A Combinational Approach to Service Identification in SOA

Shirazi H.M, Fareghzadeh N. and Seyyedi A.

Malek_Ashtar University of Technology, Faculty of ICT, Tehran, Iran,
E-mail: shirazi@mut.ac.ir

Abstract: Development of software systems based on Service Oriented Architecture requires plain and suitable standards, processes and steps. In addition, although various techniques for analyzing service-oriented process and identifying different services exist, these approaches are unable to represent a clear method for developing software based on service-oriented principles and software engineering rules. This article suggests a combinational approach in order to identify different services in the analysis step of Service Oriented Architecture development process. The proposed method has advantages of primary approaches and could represent an explicit view by describing details of analysis.

Key words: SOA, Top-Down Approach, Bottom-Up Approach, Service Taxonomy.

INTRODUCTION

In last four decades, software complexity has increased and at the same time demand for powerful software has dramatically risen. Since traditional approaches are not suitable for handling current requirements, development of effective approaches which could solve complexities in shorter period of time is essential. Because it is not possible to simply take current software serving clients away, so new software systems must be provided beside existing systems [1,2]. Service Oriented Architecture (SOA) was proposed in late 1980s by two companies named Microsoft and IBM to solve mentioned problems. SOA has relied on distributed calculations, network hidden layers and also languages for providing distributed software [3].

Concept of Service-Orientation: SOA is a collection of rules, policies and frameworks which enables software to represent its operation through a series of independent services. Consequently, other applications are able to discover and call its services by means of standard and default interfaces without awareness of their implementation methods. Web service technologies and practical implementations indicate that SOA is a suitable approach to design, develop and form current large-scale systems [4,5]. All functions in SOA are defined as services including business functions and transactions. They contain low rank and service system functions. Services are designed and implemented independently and autonomously. Since they act as black boxes, other parts can use their outputs without awareness of inner implementation. These services can be used in combination and they are reusable as well [1,3,6,9].

SOA Delivery Lifecycle Phases: Lifecycle of a SOA delivery project is comprised of a series of steps. These steps need to be completed in order to construct the services for a given service-oriented solution. Lifecycle of delivering a software project based on SOA approach is illustrated in Fig. 1.

As it could be seen from Fig. 1, at first step which named service-oriented analysis; domain and business process of organization as well as services are identified and modeled. In service-oriented design stage, working plan besides service layers and required interfaces are prepared and designed. Subsequently in service development phase, identified services are implemented using programming languages and working places. In next phase called service testing, quality of services is tested before their entrance to real operational place. Afterwards in service deployment step, generated parts and defined interfaces are loaded, plugged and ready for distribution. Finally, in service administration stage, monitoring services, managing messages and facing crises are performed [1,2,6].

Strategies for Service-Oriented Analysis: In initial step called service-oriented analysis, potential scope of SOA is determined. Service layers are planned and individual services are modeled as service candidates which comprise a preliminary SOA. Different strategies for organizing lifecycle stages exist which the most important ones are as follows:

Top-Down Strategy: In Top-Down strategy, several actions should be taken, including understanding and capturing broad functional domains; identifying detailed
business processes, sub-processes and their handshake points and also recognizing high-level service candidates based on process activities. Top-Down strategy allows architects to understand business context and provide definition and scope of SOA project boundaries and domains. It bridges the gap between IT and Business teams and makes the preliminary list of activities, which is a potential list of services for further analysis and activity/service dependency identification at a high level.

**Bottom-Up Strategy:** Botom-Up strategy involves analyzing enterprise-level application portfolio and assessing their reusability, redundancy, and rationalization. In general, aim of Botom-Up approach is providing redundant business logic and multiple copies of business data entities or implementing same business logic with multiple products which results in high license, operation and maintenance costs.

Although depending on scenario, contribution of these services to ROI under the use line could be greater; identifiable services are not necessarily composed to build the business processes. These services would become a part of basic services, such as infrastructure or technical services in SOA service layers.

**Agile Methodology:** Since waiting for integration of web services technologies into technical environments is not desirable, finding an acceptable balance between incorporation of service-oriented design principles and business analysis environments is a real challenge. Consequently, for many organizations it is useful to view these two approaches together in order to find a suitable middle ground. This is possible by defining a new process that allows business-level analysis to occur concurrently with service design and development. Agile Methodology which also called meet-in-the-middle approach is a combination of bottom-up and top-down strategies. This method needs to fulfill two opposite sets of requirements so it is highly complex in comparison with two primary approaches.

Agile Methodology has several advantages and disadvantages. On the positive side, it has some good points of both botom-up and top-down strategies. Furthermore, because services could be evaluated and amended again, this approach in addition to generality of services; supports long and short term requirements in the project. The most important deficiency of this method is its considerable increase in capacity of duties and activities in different project’s phases. Moreover, contrary to two primary approaches, meet-in-the-middle method does not have clear steps consequently service recognition is not easy.

**The Proposed Approach:** The proposed approach offers a straightforward and explicit method for identifying different services in SOA analysis step. This new approach has the benefits of other methods without any financial and operational overloads, in comparison to primary techniques. In order to make it easier, different services have been categorized into logical groups presenting operational state of service. Logic of organizational establishments could be divided into two primary regions: Application Logic and Business Logic. Different services in service-oriented architecture are used to represent both or either of two logics. Therefore, a collection of service taxonomies based on their operational states could be considered. The primary and most important service taxonomy for different services is as follow:

- **Application services:** It is a generic category which is used for representing services that contain logic derived from a solution or technology platform. Application services are mostly general and re-useable.

- **Business services:** This is a generic category which is used for presenting services that include business logic. These services are classified as follow:

- **Task Centric Services:** It is a business process-specific class that represents an atomic unit of process logic. These services usually have low re-usability.

- **Entity Centric Services:** This is a class of services showing one or more business dependent entities in addition to operation of related candidate to these entities.

**Process Services:** It is a category of services presenting the business process as is implemented by orchestration platform and described by a process definition. Process services reside in the orchestration service layer.

Classification of services into reasonable categories is performed in order to reach the principle of Loosely Coupling which is the main principle of service-orientation. After classification, services are separated physically so that organizational domains do not have any dependency on each other. Therefore business logic could be developed independent of application logic. The most important service layers include:

- **Orchestration Service Layer**
- **Business Service Layer**
- **Application Service Layer**

Application Service Layer, which is the lowest layer, mostly uses application services with the aim of preparing functions of re-usability. These functions are
bounded with process data related to domains and platforms of old or new application centric programs. Business service layer, using business service, could provide services which show business logic of organization.

Orchestration service layer includes a collection of one or more process services which combines application centric services with business services. Reasons behind this combination are mapping business process onto related services based on regulations and logic of business and process details and also making relations between different services of special discipline.

Proposed approach uses Top-Down and Bottom-Up methods in combination to identify and model mentioned services. This new approach has the benefits of both primary methods without any additional complexities. Identifying both centric services application and entity centric services is performed by Bottom-Up approach, while for recognizing business services and task centric services; Top-Down approach is applied. Fig. 2 illustrates a summary of the proposed method.

Proposed method includes several steps which are described in following section.

- **Identifying Business Process:** In the first step, it is necessary to determine functional domain of business process and business goals, understand the business requirements and reach the total vision of future organization. In addition, for simultaneous development of both Top-Down and Bottom-Up approaches, reusable services and assets should be recognized. Main outputs of this step are business requirements, business vision and business process.

- **Making Business Use-Case Model:** In this stage detailed process steps, business actors, business use cases, business entities, operation service candidates and business rules ought to be recognized and assigned. Principal outputs of this step are business actors, entity-business use case, business use case model and operation service candidate in addition to business rules.

- **Identifying Entity Centric Services:** Using outcomes of two last steps, operation candidates related to one or more dependent entities are identified. Then operation candidates are scaled and categorized as logical groups called entity centric services. Entity centric services are outputs of this stage.

- **Recognizing Application Services:** In this step, firstly process requirements depended on technology and application programs are separated. Afterwards, operation candidates are checked to find out which one could be performed through existing services of current organization. Selected candidates could reuse present assets based on principles of service-oriented. Moreover with small changes, they could be turned into required services. Finally, remaining operation candidates are categorized into logical groups. Each group shows the logic of special service. Outputs of this step are Application services.

- **Identifying Task Centric Services:** Business requirements, business use case model, operation service candidate and business rules are inputs of this stage. At first, logic of workflow and every possible relation are reviewed and amended. At the same time with the aim of supporting long-term goals, new business goals are defined and added to system. If it is necessary to combine the operation candidates of business services and application services, related activities are performed and after that the operation candidates connected with business duties are categorized as logical groups. Consequences of this stage are task centric services.

- **Recognizing Process Centric Services:** This step takes business process, business use case model and operation service candidate besides business rules as inputs. Since process services are considered as controller services, business logic and application logic should be separated. Also it is necessary to recognize and separate all business rules, condition rules, exception logic and sequence logic. After that it is vital to assign operation candidates of all controlling, conditions and exceptions operations. Finally, related operation candidates ought to be categorized as logical groups, each of which shows the logic of special service. Process centric services are outputs of this step.

**Case Study:** In this section, service identification steps of the proposed approach have been clarified by an example. The main goal of this example is to prepare an automatic processing, recording and confirming system of data connected with supervision of employee’s working hours in an organization to distributed units. In required services recognition and analysis phases, SOA approach has been used.

Considered organization includes a collection of supervisors performing surveillance activities of each distributed unit. When employees fill out their weekly timetable, spent time for each unit should be determined. Moreover, consumed working hours are recorded separately in each unit. If these two documents are the same, form and timetable of employees will be confirmed.
In this system some controlling tools are implemented. For example, consumed time which is recorded by an employee should not go beyond the defined time for each working unit. Also total working hours of an employee should not exceed the maximum assumed working hours. If sheet of working hours of an employee is not confirmed, it ought to be recorded in his profile and alarm messages about failure in confirmation should be sent to that employee and his manager. The working process mentioned above is divided into a collection of process steps, in order to recognize the required services. Required process steps are as follows:

1. Receive timesheet.
2. Compare hours recorded on timesheet with hours billed clients.
3. Confirm that authorization was given for any recorded overtime hours.
4. Confirm that hours recorded for any particular project do not exceed a predefined limit for that project.
5. Confirm that total hours recorded for one week do not exceed a predefined maximum for that worker.
6. If timesheet is verified, accept timesheet submission and proceed to step 11.
7. Reject timesheet submission.
8. Generate a message explaining the reasons for the rejection.
9. Issue a timesheet rejection notification message to the worker.
10. Issue a notification to the worker's manager.
11. Terminate the process.

In first step, entities and resources connected to working process are identified. Fig. 3 represents different entities and their inter-relations.

Afterwards, operation candidates are determined by assumed services \[13,14,17\]. Operations required for working process are illustrated in Fig. 4.

Operation candidate related to working entities is scaled, categorized and then active from Business process. In next step, service candidates of entity centric services could be identified as follows.

Since entities of employee history and employee are bounded together, operation candidates related to both entities could be described in terms of one service. This mater provides more flexibility and efficiency \[1\].

Separation of operation candidates bounded to application working area and application requests indicates that operations connected with notification to users and managers could also be categorized. Fig. 6 demonstrates this issue.

After that, it is also necessary to recognize services dependent to the duty which is related to performing business tasks. This action could be performed using process steps and combination of business service candidates. Receiving two entities of working unit including invoice and form of timesheet belongs to an employee, related services are combined and then recorded rates are compared. Therefore, operation candidate related to business task of form confirmation could be represented in the form of a service dependent to the task of timesheet verification. This action is shown in Fig. 7.

Finally, related controlling operation could be classified as process services through comparing the timesheet with presumed rates and also accessing data which are the components of condition and control operation of working process.

In Fig. 9 identified services in addition to layers connected with services has been represented.

Fig. 1: SOA Delivery Lifecycle
Fig. 2: Summary of the proposed method

Fig. 3: Entity Model
Fig. 4: operation candidate for services

Fig. 5: Entity Centric Services

Fig. 6: Application Services

Fig. 7: Task Centric Services
The following Figure shows the identified services and layers related to the services:

As it could be seen from Fig.9, using proposed approach, different services possessed to independent layers of architecture are recognized easily and clearly without any repeated and additional activity. After identification, services could be designed, tested, implemented and finally developed in a service-oriented area.

**Conclusions and Future Works:** SOA and service-orientation are implementation-agnostic paradigms that could be implemented by any suitable technology platform. Therefore, SOA has been introduced as one of the latest approaches in software development. This approach could be the best architecture of IT and communication industry in future. In spite of its positive points, SOA approach has some complexities and it is not completely elevated. Therefore, it would
be valuable to improve efficiency of SOA approach by decreasing its current complexities besides clarifying its implementation steps. In this paper with the aim of improving SOA approach, a new method has been proposed.

REFERENCES