Service identification and definition frameworks

Kaisa Rommel
kaisa.rommel@hut.fi

T-86.5165 Seminar on Enterprise Information Systems:
Service-Oriented Architecture and Software Engineering,
HUT, Finland

Abstract. Companies are interested in implementing Service Oriented Architecture (SOA) but they are missing the methods and the frameworks describing how they should do it. Some frameworks have been suggested. When designing SOA, the analysis and the service identification and definition phases are crucial to reach a good architectural plan. I present and evaluate some of the suggested analysis and design frameworks based on the guidance and means they offer for a starting SOA company. Two of the frameworks are described and evaluated in more detail.

Keywords: Service Oriented Architecture (SOA), design and development methodologies, service identification and definition, Service Oriented Analysis and Design (SOAD).

1 Introduction

At the moment, companies are very interested in Service Oriented Architecture (SOA). The benefits of SOA, e.g. agility, aligning business and IT processes, technology independence and maintainability, are well known and companies recognize the need for these qualities. Companies want to implement SOA and the demand for methods and frameworks defining the process, how to proceed to reach SOA, has grown. Some frameworks have already been suggested, but experiences of the use of the frameworks have been very little reported.

Service Oriented Architecture requires good planning and thorough designing so that agility, interoperability, technology independence and other benefits of SOA can be achieved. To achieve a good architectural design requires understanding of the business processes as well as of the existing IT architecture and systems. The comprehension of the whole, the big picture, is very crucial. Identifying and defining the services, their compositions and the processes they form, are the essential parts of the process of designing SOA. In this report, I will present some of the SOA analysis and design frameworks that have been suggested. I will evaluate the frameworks based on the usefulness and the guidance they offer for a company starting the SOA design; how detailed the process description is and how good means and methods do they offer. Based on these criteria, I will choose two to describe and evaluate in more detail.
First, I will present the different frameworks and after that, I will have closer look at two of them and describe the service identification and definition phases defined by these two frameworks. In the end, I will discuss and compare these two frameworks and finish with conclusions.

2 Suggested frameworks

The suggested frameworks have a lot of similarities. The phases of the processes cover more or less the same tasks. However, the frameworks also differ in many ways and therefore, they cannot be compared fully with each other. First of all, the scope of the frameworks differs a lot. Some of the frameworks define the complete life cycle process of SOA, some of them concentrate on certain phases of the process. Some frameworks emphasize some phases more and describe them in more detail and rush through the other phases. Some of the frameworks are iterative and might also cover the further development of the architecture after the initial deployment. The availability of the methods varies also. Some of them are proprietary and are not publicly available or only published in a book, which also limits the accessibility. In this report, I am bound to concentrate on the methods that are easily accessible and publicly available. I will next present shortly some of the suggested frameworks and evaluate their characteristics. The emphasis will be on the frameworks ability to cover the service identification and definition tasks.

2.1 Service Oriented Architecture Framework (SOAF)

Service Oriented Architecture Framework (SOAF) (Erradi et al., 2006) is very detailed and thorough method that defines the phases, their inputs and deliverables very systematically. The framework concentrates on the SOA analysis, design and planning phases and it consists of five phases, information elicitation, service identification, service definition, service realization and roadmap and planning. Each of these phases is divided into several sub-phases.

2.2 Service Oriented Unified Process (SOUP)

Service Oriented Unified Process (SOUP) (Mittal, 2006) is a framework that covers the whole SOA life cycle. SOUP offers two different but very similar versions of the framework, one for new SOA environment, the initial deployment of SOA, and another for an existing SOA environment, for further development and expanding the SOA architecture. Both of the versions consist of the same six phases, incept, define, design, construct, deploy and support (Figure 1). The framework defines the activities and deliverables of each phase only on the level of mentioning and does not elaborate how the activities should be carried out. In other words, the framework only describes the high level process and does not provide any methods how to execute it.
2.3 Service Oriented Design and Development Methodology

Papazoglou’s & van den Heuvel’s (2006) methodology covers also the whole life cycle of SOA. It is iterative and suites for the initial SOA deployment and for the later SOA projects developing and expanding the architecture. The methodology consists of six phases, planning, analysis and design, construction and testing, provisioning, deployment and execution and monitoring (Figure 2). After the execution and monitoring phase, the cycle can start again from the analysis and design phase. The methodology covers very superficially the planning phase and the identification of the services. Instead, it concentrates more on the definition phase and the other phases of the process.

![Figure 2: The phases of the service oriented design and development methodology (Papazoglou & van den Heuvel, 2006)](image)

2.4 Service Oriented Modeling and Architecture (SOMA)

Service Oriented Modeling and Architecture (SOMA) (Arsanjani, 2004) is probably one of the most known frameworks. It is developed by IBM and it is IBM’s proprietary method. Because of this, the detailed description of the method is not publicly available. The method covers only the most essential phases, identifying, defining and planning the implementation of the services. The method consists of three phases, identification, specification and realization. The identification phase
combines the top-down, bottom-up and meet-in-the-middle approaches and each of these approaches is a separate step of the identification phase. The top-down step is called Domain decomposition. The business domain is decomposed into functional areas, processes and sub-processes that can also possibly become services. The bottom-up step is called Existing system analysis. The existing systems and resources are analyzed and their existing features, API’s, modules etc. are evaluated. The meet-in-the-middle step, called Goal-service modeling, is supposed to find out the services that are still missing, services that were not discovered during the two earlier steps. The next phase, specification, categorizes the services to different layers, analyses the dependencies between the different sub-processes and sub-systems identified earlier, specifies the services in detail and their compositions.

![Figure 3: The phases of SOMA (Arsanjani, 2004)](image)

2.5 Service Oriented Analysis and Design Approach to Developing Adaptable Services

Chang and Kim (2007) have suggested a service oriented analysis and design approach that is especially targeted at developing adaptable services. Their concern is how to design services that are adjustable and able to answer to different client requirements in various situations now and in the future. The process that they define follows similar phases than the other frameworks, but it also suggests a method how to manage the service variability issues. The process consists of five phases, defining target services, defining unit services, planning service components acquisition, acquiring service components and composing services. The framework concentrates on the variability management and does not consider the basic tasks of service identification and definition in detail.

Next, I will describe and discuss two of the frameworks more closely, the methodology by Papazoglou & van den Heuvel and SOAF by Erradi et al. These frameworks seem to be the most detailed and they have the most means to give to the service identification and definition.
3  Service Oriented Design and Development Methodology

In Papazoglou’s and van den Heuvel’s (2006) methodology, the analysis and design phases define the identifying and defining the services and their compositions. Table 1 describes the steps of these phases.

Table 1: The steps of the analysis and design phases according to the Papazoglou’s and van den Heuvel’s (2006) methodology.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process identification</td>
<td>Service specification</td>
</tr>
<tr>
<td>Process scoping</td>
<td>Structural and behavioral specification</td>
</tr>
<tr>
<td>Business gap analysis</td>
<td>Service interface specification (WSDL)</td>
</tr>
<tr>
<td>Process realization</td>
<td>Service programming style</td>
</tr>
<tr>
<td></td>
<td>Policy specification</td>
</tr>
<tr>
<td></td>
<td>Process specification</td>
</tr>
<tr>
<td></td>
<td>Structural and behavioral specification</td>
</tr>
<tr>
<td></td>
<td>Process structure description</td>
</tr>
<tr>
<td></td>
<td>Identification of the services implementing a process</td>
</tr>
<tr>
<td></td>
<td>Activity dependencies and conditions</td>
</tr>
<tr>
<td></td>
<td>Implementation description of a process (BPEL, BPMN)</td>
</tr>
<tr>
<td></td>
<td>Roles description</td>
</tr>
<tr>
<td></td>
<td>Non-functional process concerns (SLA)</td>
</tr>
</tbody>
</table>

3.1 Analysis phase

The main tasks of the analysis phase are to determine the requirements for the new architecture, model and analyze the current business processes, define the new processes if necessary and investigate the current implementations to find out the existing services. The analysis phase is divided into four steps: process identification, process scoping, business gap analysis and process realization analysis.

The process identification and scoping steps identify and analyze the old and suggest the new processes. The scoping of the processes ensures that the phases of an individual process form a cohesive and logical chain of events that complete one task. Too wide scope decreases the re-usability of the process.

The business gap analysis compares the suggested processes and the services provided by the current software applications and intends to find the gaps where new services are needed. Different possibilities to fill these gaps are discussed and considered.
After the processes and the services are analyzed and the workload of the project evaluated, the **process realization analysis** considers the different options how to proceed with the planning and implementation. The analysis evaluates different realization scenarios in terms of risks, costs and suitability to the case environment. Different approaches are compared, like the top-down, bottom-up and meet-in-the-middle approaches.

### 3.2 Design phase

The design phase lays out the whole implementation plan. The services, their interfaces and interactions are designed. The design phase consists of two main tasks, specifying the services and specifying the processes. These two tasks are conducted simultaneously. The tasks are very similar. They define the same aspects, one for the services and the other one for the processes.

**The service specification** designs the actual services and creates the interface descriptions. The implementation of the services is not done yet at this point. The implementation is done according to the interface descriptions in the next phase, in the service construction phase, which is not described in this report. The service specification consists of the structural and behavioral specification and policy specification. The structural and behavioral specification defines the service interface, for example WSDL, and the programming style how the service will be implemented. Programming style means, for example, whether the service is document-oriented or RPC-oriented. The policy specification describes the non-functional concerns of the service, like the security and authorization issues and quality issues.

**The process specification** designs the implementation of the business processes. First task to do is describing the process structures. This means identifying the activities that form the process and the services performing them. The interface for the process itself needs to be also defined. Next, the conditions and the dependencies between the services are described. Once the processes are specified, they should be described using some business process modeling method, like Business Process Execution Language (BPEL) or Business Process Modeling Notation (BPMN). The roles of the parties involved in the process should be defined and the non-functional process concerns described, typically in the Service Level Agreement (SLA). (Papazoglou & van den Heuvel, 2006)

### 4 Service Oriented Architecture Mainframe (SOAF)

Service Oriented Architecture Framework (SOAF) (Erradi et al., 2006) defines input, activities and deliverables for each phase. The three first phases, information elicitation, service identification and service definition, contain the essential tasks that will be covered here. Table 2 describes the steps of these to phases.
Table 2: The steps of the information elicitation, service identification and service definition phases according to SOAF. (Erradi et al., 2006)

<table>
<thead>
<tr>
<th>Inputs:</th>
<th>Activities:</th>
<th>Deliverables:</th>
</tr>
</thead>
</table>
| **Information Elicitation** | • Business process modeling  
• Application portfolio analysis  
• Perform Process-to-Application mapping | • Current business process model  
• To-be business process model  
• Candidate business services  
• Business issues and non-functional requirements  
• Current technical architecture  
• Process-to-Application mapping  
• Candidate application services |
| • Business drivers, visions  
• Current application portfolio | |
| **Service Identification** | • Service identification  
✓ Define granularity  
✓ Service classification  
✓ Service matching  
✓ Rationalization | • Layered services:  
✓ Business services  
✓ Application services  
✓ Atomic application services  
✓ Enterprise services  
✓ Line-of-business services  
✓ Infrastructure services  
• Compositions and choreography |
| • Business use cases  
• Process-to-Application mapping | |
| **Service Definition** | • Service descriptions:  
✓ Contracts  
✓ Data models  
✓ Interface  
✓ Policies  
✓ Service composition  
✓ Inputs/outputs | • Service descriptions  
• Service non-functional requirements |
| • Information models | | |
4.1 Information elicitation

The most important task of the information elicitation phase is to model the current state of the business processes and the application portfolio. First, the existing processes are modeled and analyzed. Based on the analysis, the new improved processes are modeled and candidate business services identified. The new processes are mapped to the existing software assets. The mapping will discover the functionalities needed in different processes, possible redundancies, gaps and shared services. The application portfolio analysis evaluates the current applications and their possibilities and suitability for the new architecture. The framework presents a scoring model that can be used to the application assessment. Also the candidate services offered by the current applications are listed.

4.2 Service identification

The service identification phase of SOAF combines the top-down and the bottom-up approaches of identifying the services. The candidate business services and the candidate application services are combined and mapped to discover unused and unmet services (Figure 4). The services are iteratively refined by taking in consideration the granularity, the classification of the services to the service layers and rationalization of the service composition. The framework suggests some means how to define the right granularity of the services.

Figure 4: Service identification in SOAF (Erradi et al., 2006)

4.3 Service Definition

The service definition phase conducts the same tasks as Papazoglou’s and van den Heuvel’s methodology in the service specification phase. The interface descriptions,
the data models and messages of the services are defined, the behavior and the
conversation models of the services are decided and the non-functional requirements
described. (Erradi et al., 2006)

5 Discussion

The two frameworks, SOAF and the Papazoglou’s and van den Heuvel’s
methodology, both use the meet-in-the-middle approach to identify the services. They
also perform almost the same tasks but they differ in the way they divide the tasks
under different phases. For example, in the design phase, Papazoglou and van den
Heuvel emphasize also the process definition tasks. In SOAF, the processes are
defined already in the earlier phase, in the service identification phase.

However, the biggest difference between the frameworks is the emphasis on
different tasks. SOAF gives detailed description of the analysis and the service
identification phases. The Papazoglou’s and van den Heuvel’s methodology, instead,
describes the service definition phase in detail. Papazoglou’s and van den Heuvel’s
methodology goes over the analysis phase very quickly and gives very little tools for
the identification of the services. The emphasis in the methodology is in the design
phase and the phases coming after that. The design phase is described more in detail
and it does give concrete means how to proceed with the definition of the services.
SOAF gives very detailed framework how to proceed with the analysis and the
identification of the services and suggest several methods how to conduct certain
tasks. In SOAF, the service definition phase is handled very quickly without going
into details, but the task are the same than in the Papazoglou’s and van den Heuvel’s
methodology and that can be also found in other literature, e.g. in Alonso et al.
(2004).

One could ask do these models support the way SOA is implemented in most cases
nowadays. Very rarely SOA is rolled out to cover the whole IT infrastructure at once.
This would not even be smart. The project would be huge, long and the risks and the
costs would be high. In addition to that, the managers often want to see quick results.
It is considered to be smarter to move in smaller portions, one piece at a time.
However, this kind of advancing requires a good design and architectural plan for the
whole architecture before starting, even though the plan would be implemented step
by step. The lack of the preliminary blueprints may cause issues later, when the
pieces of the puzzle do not match. The frameworks seem to consider this quite poorly.
Papazoglou’s and van den Heuvel’s methodology is meant to be iterative and
therefore, it should support also the later development of the SOA, but Papazoglou
and van den Heuvel hardly discuss the iterative nature of the process. In addition, the
iterative circle continues from the analysis and design phase. However, the
architecture cannot be designed piece by piece. Some kind of preliminary plan of the
whole architecture should be in place after the beginning of the first cycle. Earlier
presented SOUP framework, as a process, might offer opportunity for this kind
development approach, but Mittal does not discuss the matter either. It is difficult to
evaluate how well the frameworks would work for this kind of scenario, because very few case studies reporting the experiences of the frameworks have been published.

6 Conclusions

Several frameworks have been proposed for the SOA analysis and design process. I have presented some of them and described two of them in more detail. The frameworks differ in their scope and in the tasks they emphasize. However, the same basic tasks can be identified in all of the frameworks. These similarities could be summarized with a general service identification and definition process (Table 3). The process consists of 3 phases: analysis, service identification and service definition. The analysis phase includes the tasks of identifying the current business processes, defining the improved business processes and analyzing the current application portfolio. The service identification phase includes the tasks of identifying the candidate business services, identifying the candidate application services, matching the candidate services, filling the service gaps, categorization and rationalization of the services and defining the service compositions. The service definition phase consists of the structural and behavioral specification of the services, interface descriptions, data model, message and policy descriptions.

Table 3: The general service identification and definition process.

<table>
<thead>
<tr>
<th>Analysis</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identifying the current business processes</td>
<td></td>
</tr>
<tr>
<td>• Defining the improved business processes</td>
<td></td>
</tr>
<tr>
<td>• Analyzing the current application portfolio</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service identification</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identifying the candidate business services</td>
<td></td>
</tr>
<tr>
<td>• Identifying the candidate application services</td>
<td></td>
</tr>
<tr>
<td>• Matching the candidate services</td>
<td></td>
</tr>
<tr>
<td>• Filling the service gaps</td>
<td></td>
</tr>
<tr>
<td>• Categorization and rationalization</td>
<td></td>
</tr>
<tr>
<td>• Defining the service compositions</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service definition</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Structural and behavioral specification of the services</td>
<td></td>
</tr>
<tr>
<td>• Interface descriptions</td>
<td></td>
</tr>
<tr>
<td>• Data models</td>
<td></td>
</tr>
<tr>
<td>• Messages</td>
<td></td>
</tr>
<tr>
<td>• Policies</td>
<td></td>
</tr>
</tbody>
</table>

Because of the different scopes and emphasis, it is impossible to say which one of the frameworks would be the best and the most mature. By combining the best parts of the frameworks, it is possible to come up with a framework that is complete and detailed guide through the SOA designing process. Naturally, one framework does
not suite all, but by combining parts from here and there, enterprises can tailor themselves a framework that fits their environment and requirements.
7 References


