2 Economical Requirements

Fig. 2 shows the expected cost model of SOA (Schmelzer 2007[6], Munehira and Shimada 2008[5]). One of the important reasons to choose SOA is to reduce maintenance costs. To realize this cost model is the most important requirement for an SOA development model.

![Fig.2 The SOA economics](image)

3 Mass Customization as a reference model

Just as the waterfall development methods originated in the construction industry, we can find the similar solution of SOA in the manufacturing industry. That solution is based on “Mass Customization.

3.1 The definition of Mass Customization

Mass Customization is defined as follows (Pine, Victor, and Boynton 1993[8]). “Mass customization calls for flexibility and quick responsiveness. In an ever-changing environment, people, processes, units, and technology reconfigure to give customers exactly what they want. Managers coordinate independent, capable individuals, and an efficient linkage system is crucial. Result: low-cost, high-quality, customized goods and services.”

This definition has many similarities with those in Table 1.
In “mass customization,” there are these four customization approaches (Gilmore and Pine 1997[9])

**Collaborative:** Collaborative customizers conduct a dialogue with individual customers to help them articulate their needs, to identify the precise offering that fulfills those needs, and to make customized products for them.

**Adaptive:** Adaptive customizers offer one standard, but customizable, product that is designed so that users can alter it themselves.

**Cosmetic:** Cosmetic customizers present a standard product differently to different customers.

**Transparent:** Transparent customizers provide individual customers with unique goods or services without letting them know explicitly that those products and services have been customized for them.

3.2 Requirements for Mass Customization in designing and manufacturing

Feitzinger and Lee pointed out three organizational-design principles for an effective mass-customization program (Feitzinger and Lee 1997[10]).
1) A product should be designed so it consists of independent modules that can be assembled into different forms of the product easily and inexpensively.
2) Manufacturing processes should be designed so that they consist of independent modules that can be moved or rearranged easily to support different distribution-network designs.
3) The supply network—the positioning of inventory and the location, number, and structure of manufacturing and distribution facilities—should be designed to provide two capabilities. First, it must be able to supply the basic product to the facilities performing the customization in a cost-effective manner. Second, it must have the flexibility and the re-
sponsiveness to take individual customers’ orders and deliver the finished, customized goods quickly.

3.3 SaaS, an example of Mass Customization

Salesforce.com typically shows the characteristics of Mass Customization (Munehira 2008[11]).

3.3.1 Customization approach

Salesforce.com adopts 4 customization approaches.

Collaborative: By introducing applied samples, it can clarify prospective customers’ subconscious requirements. Since prospective customers can experience actual working systems, Salesforce.com can very effectively define their requirements.

Adaptive: After taking in a brief lecture, customers can make their own customizations themselves. If they need complicated customizations, such as an interconnection with their own systems, then an integrated development platform will be provided, which allows them to make their own programs.

Cosmetic: Users feel like they are using specially customized software, but Salesforce.com supplies only the usual service.

Transparent: As Salesforce.com adapts to various customers’ needs, it continues updating systems without stopping the services. One day customers suddenly find that they can use new services without any configuration change.

3.3.2 Designing and manufacturing

Using software, Salesforce.com has built a required designing and manufacturing system. It can “turn its processes into modules and create an architecture for linking them that will permit them to integrate rapidly in the best combination or sequence required to tailor products or services.” (Pine, Victor, and Boynton [8])

3.4 Requirements from Mass Customization principles

As the SaaS example shows, mass customization is exactly what we want to realize in SOA. From the software engineering view, we translated Feitzinger and Lee’s three principles into these two modularization requirements.

Modularized business processes and services

Design service systems as compounds of independent modules, and by combining these components, make it possible to provide each service with lower costs and fewer efforts.

Modularized service design and production process

In the design process, let business or system designers design the appropriate combinations of those individual modules, so that the business systems thus designed by business users are implemented as system services without any further business level modifications.

4 The outline of a SOA development process

Fig. 3 shows a SOA development process model we designed to fulfill the requirements described in chapters 2 and 3.

The 1st step is business modeling. According to the company’s strategy (BSC: Balanced Score Card), we design To-Be business processes that would achieve business goals.

The 2nd step is service mapping. Activities in the system swim lanes are detailed into the BCE (Boundary, Control, and Entity) models. Controls are mapped to services stored in the service repository.

The 3rd step is service development. If proper services are not found in the service repository or mapped services lack some functionality, we will produce new services or modify the existing services. In this step, service modification is allowed but replication is strongly forbidden. Replication will seriously decrease the reusability of services.

From step 1 through to step 3, we do data designing using an orthodox method.

The 4th step is system implementation. Screens are designed simultaneously but independently. Designed business process data and related screens are set in a way that a process engine can understand. ESB-related configurations are also set.
Fig.3 the SOA development model

5 Detail processes definition with a case study

In order to define the detail development processes and to evaluate the SOA economics, we did a case study on an imaginary company.

5.1 Preconditions of a case study

5.1.1 About an imaginary company

This imaginary company deals in sports bicycles and bicycle parts. This company has a wholesale department, nation-wide franchise shops, and a Web site, but it does not have any factory. This company imports bicycle parts from all over the world and sells them to bicycle fans through three channels, that is, specialty shops, franchise shops, and the Web site.

5.1.2 OSS-based development environment

The purpose of this study is to design a generic development process. For SOA, we have to design and implement both human processes and system processes. When we considered using COTS to build this environment, we found the following problems.

- It is a de facto standard to adopt BPEL for implementing a system process.
- Because BPEL engines are usually supplied from software vendors coupled with ESB, these engines usually have vendor original specifications. This would force us to implement special functions that work only under specific circumstances.
- BPEL for People is prepared for implementing human processes under BPEL conditions, but few engines are supplied, and it requires high-level technical skills. Therefore, it does not fulfill the requirements written in II-B.
- BPM suites support both processes. However, if we started to use a BPM suite, we would be locked in afterward by this tool.
- Many BPM suites require us to develop screens with this tool.
- The development process is also dependent on this tool. BPM suites support both processes. However, if we started to use a BPM suite, we would be locked in afterward by this tool.

In order to avoid the vendor-lock-in, we prepared OSS-based SOA circumstances for the implementation of this imaginary company’s model.

5.1.3 Development team

For SOA development, these three players are
required. We organized an international team.

- Business Modeler: Osaka
- SOA Designer: Tokyo
- SOA Implementer: Shanghai

5.2 Adopted development processes

5.2.1 Business process modeling

Fig. 4 is the To-Be process of “procure cycle parts designed to achieve the objective, “Shorten the lead time from sales order to purchase order.” The upper activity diagram is the output of the 1st step, and the lower sub-activity diagram is the output of the 2nd step. Screen, Main-process, and DB correspond to BCE (Boundary, Control, and Entity).

The end of a business modeler’s task is to register desirable services in a service repository. In Fig. 4, the ‘T’ mark shows this activity is mapped to some registered service. If a business modeler cannot find a proper service, then he or she creates a new service class in the “Service Repository package” and makes a mapping. After a business modeler finishes the business process modeling, a SOA designer’s task begins.
5.2.2 Service analysis and design

The SOA designer’s tasks are defined as follows (Kranfzig, Banke, and Slama 2004[13], Oba et al. 2005[14]).

1. Service Analysis
   - Service candidate analysis
   - Conceptual model design
   - Service candidate refining
   - Service analysis
   - Message analysis
   - Service protocol analysis

2. Service design
   - Service design
   - Message design
   - Service definition documentation

Firstly we followed these steps, and modified them through the feedback discussion between team members.

5.3 Defined detail development processes

5.3.1 Detail process definition

Through this experience of SOA developments, we could define the generic and practical detail processes as below.

Service Analysis
   - Service Candidate Analysis
   - Service Candidate Optimization
   - Service Analysis
   - Message Analysis
   - Business Process Analysis
   - Service Protocol Analysis
   - Feedback to Business Model

Service Design
   - Message Design
   - Service Design
   - Process Design
   - SOA Infrastructure Architecture Design
   - Service Implementation

Table 3 Sample of the definitions of the generic SOA development process

<table>
<thead>
<tr>
<th>Activity</th>
<th>Subactivity</th>
<th>Details</th>
<th>Role</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Analysis</td>
<td>Service Candidate Analysis</td>
<td>Extract service candidates from these elements</td>
<td>Service Analyst</td>
<td>Business Process Model</td>
<td>Service Candidates (Class Diagram)</td>
</tr>
<tr>
<td></td>
<td>Service Candidate Optimization</td>
<td>- Services from the service repository&lt;br&gt;- Activities from activity diagrams&lt;br&gt;- Business Entities from activity diagrams</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Analysis</td>
<td>Service Candidate Optimization</td>
<td>Redefine service candidates based on relations that are CRUD with business entities</td>
<td>Service Analyst</td>
<td>Service Candidates (Class Diagram)</td>
<td>Optimized Service Candidates (Class Diagram)</td>
</tr>
</tbody>
</table>
6 Validation of economics of SOA

Table 3 shows the difference between the FP (Function Points) based estimation (= Orthodox method) and the real data (= SOA development). By means of this comparison we found the following:

- Initial SOA development cost (1st Step) is almost equal to that of an orthodox method.
- In updating the system (2nd Step), SOA development cost is lower than that of the orthodox method as the Fig.2 suggests.

Table 3 Actual results and estimation

<table>
<thead>
<tr>
<th>Estimation (Orthodox)</th>
<th>FP method</th>
<th>1st-Step</th>
<th>2nd-Step*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-Func</td>
<td>36</td>
<td>20+36*</td>
<td></td>
</tr>
<tr>
<td>Tran-Func</td>
<td>69</td>
<td>37+60</td>
<td></td>
</tr>
<tr>
<td>No-Adjustment</td>
<td>105fpt</td>
<td>57+96fpt</td>
<td></td>
</tr>
<tr>
<td>Translate to man days</td>
<td>141.13</td>
<td>179.18</td>
<td></td>
</tr>
</tbody>
</table>

Actual Results (SOA)

| Analysis               | 10.00     | 20.00    |
| Design                 | 30.75     |          |
| Coding                 | 46.55     | 49.10    |
| Test                   | 49.40     | 55.60    |
| Total (man days)       | 136.70    | 124.70   |

7 Conclusion

SOA has been developed to meet users’ needs to adjust smoothly to the changes of their business environment and to reduce system maintenance costs that might become a fixed cost, as well as to fulfill the software engineers’ desire to create flexible and stable system architecture. The orthodox system development models cannot meet these needs.

The present situation in Japan with regard to SOA implementation methods is not good. Companies introducing SOA are making silo SOAs. If they continue in this manner, they will not achieve the purpose of SOA.

To prevent this situation, we designed a desirable model for introducing SOA, referring to the architecture of “mass customization.”

To define the generic detail development processes, we did a case study on an imaginary company. In order to maintain independence from vendor software, we prepared the OSS-based SOA environment. Through this case study, we could define the practical detail development processes of SOA and found this would vastly reduce updating development costs.

References

Fig. 5 The Generic Development Processes