Service Identification and Modeling for Service Oriented Architecture Applications*

Min Je Cho, Hyung Rim Choi, Hyun Soo Kim, Soon Goo Hong, Yavuz Keceli, Jae Young Park
Department of Management Information Systems
Dong-A University
840 Hadan-dong, Saha-gu, 604-714, Busan
South Korea
er6ty@donga.ac.kr, {hrchoi, hskim, shong}@dau.ac.kr, ykeceli@naver.com, powerbrain80@gmail.com

Abstract: SOA (Service Oriented Architecture) is not a new concept, indeed it is a software design method which is the combination and enhanced version of various existing software methods. It is rapidly gaining ground, especially as a method to solve the common problems of software re-usability and inter-compatibility. For this reason, SOA is considered to be the optimum alternative for responding the new changes in the business environment. For the successful implementation of a SOA-based system, the composition of the services is essential, and service identification and modeling is the essential starting point of successful implementation of SOA systems. On the other hand, it is a fact that there are neither any standards nor standardized methods for business service modeling. Thus this study proposes a service identification and modeling method for deducting services that fit system purposes, and deducts actual services for agent-based purchase management systems.

Key-Words: SOA (Service Oriented Architecture), Service Oriented Analysis, Service Identification, Service Modeling

1 Introduction

Lately changes in business environment are expanding the business management paradigm from internal problems to external problems of an enterprise. Finally, the implementation and operation of information systems that enables connection with various partners determine the competitive power of the enterprise to survive. For this reason an approach that considers business and technology together is necessary in order to ensure flexibility and safety, which are most important problems information system implementation. This means a system that considers a user interface enabling the mutual cooperation of the system users. Such systems need to be implemented using Service Oriented Architecture (SOA). Being based on services, SOA makes it possible since it enables the merits which are demanded by the latest systems such as, smartness, re-usability of processes and existing assets, extensibility, task flexibility through loosely coupled processes (Krafzig, 2005).

On the other hand, although the interest in SOA and the efforts to deploy it are currently increasing, concrete procedures for service identification and definition have not been given in SOA related methodologies proposed until now. For this reason there is certain difficulty in applying SOA to real systems.

Thus, this paper proposes a service identification and modeling technique which is necessary for designing SOA-based systems that can rapidly cope with the changes in business and technology environment. For this purpose, definition and characteristics of SOA and service were examined in Chapter 2. Then in Chapter 3, a new modeling method for services that satisfies the principles of service orientation. Services of an agent system were modeled using this method in Chapter 4, and finally in the conclusion part, implications of this research and directions for future research were searched.

2 SOA AND SERVICE

2.1 SOA and Service Orientation

The improvements in information technology led the revitalization of e-business, and the e-business environment requires connections between internal information systems and processes of an enterprise and external customers and information. Standardized business processes are demanded and the limits of traditional systems must be overcome through

* This work was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2006-042-B0006)

development of interface criteria between businesses and active application of information technology.

For this purpose, previous vision must be changed by separating business and technology. Also, all business entities, constitutes and application systems must be recognized as a whole system. Based on such a demand, SOA was first introduces in mid-90's by Gartner Group, and now it is regarded as optimum alternative to confront new changes in business environment.

The concept of service orientation was derived from the concept of separating concern, which means if the demand for solution for complexity problem and the logics necessary for solving enormous problems are separated into smaller related pieces, it would be much easier to implement, execute and manage the system (Thomas Erl, 2006). It was first introduced in 1996 by Gartner Group, and then improved by definitions and concepts in various forms. Most of the definitions about SOA are derived from distributed software architecture or framework, including combination of services (CBD Group, 2003, W3C, 2005; Munindar, 2005; Erl, 2005). SOA can be defined as a software design paradigm which composes business functions as a set of shared services that can be re-used on the network.

Krafzig explains the composition of SOA on the basis of 4 basic abstract concepts, namely Application frontend, Service, Service repository and Service bus, as it is shown on Fig.1.

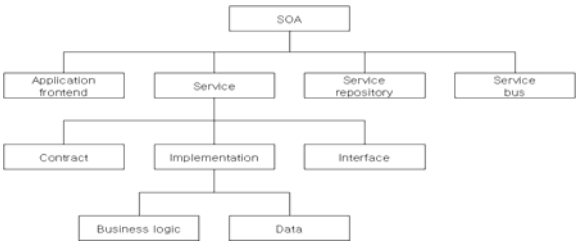

Fig.1 Composition of SOA

In the Fig.1 above, the factors except service has the characteristics of the system infrastructure, and the service itself can be considered as the actual software component. But at the moment SOA does not have a standardized industry definition, service orientation principles are not fixed as international standards, and there is no standard method for modeling business services. There are various opinions on service modeling for SOA. Typically, methods such as SOMA, SOAD, SODA, and SOUP (Arsanjani, 2004; Zimmermann, 2004; Nigam, 2005; Mittal, 2005) have been proposed, but a concrete and empirical method has not been proposed yet (Erl, 2006). The previous models were nothing but defining service patterns to some extend according to service orientation principles and service characteristics as a guideline for service modeling. But such definitions use different terminology depending on the point of view.

As it is seen in Table 1, service orientation principles can be defined under 8 broad categories. Among them, autonomy, re-usability, loosely coupled, abstraction, necessity for formal contract are the ones that form the basis SOA. These four principles not only help each other to be realized easily, but also help the realization of other characteristics directly.

TABLE 1 Service Orientation Principles

Service Characteristics	Explanation
Re-usability	Design supporting re-usability of the services, even though there is not any immediate use opportunity
Sharing of Formal Contracts	Necessity for a formal contract that describes the service and contains the dictionary of necessary idioms to communicate between services for inter-compatibility
Loosely Coupling	Design of the services such that they are inter-operable while they do not strongly depend on each other
Abstraction of Internal Logic	Not sharing any internal logic except for those listed in the formal contract
Collaboration	Ability to collaborate with other services, while each of them can have their own logic to improve re-usability and generate abstraction layer
Autonomy	The managing logic has clear boundaries, and within those boundaries the service has its own control capabilities without depending on other services
No situation information	In order not to interfere with loosely coupling property, services do not have situation information
Discoverability	Exposing its specifications and enabling the service user to understand those specifications

2.1 Service and Service Levels

A service is a software component that represents a clear functional meaning. Services encapsulate the business concept, and the other factors form the basis for discovery, combination, and use of the services. Thus, the concept of service assumes that service and the application that calls the service do not need to be in the same physical environment, and implies the meaningful movements executed by a computer program upon the request of another computer program. As it is seen in Fig.1, services include service contracts, and shares the specifications of the service through it. The specifications of the service include the contents about purpose, functions, restrictions, rules of the service, and definitions about the interface. More independence and abstraction can be provided from this technique. It means, by using the

contract of the service, loosely coupling of services is possible. Unrelated to the internal characteristics of other services, communication between services is possible if interface requirements expressed in service contract specifications are simply known.

It is not easy to develop business services by reflecting the previously-explained service orientation principles onto such services. Each of the business services possesses different characteristics according to business entity, function and processes. In order to develop as a service, the logic must be expresses as service, and loose coupling must be considered. Finally, the logic must be expressed as an extension of business model, and an abstraction layer is needed to make sure that the service is intelligent. In a general enterprise model, the service layer is situated in between the business process layer and the application layer. Thus, with the service layer, the business process logic is made independent from technical application layer which automates the business process logic and loose coupling between business and application logic is enabled.

In such a service layer, the characteristics and role of each service are re-classified according to other service patterns, such service pattern and service layers are given in Table 2.

TABLE 2 Service Layer and Service Type

	Service Layer	Service Type
Sundblad (2004)	Presentation Layer	Presentation Service
	Process Layer	Process Service
	Activity Layer	Activity Service
	Entity Layer	Entity Service
Krafzig (2005)	Enterprise Layer	Public Enterprise Service
	Process Layer	Process Centric Service
	Intermediary Layer	Technology Gateways Adapters Facades Functionality-Adding Service
	Basic Layer	Data-Centric Service Logic-Centric Service
Erl (2006)	Orchestration Layer	Process Service
	Business Service Layer	Task-Centric Business Service Entity-Centric Business Service
	Application Service Layer	Utility Service Wrapper Service

The classification of service layers and service patterns both enable the independence of each of technical application logic and business process logic, and connects them loosely as described before. Also the problems are separated for simpler solutions, classified vertically to make it easier to re-use and expand, and abstraction is discriminated according to specialized viewpoints to enable rapid management of change. Each of the classes is given in Table 2. Finally in SOA, the separation of service and service layer enables the separation of business and application as different layers, and services are classified again according to characteristics of the service by considering re-usability in each separated layer. And services and layer that controls and manages them are formed by combining them as building blocks.

3 SERVICE IDENTIFICATION AND MODELING

It is already mentioned that international industrial standard definitions about SOA and service orientation principles are not fixed; moreover there exists no standard method for business service modeling at the moment. But when conducting service oriented solution projects the point that must be considered is that the characteristics of SOA's service orientation theory must penetrate in the project starting from the analysis and design phases.

Michael (2006) proposes a methodology for service design and development according to service oriented architecture, and indicated the steps of service design and development as (1)Planning, (2)Analysis and Design (A&D), (3)Construction and Testing, (4)Provisioning, (5)Deployment, and (6)Execution and Monitoring. The service analysis step is also divided into sub-steps of process identification, process scoping, business gap analysis, and process realization analysis, while service design step covers the sub-steps of service design concerns, specifying services, and specifying business processes. On the other hand, these principles only provide a detailed guideline; the method was not actually applied to real service analysis and design. Zimmermann of IBM (2005) proposes SOMA (Service-Oriented Modeling and Architecture), which approaches service development process from business point of view. This method covers three steps, namely identification, specification, and realization. Through these steps, the method for mapping the business structure in IT level is explained, business goals and SOA are matched, and, finally the business processes and the basic software applications supporting these processes are connected. This model defines service model on the basis of deducting business process model and use case model, but it fails to provide any method for analysis and modeling of particular services. Han (2007) classified services into business aspect, service aspect, component aspect, and deducted prospective business process service from TO-BE process model for service identification. And deducted prospective business services form business processes, use cases,

and user requirements. The final services are deducted through a refining process in which business services deducted in this way and prospective application services deducted from the legacy information system, are confirmed so that they all satisfy the service requirements and the services do not overlap. But such a method is not applied and confirmed in the study. Shin (2007) deducted services with a simpler method, using 1:1 mapping and partition method on use cases. 1:1 mapping means deducting one service out of one single use case, and partition means deducting final services when one use case is divided into smaller service units. This method has several limitations, such as difficulty in determining the level of the services and dependency of the size of the service solely on the use case. In the study of Kim (2007), a CBD method that consists of seven phases from service identification to orchestration design was proposed for service development, but only definitions of each phase and contents were given, no real method was proposed.

When previous studies on service identification and modeling are inquired, most of them discuss the particular processes or methods, basically services are defined and classified through service identification process that deduct prospective services, and the services are extended through a refining process.

In this research, business processes are built according to service orientation, and service modeling using bottom-up strategy in which analysis comes first was done. Also, considering service orientation principles, an agent-based service was modeled on the basis of the proposed model. The proposed service modeling procedure is given in Table 3.

TABLE 3. Service Modeling Procedure

Step	Name of the Step	Activity
0	Deduction of Entity Model	Deduction of additional entities and business processes on the basis of critical entities used in the defined task scope
1	Business Process Disintegration	Separating the deducted processes into individual steps
2	Identification of Prospecting Service Operations	Identification and grouping of prospecting service operations through the separated processes
3	Deduction of Prospecting Services	Deducting prospecting services by re-separating the operations according to the service orientation principles
4	Service Refinement and Confirmation	Refining the deducted prospecting services, correction and confirmation of the services

The deduction of entity model, as a prerequisite task necessary for service modeling, first previous task model is updated or newly generated through entity modeling for the defined task scope. This is for deducting additional entities focusing on the critical entities used in the define task scope, and for deducting definite business processes in the task scope. There are 1:1 or 1:n relationships between these entities. When the critical entities are deducted and their relations are defined, the transaction domain of the corresponding task has been expressed.

Then, service oriented analysis are carried out in order to reflect the business automation requirements. In this phase, which services are to be implemented and business logic that each service must to contain are examined. Each phase of the business process can differ according to user requirements and viewpoints of service definers. Separating process workflow logic into process phases is extremely important, and the level of separation of the phases can differ according to process size. After that, according to the separated levels, prospective service operations are defined, and individual prospective services are deducted by grouping according to logical separation criteria.

In such a process workflow each process phase is expressed as prospective operations and business service operations are identified. After excluding manual process phases that cannot be automated, and process phases that represent the legacy system logic, only treatment phases that are related to service modeling processes would be left.

These prospective operations will form the basis of prospective services, and refined into services that satisfy the basic principles of service orientation, such as re-usability and autonomy. Through this refinement process service and prospective operations are deleted or added. Service operations control and maintain the state of the process. Then an upper business service layer is implemented and all treatment processes that are necessary to combine basic prospective services are included. Thus, service combination when needed is enabled, as explained below.

The process layer is an essential layer emphasized in SOA. The basis of re-usability is on the strict separation of process and basic service layers. When the process layer is once defined, it can only be used within a single context. On the other hand, in the case of basic services, when once defined, they can be used repetitively in different contexts such as business processes, branches and work sites (Bieverstein, 2006). Hence, except for process centered services in the upper level, remaining services can be named as basic services. The abstraction of service layer is composed of various services that perform individual business functions. The ones indicated as basic services for combinations and these services are used on calls as

they are lower level services. Finally, since the main purpose of SOA is maximizing re-usability in each of the abstracted layers, it can be achieved through spotting overlapping capacities, making new services out of common functions, and assembling new business functions by connecting the re-usable services together.

4 SERVICE MODELING FOR AN AGENT-BASED SYSTEM

The concept of service orientation can be applied to not only general enterprise systems but also to systems of various fields. Especially in the case of multi-agent systems where management of change and communication play crucial roles, the necessity of application of SOA is more obvious. In multi-agent system development, the problems of heterogeneity and distribution has always been an issue (Jennings, 1991), and overcoming such limitations has been the major motive of multi-agent related research. But service-based agent systems can solve the problems of re-usability, application dependence and complexity, better than the previous architectures.

In this research we defined the system to be modeled as the supply chain of a manufacturing company. The scope of application was determined to be a part of service offering for customer order management and order management tasks which is the response to the order. In the previous system, requests for orders were carried out manually when purchase officer fills in the order form. The order officer gets the order information and keys in the information to the system. After checking the inventory and manufacturing status, the order officer informs the purchase officer if the order can be acceptable or not. The main problem of this system is that it does not allow flexible transactions with various purchasers through an online system. In order to process the orders of various purchase customers, the system must support various order request types. For this reason this research proposes a service model for SOA-based multi-agent systems. Through the application of SOA, the agents that are responsible for purchase requests are able to use services instead. The basic purpose of this approach is standardization of the external access point of the system with SOA, increase the number of online customers, and enhancing operational efficiency through the automation of order processing.

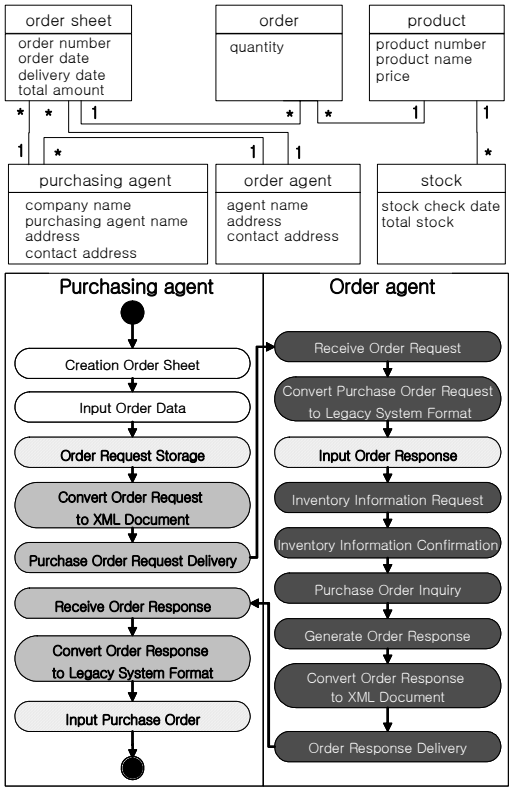

Fig.2 Entity Model and Order Management Process in Order Processing Task

First of all, the entity model was derived according to the defined tasks, as it is seen in Fig.2. As explained before, the entities of business processes related to order are in order request form. And additional entities such as order, product, purchase officer, order officer, inventory are also used in order processing. There is 1:n relation between order officer and purchase officers. Similarly, there are 1:n relations between purchase officer and order form, and between order form and ordered product. A single product can be ordered by various purchasers. Through the deduction of such entities and the relations between them, the domain of corresponding tasks and transactions are expressed.

The second step is service oriented analysis to reflect the business automation requirements. The defined process is given in Fig.2 according to the given task scope. In such a process workflow, each process step is considered as a prospective operation, and business service operations are identified. The steps that cannot be automated and the ones related to logic of the previous system were excluded, and only the steps related to order processing were left for service modeling, which are the potential prospective operations and these are shaded in Fig.2.

These prospective operations are divided in to three prospective service groups according to the service classification criteria as shown in Fig.3 and they are refined according to their satisfaction of the basic principles of service orientation, namely re-usability and autonomy.

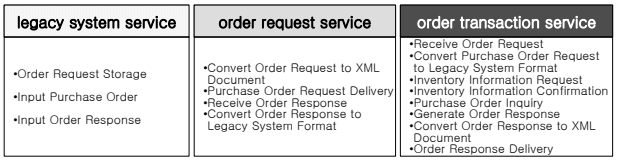

Fig.3 Deduction of Early Prospecting Services and Operations

First service related to legacy system includes data storages for order requests and other related data. And, a new prospective service called document conversion service was deducted by combining the operations related to conversion of XML documents. On the other hand the operations related to document delivery and acceptance were selected as prospective services but they were not considered to be appropriate. Since messaging is one of the basic characteristics of service operations, they do not need to be referred as separate services. Thus, these were replaced by new prospective services that inform the users about message delivery results and arrival of new messages. As the results of these refining processes, prospective service groups were derived from the prospective operation groups given in Fig.3, and these are given in Fig.4.

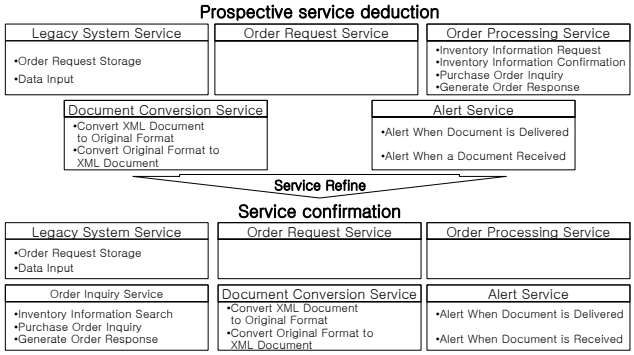

Fig.4 Deduction of Services after Service Refinement Process

In prospective order processing services, the generation of order response directly depends on request and confirmation of inventory information, and purchase order inquiry. Thus, these operations were extracted separately as a new order inquiry service. In the case of operations like inventory information request or inventory information confirmation, these can also be combined as legacy system services. But since the characteristics of information related to these two operations are diverse, these were not included in legacy system services, but they were combined in an operation called inventory information search and it was included in the order inquiry service.

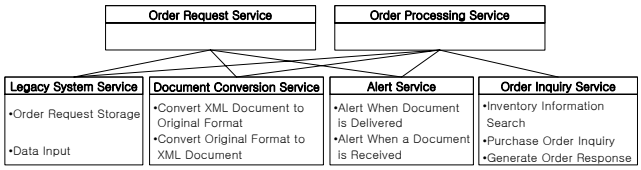

Fig.5. Service Collaboration for Order Management

As it is seen in Fig.5, prospective operations of order request service and order processing service were deleted through such a refining process. But these two prospective services were not totally unnecessary. These two are encapsulating the knowledge about order management processes as central service.

Thus, except for these two process centered services, the remaining services are indicated as basic services. It has already been explained that the abstraction of service layer is composed of various services to conduct an individual business function. The services indicated as basic services, i.e. legacy system service, document conversion service, alert service, and order inquiry service collaborate with each other. Being the inferior services of order request service and order processing service, they are all activated by calls when there is an order request or processing. As it is seen in Fig.5, order request service and order processing service are modeled such that re-usable basic services can be used as a combination.

The services that are modeled in this way are stored and managed separately from the service-based agent system, as given in Fig.6. And when the tasks and roles of the agents change, it is till possible to use the other services.

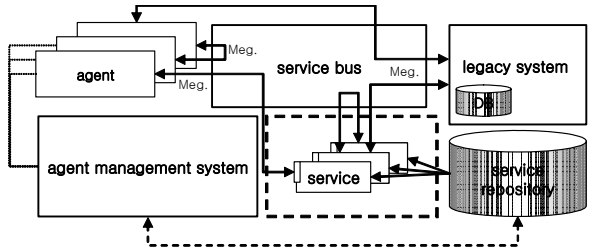

Fig.6. The Architecture of Service Oriented Multi Agent System

5 Conclusion

Nowadays, countless approaches have been tried related to SOA. But such studies couldn't go further than applying SOA to an individual system or simply conceptual approaches. Especially, in the current situation where there are no standardized methods for business service modeling, big solution vendor companies are doing great effort to expand their reference projects on SOA. It is very important to use modeling method which is in accordance with system purposes for the application of SOA. This is the optimum method for SOA implementation to respond the rapidly changing business environment and to ensure the competitiveness of the company.

This paper proposes service identification and modeling method to be used in modeling business processes of a multi-agent system. Some methods have already been proposed, such as SOMA, SOAD, SODA and SOUP, but these methods describe basic requirements rather than focusing on service identification. Moreover the steps of modeling were defined but particular activities in the steps were not. For this reason, due to the absence of a systematic and realistic method, modeling of the services needed for an agent system were done according to service orientation principles and service characteristics.

This study has solved various limitations and problems faced by software developers, namely re-usability and inter-compatibility. In the future, the results of the study are expected to be applied in service applications in various fields.

References:

[1] Ali Arsanjani, Service-Oriented Modeling and Architecture(SOMA), *IBM developerWorks*, 2004

[2] Dirk Krafzig, Karl Banke, Dirk Slama, *Enterprise SOA*, Prentice Hall PTR, 2005, p.6

[3] David Sprottand Lawrence Wilkes, Understanding SOA, *CBDI Journal*, 2003.

[4] Frederick P. Brooks, Jr., No Silver Bullet : Essence and Accidents of Software Engineering, *IEEE Computer*, p.10-19, April 1987.

[5] Kim myung-il, You beom-jong, Service Oriented Architecture-Based Design for the Construction of the National R&D Information System, *Proceedings of 2007 Spring International Digital Design Invitation Exhibition*, Vol 5, No 1, 2007, pp. 34 ~ 37.

[6] Michael P. Papazoglou and Willem-Jan van den Heuvel, Service-oriented design and development methodology, International Journal of Web Engineering and Technology, Vol 2, No 4, 2006

[7] Nobert Bievirstein, Sanjay Bose, Marc Fiammante, Keith Jones, Rawn Shah, *Service-Oriented Architecture Compass*, IBM, 2006, p.178

[8] Olaf Zimmermann, Pal Krogdahl, Clive Gee, *Elements of Service-Oriented Analysis and Design*, IBM, 2004

[9] Olaf Zimmermann, Niklas Schlimm, Günter Waller, Marc Pestel, Analysis and Design Techniques for Service-Oriented Development and Integration, *Lecture Notes of INFORMATIK*, 2005. 9.

[10] Samir Nigam, Service Oriented Unified Process(SOUP), http://www-128.ibm.com/developerworks/kr/library/ws-soa-method1.html,

[11] Sang Woo Han, Sun Hee Park, Jae Ho Ro, How to identify services for your SOA, *Journal of Computing Science and Engineering*, Vol 24, No 1, 2006, pp. 27 ~ 31.

[12] Sten Sundblad, *Service Oriented Architecture and processes*, Sundblad & Sundblad ADB-Arkitektur, 2004, p.6.

[13] Sumi Shin, Bum-jong Yu, Dong-gu Shin, Effective Service Modeling Method through Use case Analysis, *Proceedings of Korea Computer Congress 2007*, Vol 34, No 1, 2007, pp. 58 ~ 61.

[14] Thomas Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall, 2006, p.32.