



2013.11.27

A Service Modeling Methodology for Service-Oriented Architecture

Dr. Younkon Lee
Korea Polythec University

Abstract

As a paradigm for implementing software architecture from a business point of view, SOA (Service-Oriented Architecture) has emerged in recent years. The services are self-contained and do not depend on the context of the other services, so they are expected to help meet the demands of reusing services. Especially, in the area of international trades which involve so various software, architectures and systems, SOA is expected to be a major role for interoperability between systems in near future. However, it is hard to recommend preferable examples which successfully implement SOA in practice. The main reason is that there is no practicable guideline for executing SOA. First, this specification clearly defined the concepts of service and component which remain confused with each other. Comprehending these definitions make it possible that the important characteristics which SOA should reflect can be caught.

Second, this specification reviewed the reason why organizations should adapt SOA process to their main process and reported the status of other countries which made the best use of SOA as their framework for trade organizations. In addition, domestic and foreign methodologies about SOA are surveyed. Through these reviews and surveys, we could understand what SOA process should be and which tasks SOA process should include. As the result, a new methodology is suggested for modeling services based on SOA in an effective and efficient way. It is a process which identifies, analyses, designs, and realizes services with a series of stages, tasks, and templates of artifacts. Moreover, examples attached in this specification can be directly used as standard templates in a real SOA project.

Finally, to activate applying SOA more vigorously, we present how to measure the return on investment of SOA and educate good architects for enterprise. Though these works for activation of applying SOA and education of architects were suggested, more researches and experiments are necessary to achieve more practical output for future work.

Table of Contents

1. Introduction	5
1.1 SOA modeling: Necessity	5
1.2 Scope.....	5
2. SOA(Service Oriented Architecture): Overview	6
2.1 SOA: Concept.....	6
2.2 Principles of service orientation	6
2.2.1 Services are reusable.	7
2.2.2 Services share official agreements.....	7
2.2.3 Services are combined loosely.	7
2.2.4 Services are abstractions of internal logics.	7
2.2.5 Services can be combined.....	7
2.2.6 Services are autonomous.	8
2.2.7 Services are stateless.	8
2.2.8 Services can be discovered.	8
2.3 Types of SOA adoption.....	8
2.4 SOA life-cycle	9
2.4.1 Initiate SOA.....	10
2.4.2 Develop road maps.....	10
2.4.3 Execution plan	10
2.5 Service life-cycle.....	11

3. SOA service modeling methodology	12
3.1 Overview	12
3.1.1 Service Strategy Development	12
3.1.2 Service analysis/design	13
3.1.3 Service identification/specification	13
3.1.3 Service reification.....	13
3.1.4 Service development.....	13
3.1.5 Service control	14
3.2 SOA service modeling methodology: Specifics	15
3.2.1 Service analysis/design.....	15
3.2.2 Service identification and regulation.....	20
3.2.3 Service reification	21
4. Conformance	22

1. Introduction

1.1 SOA modeling: Necessity

SOA (Service-Oriented Architecture) has recently emerged as a new paradigm to construct IT architecture from the perspective of business because it has been regarded as a new IT strategy that minimizes both the complexity and maintenance cost of corporate IT infrastructures and maximizes productivity and flexibility.

In the late 1990s, many companies began to expand investments in IT to improve productivity and customer services. As a result, numerous heterogeneous systems have come to exist in enterprises. That is, incompatible enterprise systems have caused difficulties in business activities. Thus, industries have made efforts to integrate complex and distributed systems. Currently, SOA draws more attention than any other alternatives as a solution for those problems aforesaid.

In the same vein, SOA paradigm has been applied to implementing diverse information systems including the next-generation financial systems. Yet, companies participating in such implementation projects apply SOA-related methodologies of their own, and hardly publish relevant data. Notably, service modeling, which could be referred to as the essence of SOA, is expected to face difficulties in flexible interconnections due to distinct service units derived and different criteria for service selection. Hence, a standardized SOA modeling methodology is needed that is applicable to SOA-based information system implementation.

SOA life-cycle includes the steps necessary for enterprise-wide SOA adoption, encompassing overall contents to be considered in adopting and implementing SOA. As diverse definitions and contents regarding SOA life-cycles have been suggested, current states are investigated and analyzed here to put overall contents about SOA adoption and implementation in perspective.

Analyzing the roles of service modeling in SOA life-cycle helps understand the significance of service modeling, which is taken into account prior to adoption and implementation of SOA, and further apply the findings to a service modeling methodology.

Services in SOA are based on actual corporate business activities and processes. To derive services, corporate business processes need be put in broad perspective, and requirements need be analyzed. The proposed service modeling methodology involves deriving candidate services from the analysis of business processes, selecting procedures to reify services derived and applying the results.

1.2 Scope

This specification treats on SOA modeling methodology in the respect of service design. The readers can get the information for extracting services from business context, determining service granularity, and making operations in each services.

2. SOA(Service Oriented Architecture): Overview

Lower cost and higher efficiency are key words in current corporate investments in IT, without doubt. Indeed, corporate IT departments focus on integrating systems and applications to streamline and simplify the complex computer systems. In this process, most companies seek to increase productivity using integrated IT resources that enable the legacy systems to cope seamlessly with rapidly changing business environment. After all, they have found SOA as a solution. Even though national and international trade systems involve the most complex IT systems, they should be connected interoperable way for implementing single window systems.

SOA helps develop single window applications into modular business services guaranteeing easy integration and reusability, embodying true meaning of flexible and adaptive IT infrastructures that are to serve as service-oriented trade backbones.

This chapter discusses the concept of SOA and the principles of service orientation to better understand the principles applicable to service modeling and to review the methodological differences between service-oriented development and component-based development (CBD).

2.1 SOA: Concept

As a rule, services refer to offering labor for production or consumption of goods as well as transporting or distributing products. That is, services are acts provided via interface agreements between components. In fact, the term “service(s)” has been in wide use in IT for the past decade. For instance, back in the early 1990s, transaction monitoring software used the term “service.” Transaction monitoring software monitored transactions in middleware regardless of application platforms. Also, since the latter half of 1990s, many web-based application systems have provided commonly shared component services for search engines, authorization and log-in controls.

Services mean functional blocks of software in an enterprise as well. Here, remote and cross-platform uses are assumed, irrespective of internal implementation. Apart from technical perspectives, SOA is a conceptual framework ensuring flexible interoperability with a view to software reusability from the perspective of business.

Therefore, services are defined as software components used to implement independent business functions, which are visible to applications or other services externally, accessed via open interfaces and used as loosely coupled or asynchronous message-based communication. In short, SOA is a way to design software on the basis of services and to build applications as well as services.

2.2 Principles of service orientation

This section mentions the principles of service orientation. Service orientation is based on the “separation of concerns,” a software engineering theory, upholding the idea that problems should be divided and classified into individual areas of interest. Further, the idea is to solve certain problems by breaking down logics into smaller pieces. The disintegrated logic pieces represent original areas of interest.

Service orientation is a way to realize the separation of concerns. The principles of service orientation provide ways to support the theory and underpin the paradigm for implementing many characteristics of SOA. Indeed, some of these characteristics are directly or indirectly associated with the theory of separation of concerns.

Service orientation comes to assume technical significance only when it is combined with “architecture.” “Service-Oriented Architecture” implies a model that has smaller automation logics and can be partitioned into certain logics.

The characteristic principles of service orientation that need be reflected in service design are as below.

2.2.1 Services are reusable.

Whether reusability is an imminent issue or not, services are designed to support reusability. That is, a design standard is applied to offer potential reusability, and to accommodate future demands for reusability, reducing development cost.

2.2.2 Services share official agreements.

For services to interact with one another, they just need share official agreements. These agreements stipulate the content of each service and define the terminology required to exchange information. Service agreements provide official definitions of the following items.

- Service end-points
- Respective service operation
- All I/O messages supported by each operation
- Rules and characteristics of services and operations

Service agreements should define nearly every important aspect of SOA. A good service agreement would provide significant information accounting for how a particular service fulfills a particular task. This information is subject to mutual agreements of service providers and customers requesting the service. All service agreements are shared by services, which is why service agreement design is highly important. Once both parties reach an agreement, the customer should rely completely on the agreed definitions. Thus, once the initial version of agreements is executed and distributed, it is necessary to take care of maintenance and version controls of service agreements.

2.2.3 Services are combined loosely.

Service design should be far from cross-service dependencies. No one can predict how IT environment will evolve. Likewise, it is not possible to plan precisely how automated solutions will grow, be integrated and replaced over time because these changes result from factors external to IT environment at all times. The overriding goal of service orientation is the flexibility to cope with effectively unexpected changes. Loose combinations between services are to directly support and realize such flexible forms.

2.2.4 Services are abstractions of internal logics.

The only aspect of services externally visible is the part exposed through service agreements. The internal logics of services excluded in such agreements are not externally visible, and irrelevant to the party that requests services. This principle is called 'abstraction of service interface levels,' which enables services to act like black boxes, so that specifics can be invisible to the outside world.

The scope of the logic represented by a service has influence on designing the service operation within the process and the service itself. As the amount of logics a service can represent is limitless, it is possible to design a service performing simple tasks only, or to use it as a gateway to an entire automated solution. Also, the sources of application logics available to services are limitless as well. For example, a single service can possibly use application logics from two different systems.

The granularity of operations is directly related to the scope and the essence of functions exposed by services, which should be taken into account in design. Individual operations engage in abstraction of internal logics overall. Services serve as simple containers for those operations.

2.2.5 Services can be combined.

Services can be combined into other services. Due to this aspect, logics can be represented in many different sizes of granularity, which increases reusability and abstraction layers of services. That is, a service can represent any general logic via any forms of sources including other services. The reason to set up this principle is to ensure that services are designed to join other services, if necessary, as effective components.

Orchestration is a concept that the viability for combination is emphasized to expand the SOA. That is, a service-oriented process is controlled by its parent process service engaging in the process.

The demand for every service to be viable for combination highlights the importance of service operation design. The viability for combination is another form of reusability. To maximize the viability for combination, operations need be designed in standardized ways and proper sizes.

2.2.6 Services are autonomous.

Logics controlled by a service have explicit boundaries. That is, services control the logics only that are present within the boundaries, and will not rely on other services for such controls. Without reliance upon other services, a service can freely evolve and spread. The autonomous aspect of services is an important factor to be considered prior to decisions made on how to divide application logics into services or which operations to be clustered into a service.

Deferring the location of business rules is a way to strengthen autonomy and render services more generic. In general, processes take charge of business rules, which dictate how to structuralize processes, and how to construct services with a view to automating process logics, after all.

2.2.7 Services are stateless.

To maintain the loose combination, services should not bear on state information. Although services may defer controlling states, they should be designed to maximize the statelessness. When state control is a must, the responsibility for controlling the state should be imposed upon some other parts, with the service being allowed to remain stateless. Such statelessness is an overarching condition for services, enhancing their reusability and scalability.

2.2.8 Services can be discovered.

Even those who request services without knowing how to use service logics should be able to see and understand service specs. This prevents services from overlapping with the existing services or from implementing duplicated logics.

2.3 Types of SOA adoption

Types of SOA adoption largely involve the perspective of SOA from application development toward application architecture, and the perspective of EA (Enterprise Architecture) at the stage of re-implementing an enterprise-wide system.

Most organizations consider adopting SOA as nothing but application architecture to yield merely visible effects, which is far from high agility and scalability in line with the electronic government services. Arguably, it may be necessary to alter services to a great extent once they are implemented or to attempt to standardize non-standardized services.

Hence, to meet the requirements of SOA adoption, it is rational to model enterprise-wide services from the perspective of EA and then to apply legacy systems or to develop new systems or to employ packages by stages.

As the size and complexity of software systems increase, designing an entire system structure has become a much more important issue than selecting data structures or algorithms for operations.

Structural issues include the organization of systems represented as combinations of components, overall control structures, protocols for communication or data access, allocation of design components to functions, combination of design components, physical distribution, expansion and performance of components and directions of evolution. These issues are defined as design-level software architecture.

For a more abstract definition, software architecture is defined as descriptions on components building systems and interactions between these components, as well as on the patterns as guidelines to combine these components and the restrictions of these patterns.

To apply the perspective of SOA application architecture is to define application components from the perspective of services, when developing applications or when altering legacy systems, and to apply the SOA concept to the control structures between components. SOA can be adopted and applied as application architecture from the following perspectives.

SOA adoption patterns: There are 5 types or patterns of applying SOA as application architecture as below.

- SOA -based distributed application
- SOA -based Websites
- SOA -based portals
- SOA -based EAI
- SOA -based EDI/B2B

As for the development methodology supporting SOA, development units have changed unlike object-oriented development or CBD. Thus, a development methodology with more abstract superior concepts instead of classes or components need be suggested. That is, existing development methodology would mainly focus on enterprise component stages, whereas SOA-based methodology should take business process and enterprise components into account. Hence, services reflecting business processes need be identified.

2.4 SOA life-cycle

SOA life-cycle is a model used to account for the relevance and reliance between diverse independent life-cycle stages so as to successfully implement and manage corporate SOA. The concept of SOA life-cycle is not much different from the existing application development life-cycle. Still, SOA life-cycle differs from the one for application development, firstly, in that its life-cycle includes business process modeling; it derives and designs services from the perspective of services; and it requires the processes to integrate and assemble multiple services. Secondly, each stage of SOA life-cycle need be supported by suitable IT governance across the processes. Also, SOA life-cycle stresses business process, which is comparable to EA, whereas SOA differs from EA in that it introduces more flexible methodology to the existing governance and organization, which requires changes of both business and IT.

The SOA life-cycle model supports a paradigm including SOA strategies and planning as well as implementation and distribution, without being limited to the concept of service life-cycle that supports a series of relevant stages from elicitation to termination of services. That is, the entire SOA life-cycle supports not only service development but also every aspect of corporate SOA.

According to the "SOA Practitioner's Guide" created based on practical SOA experiences of many enterprises and individuals, "SOA life-cycle refers to the stages to implement SOA within enterprises, and differs from the service life-cycle governing individual services." Also, the guide divides the SOA life-cycle into three stages, i.e. Initiate, Develop Roadmap and Execute Plan, as in Figure 1.

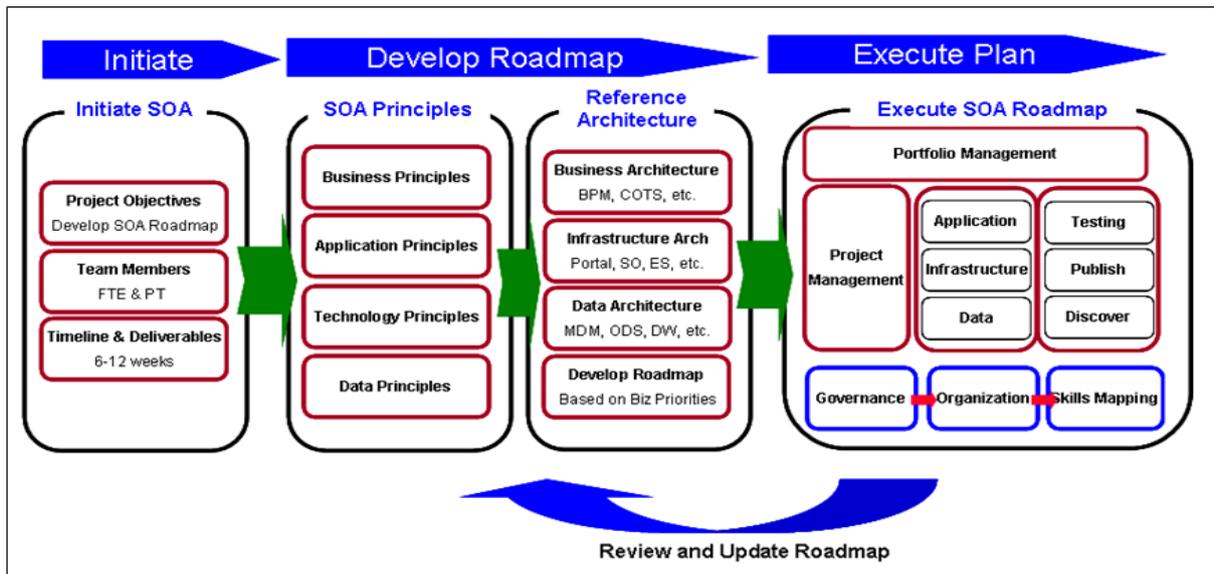


Figure 1 SOA Life cycle

2.4.1 Initiate SOA

This is the first step to decide on to what IT and business functions and potential processes the SOA is applicable, whether SOA applications will lead to improvements and whether they are capable of replacing the existing business. Such decisions are made in the process where IT departments propose SOA-based strategies and explain the effects of SOA to those who are concerned with business. Specifically, this step is to organize project teams, goals, time periods and outputs, so as to develop a road map for cooperation between business and IT. Such decisions need approving by the persons concerned.

2.4.2 Develop road maps

At this step, project teams assess organizational SOA, set up SOA principles, define organizational to-be models (reference architecture) and delineate the process for switching to the defined to-be models from current states. That is, they 1) define explicit and specific SOA principles; 2) set up to-be model images of business; 3) specify resultant to-be model images of IT units; 4) and establish SOA road maps for entire organizations. SOA road maps delineate steps to supply business solutions and infrastructures required to support the to-be model images. Besides, 5) they identify opportunities to prove the advantages of SOA through SOA road maps.

2.4.3 Execution plan

The SOA execution stage describes the methods to execute SOA road maps. Road-map execution can be carried out in terms of project teams and governance and/or organizations. Project teams perform projects in compliance with the particular sequences described on road maps, and build infrastructures needed to retain business competencies. Organizational controls should be practiced simultaneously based on SOA projects. Project teams should review SOA execution periodically and renew road maps whenever major changes are made to plans.

2.5 Service life-cycle

Upon establishing architecture baselines based on the to-be model of SOA, project teams need to review the life-cycles of services for actual execution. Service life-cycles begin with definitions of services and end with retirements of services, which characteristic aspects should be reflected in SOA life-cycles. Service life-cycles consist of such stages as in Figure 2, i.e. requirement and analysis; design and development; and IT operations. Service life-cycles are part of the execution of SOA road maps and that of SOA itself in the SOA life-cycles.

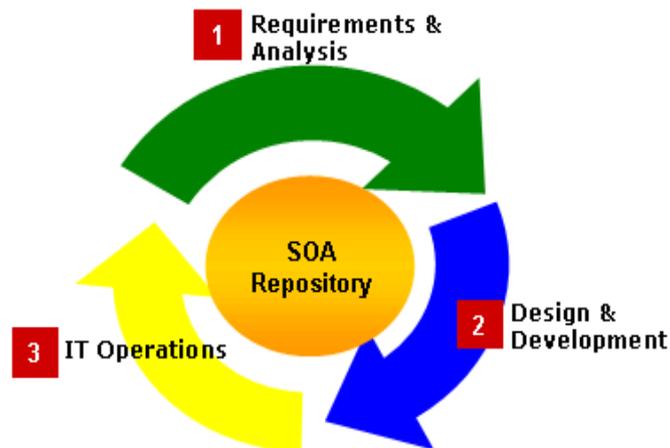


Figure 2 Service Life-cycle

3. SOA service modeling methodology

3.1 Overview

To carry out an actual SOA project in a delineated process by developing and modeling service strategies in line with the analyses of SOA and service life-cycles, SOAM(SOA Service Modeling Methodology) is suggested here. Figure 3 illustrates the overall framework of SOAM.

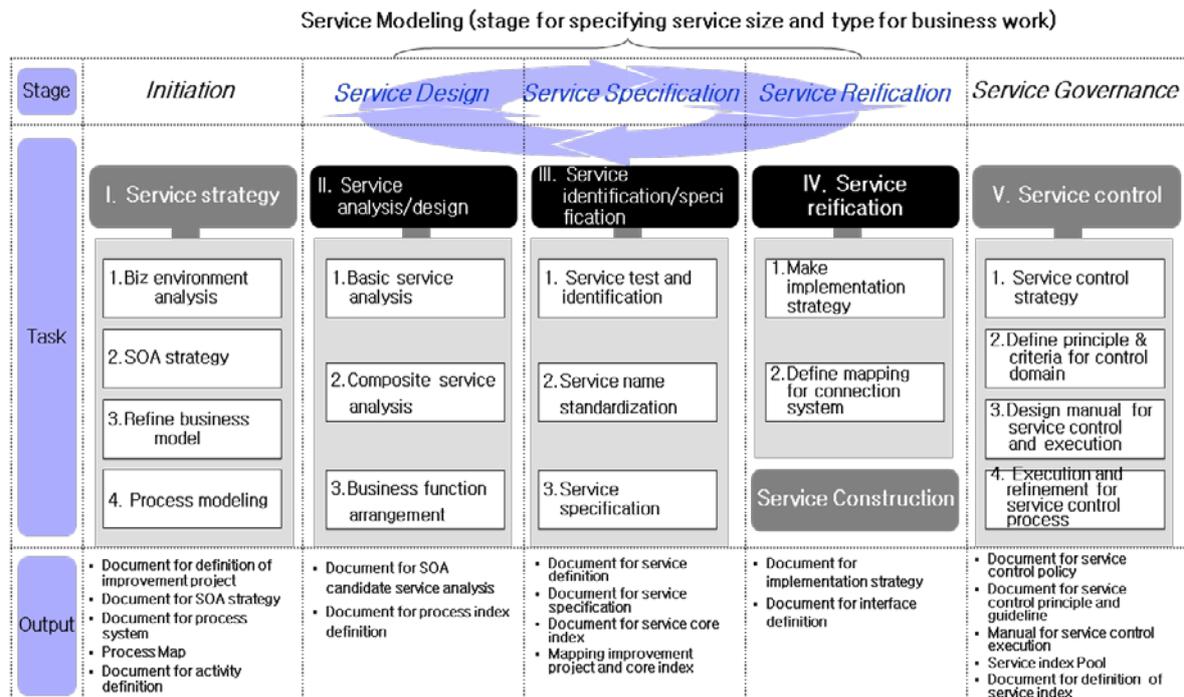


Figure 3 SOAM Framework

SOAM is largely subdivided into Initiation, Service Design, Service Specification, Service Reification, Service Construction and Service Governance stages. Each stage includes such steps as Service Strategy Development, Service Analysis/Design, Service Reification and Service Control.

3.1.1 Service Strategy Development

Developing SOA service strategies is comparable to the existing BPR(Business Process Reengineering) or PI(Process Innovation) processes in terms of rationales. The difference is that developing SOA strategies requires business processes to be analyzed from the perspective of services, and relevant BRM (Business Reference Models) to be refined. That is, with reference to the refined BRM, outputs from the process can be used to analyze and design services.

- Analyze business environment

As part of analyzing overall status of business environment, not only current business environment but also business models and processes are analyzed to derive implications for future improvements. First, the analysis of business environment involves reviewing general status of an organization and its existing strategies, interviewing executives and staff members regarding internal and external competencies, and checking the validity of organizational vision, mission and management strategies as well as the need for changes. Also, the analysis includes identifying opportunity factors and challenges based on the trends and influence factors in relevant industries. The analysis of business

environment helps elicit novel business strategies when the existing strategies are absent or insufficient.

For the inter-relatedness between business and IT based on the analysis of business environment, understanding of overall corporate process is necessary. Analysis of overall IT process status and issues, establishing the scope of performing an appropriate process status analysis of resources invested in the process, and understanding the relevance and characteristics of each process from the perspective of IT in line with the established scope help systematize and clarify the process.

Problems and issues derived from the analysis of business environment are defined as lists of improvement opportunities, which are subsequently reflected in building a target model to set up directions for future information strategies.

- Establish SOA application strategies

The list of improvement opportunities derived from the analysis of business environment is integrated and re-classified for applicable SOA areas. Based on the reference table for SOA application strategy selection, such issues are mapped and specific business areas for SOA application are selected.

- Refine business models

Fundamentally, terms used in an organization need be standardized to apply SOA. The standardization of terminology is to define not just common terms but also how the information sets are inter-related. Then, based on the issues and problems found in the analysis of business environment, business reference models are refined. Likewise, data reference models are refined conforming to the standardized terminology. Here, an enterprise-wide standard glossary serves for SOA metadata, which is why it is a very important output.

- Process modeling

Specification of current processes and changes in target processes explicitly represent the conditions of IT and organization following changes. The changes in current processes resulting from process redesigning of intensive redesign areas need be formalized. Also, the changed conditions of IT and organization need be clarified.

Process maps represent the process-related business flows in a series of activity sets. Process-specific logical flows of business are represented in line with the definitions of activities, describing the overview of activities forming the processes, principal agents and major I/O.

3.1.2 Service analysis/design

Service analysis/design focuses on organizational process and information to identify service candidates, service operation specs and I/O messages of operation.

3.1.3 Service identification/specification

Service identification/specification involves identifying service candidates based on the characteristics of services needed and the test reference table, and defining the quality functions required of the identified services. For further elucidation of the service modeling methodology, see the section 3.2 (Service Modeling Methodology: Specifics).

3.1.3 Service reification

Service reification involves setting up strategies for service implementation regarding development with reference to the list of service definitions, application and alteration of existing systems, and purchase of solutions, and defining the system mapping concerning the aspects that need be interlocked with the existing systems.

3.1.4 Service development

Service development refers to implementing actual services in line with the items defined in the service reification and the requirements of platforms. Service development follows the methodology of implementation in ordinary software development cycles.

3.1.5 Service control

Service control involves establishing strategies for controlling services and refining execution process for the purpose of comprehensive and systematic supports for service modeling. Here, criteria for controlling service levels, service security, service change and service trouble-shooting are defined for a seamless operation of SOA. Also, service control deals with establishing assessment, policy and control mechanisms relevant to responsibilities for authorization, authority and communication chains and decision makers so that the personnel will fulfill duties and responsibilities based on prescribed criteria.

Controlling service levels includes indexing specific items about service use/supply agreed upon between service suppliers and users and defining measurement methods and cycles. Controlling service security requires a range of security types such as controlling service users, access rights, user authentication, transmission levels, message security and electronic signatures. Therefore, service design should include a system specifying the security model. Controlling service changes calls for requesting/confirming service changes, analyzing the effects of service changes, changing services, testing services changed and defining the process of registering/distributing changed services. Controlling service faults includes defining the process of early detection/handling/restoration of services when troubles arise.

3.2 SOA service modeling methodology: Specifics

Service modeling engages with the most diverse issues in the SOA service modeling methodology. The proposed SOAM framework presents process details and practical ways for implementation in regard of the Service Design and Service Specification stages, so as to guide the SOA implementation in practice.

SOAM's modeling framework, as in Fig. 4, is sub-divided largely into service analysis/design and service identification/specification and service reification stages.

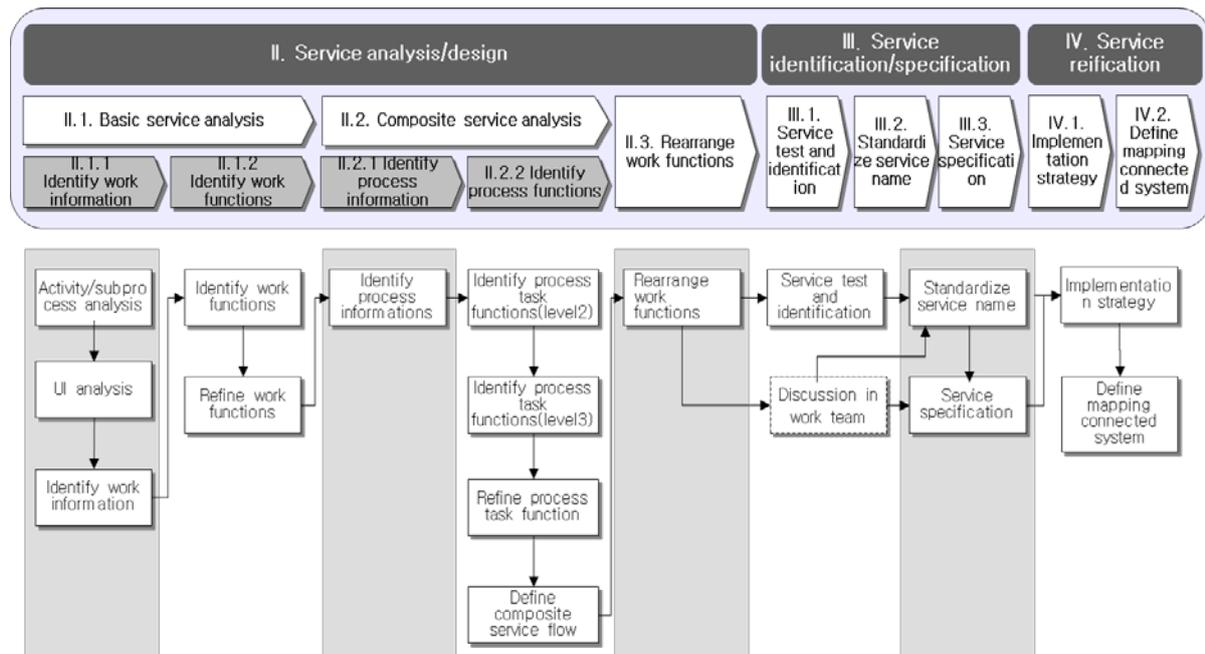


Figure 4 Specifics of service modeling in SOAM

3.2.1 Service analysis/design

Service analysis/design involves identifying business processes or business information, that is, candidate services, as well as operations of services, that is, functions of business information.

Process-oriented service identification is easily applicable, whilst it is not suitable in view of reusability. By contrast, entity-oriented business services are part of the analysis results for fitting the existing corporate business models to the business services from both long- and short-term views. These services are inherently reusable for several business processes.

Entity-oriented business services differ from process-oriented services in the sense that, though entity-oriented business services are built as part of a certain business process-oriented application development project, they do not offer interfaces for certain business processes. Defining this type of services requires entity-oriented services. Compared with the process-oriented services, entity-oriented services have higher agility to remodel service-oriented processes. As process-oriented services are designed to automatize business processes, they may depend on such processes.

Different process logics may lead to different environment, where services are used and combined. Thus, initially defined services become invalid, calling for re-designing or re-developing.

Entity-oriented services require preliminary analyses, which increases time and cost for deriving services. Furthermore, without including the concept of business process logics, they are essentially characterized by generality. Using these services requires higher business controllers, e.g. process services or process control services. That is, generating a business service layer consisting of entity-

oriented services combined by a higher orchestration service layer leads to a desirable structure of SOA, which guarantees high agility and accuracy in representing business models.

With this advantage, the analysis of entity-oriented services demands much knowledge about business for identification and it is hard to offer a guide. As an alternative to this, a user-friendly environment and guide can analyze services on a screen.

For service analysis, first, business information is analyzed to identify business information groups and business functions. Then, the current system's process information is analyzed to identify process information groups and process functions so that business functions can be integrated based on relevance. This step is to extract basic service candidates so as to derive services.

For service analysis, first, it is necessary to identify business information groups, which serve as the basics of inputs and outputs of business functions. For this, individual business activities and sub-processes are analyzed, where manual work activities are excluded and common activities applicable to multiple processes are identified as common services. Via the screen analysis, the components constituting the GUI groups are identified as service candidates.

1) Analyzing activity/sub-process

As the lowest units on the process system diagram, activities and sub-processes are analyzed here. Activity/sub-process analysis also identifies basic information groups as business units via activity I/O data partitioning. When BPM is adopted, the definitions of processes within BPM are used. When BPM is not adopted, business processes are disintegrated to identify commonly used components among the lowest activities and sub-processes as service candidates, excluding those processes that cannot or must not be automated.

2) Analyzing screens

Screen analyses and user interviews are used to identify business information groups. Among the data on a screen, the data group of a classification unit, or a division, is identified as a business information group. Then, the screen information group is classified in the analysis of SOA candidates, which is based on the analysis of activity/sub-process.

3) Analyzing applications

Application service operation is derived from security processing, exception processing and message conversion processing, which are necessary to support business services. The analysis of internal flows of business services helps elicit application service operations that serve as external interfaces to access external/other systems. System events (e.g. completion of processed results and timer events for exceptional situations) are derived as those services corresponding to the 'notify services' used to call other applications for service operations.

4) Identifying business information groups

Based on the results from the analyses above, candidate services corresponding to business information groups are identified. Referring to common business duties of each person, it should be checked whether common functions exist among service candidates derived from individual business areas.

Also, for standardization of terms, it need be checked whether services of same functions are identified under different names, or whether services of different functions are identified under same names (standardization of homonyms and heteronyms). Then, to identify the information sources of service candidates, categories and divisions are described.

- Category

Item	Marking	Remarks
Common	Common	Common processed or activities
Process	Process	All cases excluding common ones.

- Division

Item	Marking	Remarks
Business information service	Business information	Service candidates regarding information groups on the screen
Process service	Process	Service candidates based on the analyses of processes, sub-processes and activities

- Name candidate services.
- As for business information services, use the I/O information names marked in the process system diagram.
- Refine the names based on the outputs of terminology standardization.
- Mark names in English based on the outputs of terminology standardization.
- Assign reference information group ID and describe classification (category/division) data.
- Assign and describe ID numbers to identified information groups (process, business information).
- Importance of business

Item	Marking	Remarks
Basic information service	Basic	Core information on strategies and business processes
Support information service	Support	Information supporting processing basic business
Reference information service	Reference	Reference information such as management indices

- Frequency of use

Item	Marking	Remarks
Cases used in general	H	
Cases used as necessary	M	
Cases used for specific purposes	L	

- I/O information of candidate services.

Item	Definition	Marking	Remarks
Key	Key values needed to obtain the given info	K	
Return	Detailed attributes of services	R	

- Sources of analyses regarding candidate services

Item	Definition	Marking	Remarks
Screen	Derived from screen analysis	S	
Process	Derived from process analysis	P	

Use case	Derived from use case analysis	U	
----------	--------------------------------	---	--

- Assign IDs to candidate services.

5) Identifying business functions

Previously derived business information groups are used here to elicit functions to be offered. These functions are defined as the operation lists of services.

Functions to be offered using business information groups (e.g. view, edit, create and delete) are identified as business functions.

6) Refining business functions

Business information groups and functions are refined to describe the analysis of SOA candidate services, that is, the business functions.

Using the company-wide standard glossary, business functions are named in Korean and English. As per the criteria for classifying unit systems, descriptions on the system keeping business functions are created.

7) Identifying process information groups

The information is the results of identifying process-level business information groups and classifying previously identified business information groups in light of the process.

The process levels are based on the screen specified in the screen design description. Previously identified business information groups are clustered into process levels to derive process information groups. Then, the results are described on the information-group sheet of the SOA candidate service analysis. The following need be considered here.

- Combinations of highly reusable business functions should be created.
- Combinations of business functions should be applicable to other teams.
- A union of functional sets complying with the above conditions should be made.

8) Identifying process business functions

Process business functions are identified based on the following criteria.

- Individual screens.
- Sets of business functions identified based on individual screens are defined. (They will be provided as service functions for future BPM.)

9) Refining process business functions

Process business functions (level2 and level 3) are refined to describe the SOA candidate service analysis (or the business functions). Business functions are refined in the same way as the business information groups.

10) Defining composite service flows

Relationships between business functions constituting individual process and business function are defined. Relationships between business functions are based on non-manual units of functions. Composite flows are defined with reference to the criteria for classifying the pre and post reliance of business functions.

- Service sets (service composition): Packs of services necessary to achieve business goals.
- Set services are identified from business process models. Then, the components of the set services are identified. When the basic services are identified from the L3 business processes, the L2 business processes can be considered a set service.

- Component services of set services share the same business goals.
- Flow specs of component services constituting service sets are defined.
 - The sequence and flow of basic services forming the sets (Composite Service) are defined.
 - Service flows are defined based on the business process models.

11) Aligning business functions

Functions of the analyzed candidate services are aligned in light of the criteria for service integration.

- The information of individual business value stages and the business functions are classified using sequential alignment based on the analyzed items of candidate services.
- The alignment should follow the classification (category and division) of candidate services.
- The aligned data are used to align the classification of business functions/the importance of business.
- The aligned data are used to align the business functions of candidate services.
- Although candidate services belong to the same process, they are identified as candidate services, as long as they meet at least one of the criteria below and provided they exist for business purposes.

< Classification >

- Division = Information groups belonging to 'process' and sub candidate service ID
- Category = 'Common'

< Business functions >

- All business functions belonging to a given service
- Groups of general common attributes are identified and integrated. Groups of common attributes are identified and integrated for each information group ID classified as business information groups.
- Partially common attributes, if any, are identified and integrated.
- Partially common attributes are integrated based on the following.
- Business functions related to integrated common attributes groups are integrated.
 - Integration of business functions via the integration of information groups follows the guidelines below.
 - The entire common attribute groups are integrated.
 - When business functions are identical, they should be integrated into one business function.
 - Different business functions are integrated into two business functions.
 - Here, extra attention should be paid to homonyms and heteronyms.
- Service integration follows the criteria below.

- Functional relevance: Services offering identical or similar functions are integrated or disintegrated. When services provide identical or similar functions, the services are integrated. When messages exchanged to request or offer services are identical or similar, those services are integrated. When users are not capable of identifying business functions offered by services explicitly, those services are integrated (or disintegrated).
- Easiness of implementation: Technical easiness and constraints relevant to service implementation are considered in integrating or disintegrating services. Integration of identical applications on identical systems and of services supported by identical database instances is considered. Integration of operations offered by identical program modules or of services identified by method levels is considered. Services for accessing identical data models or tables to create/delete/modify are integrated.
- Easiness of operation: Integration (or disintegration) of services implemented is considered in favor of the easiness of operation/control. Disintegration of services is considered for users who apply different security items. Services are disintegrated when non-functional requirements (performance, fault tolerance levels and valid time for service) of service users are different. ADD(AD), MAP(MA) and ME(Merge) are marked as influential principles to track down changes in business functions.

Item	Definition	Marking	Remarks
Add	Add new business functions	AD	
Map	Add relevance of given business functions	MA	
Merge	Modify similar business functions in terms of contents, and register them as new business functions	ME	

3.2.2 Service identification and regulation

1) Service identification and specification

This is to verify identified service candidates' compliance with SOA goals so as to confirm services. First, a reference table to be used for testing and selecting services is created. Business items are selected based on interviews with business organizations. Items selected are used to assess the compliance of services with business goals, if service users can identify and use given services as per the service specification and other compliances with other conditions required by business organizations. IT items are selected based on interviews with IT staff members. Items independent enough to assess the easiness of replacement and assembly are selected.

2) Standardizing service naming

End user groups review service names. Services that pass the final test are reviewed. Target names are those that pass the test among the service names mentioned in the SOA service candidate analysis. Users get involved in modifying and creating service names. User-modified results are reflected in service names. English names are defined based on the standard glossary data. Service names convey the functions of integrated services. Service names and functions should be understandable to the users of given domains without any prior technical knowledge base. Whether to name services using verb combinations implying operations or nouns should be marked in line with the standard terminology.

3) Service specification

Operator-oriented service list structures are reviewed. This is to provide easiness of searching and using services when service repositories are implemented. The specification is based on the definitions in the service spec. Practitioner-oriented service specification is examined. Service messages are defined. The contents and formats of service I/O messages are defined.

4. Conformance

This specification presents the modeling methodology for services in SOA environment and additionally characteristics of services and service-oriented architecture and gives an account of the principles of service orientation. Services on SOA could be more reusable and interoperable, if they are designed according to this specification. The services are also more appropriate and conformable in the context of organization's business.