Describing case studies and classifying research approaches

Antonio Bucchiarone, Raman Kazhamiakin, Valentina Mazza, and Pierluigi Plebani

Abstract This initial chapter aims at providing a useful introduction and reference point to the research described in the following chapters. First of all, to provide a homogenous description of the existing approaches on service engineering, we introduce a common case study referring to the telecommunication domain. This case study will be used in the other chapters of this book to motivate and describe the presented approaches. The way in which the case study is described follows the approach developed in the S-Cube Network of Excellence. In addition, we also provide a classification of the research results proposed in the rest of the book, by relying on the S-Cube Integrated Research Framework. Such a classification allows the reader to have an idea about how the contributions deals with research in the service engineering field.

1 Introduction

Case studies can be described in various ways depending on their purposes. For instance, they can describe a specific development or proof of concept using a specific technology, or they can simply describe an application case without offering a specific implementation solution. Of course, while in the first case the use case description contains also design, implementation, and even deployment and operation details, in the second case it should be implementation and technology agnostic.

Antonio Bucchiarone and Raman Kazhamiakin
FBK-IRST
Via Sommarive 18 - 38050 Trento, Italy
e-mail: [bucchiarone,raman]@fbk.eu

Valentina Mazza and Pierluigi Plebani
Dipartimento di Elettronica ed Informazione, Politecnico di Milano
Via Ponzio 34/5 - 20133 Milano, Italy
e-mail: [vmazza,plebani]@elet.polimi.it

Since, of course, we are thinking of case studies supported by software, the description should focus on *what* expectations the software should address more than on *how* these should be addressed. In other terms, the description should be focusing on eliciting those *goals* and *assumptions* that the software should address.

As, the term case study has been used in the literature to mean a problem together with a specific solution, the goal of this chapter is to introduce two results from the S-Cube Network of Excellence [1]. On the one hand, in Section 2 we introduce the case study description approach proposed in the S-Cube project [1]. By relying on this approach, this chapter describes a common case study that will be referred by the contributions in this book presented in the next chapters. On the other hand, in Section 3 we discuss the S-Cube Integrated Research Framework (IRF): a coherent holistic framework that integrates the principles, techniques, methods and mechanisms provided by the research activities done in the S-Cube project. Using the IRF, we are able to classify and compare the solutions about the described case study [2].

2 Mobile phone services portability case study

The case study introduced in this chapter and used all along the book is about *mobile phone services portability*.

Voice and data services are provided by a set of mobile telecommunication companies around Europe. There companies are usually called Cell Phone Operators (CPOs). Some of these CPOs operates in a single country, whereas some others are big CPOs that have branches in several countries. Inhouse services are offered by the CPO to manage all the customers and the stipulated contracts. Moreover the customers can view personal information about their contracts accessing to a suitable services made available by the telephone operator, or some user could query all the services offered by the CPO accessing its public services. Not all the CPOs have their own telecom infrastructures; some of them rent the network services from big telecom companies and provide a service to the users. Such operator are called Virtual Operators (VOs).

The case study presented in this chapter is about the additional services that a company support to allow the portability of the phone number and the portability of services.

The possibility to change a mobile phone operator without changing the mobile phone number is one of the mandatory services that a mobile phone operator must provide. National and European laws regulate this procedure with the aim of allowing the customers to freely select the best company according to their requirements to be advantaged by the more suitable or cheapest services offered by the new phone operator, without advising his contacts of a new telephone number.

It might also happen that the portability is only about the services on which a customer is subscribed. For instance, a customer might move from an European

[1] S-Cube NoE Web site: http://www.s-cube-network.eu

Country to another for a limited period, e.g., one year, but keeping his subscription, even with a new phone number, with the company and using the same services already available but not by roaming. At the end of such period, the customer could move back to his home country again, picking up his original telephone number and subscription. Starting from this description of the case study, we could identify the following stakeholders:

- Cell Phone Operators (CPOs)
- Customers

In order to make case studies comparable and easy to understand, S-Cube have defined a case study description approach that leverages from the results achieved by NEXOF-RA [3] and from the Requirements Engineering literature [4]. The usage of such approach for revising and describing all cases has been very useful to highlight inconsistencies and to identify those aspects in the case studies that cover the elements considered relevant for the research.

The proposed approach is adopted in this chapter to describe in a more systematic way the mobile service portability case study that is used along the book. In more detail, the case study is described in terms of:

- A list of *Business Goals* and *Domain Assumptions* for the case study.
- A *Case Study Domain Description*.
- A list of *Scenario Descriptions*.

2.1 Business Goal and Domain Assumptions

Business Goals and the Domain Assumptions define the functionalities and the properties of both the machines and the environment where the systems operate. Whereas the business goals state the functionalities of the product, the domain assumptions report properties or restrictions of the system. For each Domain Assumption and Business Goal a table reports a description of it, the stakeholders involved, the rationale, the priority and the material supporting the description, if any.

To better understand the domain of the case study, it would be useful a glossary to explain some specific terms that could be unknown outside the specific domain. Often the glossary is not enough, consequently the description should be enriched using UML diagrams [5], or Entity-Relationship diagrams, and so on.

2.1.1 Business Goals

Business Goals report all the functionalities of the system; referring to this case study we could define the following goals:

- To provide mobile phone number portability
- To provide mobile services portability

- To satisfy national and european rules
- To make the portability as much transparent as possible to the customer
- To provide new added-value services

The following tables describe each listed business goal.

Table 1: To provide mobile phone number portability

Field	Description
Unique ID	TELCO_BG_01
Short Name	To provide mobile phone number portability
Type	Business Goal
Description	The system shall offer some capabilities to permit the portability of a mobile phone number across different CPOs. Service portability makes possible to the customers change the telephone operator keeping their old phone numbers. The customer could decide to move to a certain CPO, since the charges it has for a particular service are more convenient for him. The customer would avoid to have a new telephone number, since he should have to notify all his contacts of the changed number.
Rationale	Satisfy the need of a customer that want to take advantage of the services of a certain CPO without change his old telephone number.
Involved Stakeholders	CPOs and customer
Priority of accomplishment	Must have

Table 2: To provide mobile services portability

Field	Description
Unique ID	TELCO_BG_02
Short Name	To provide mobile services portability
Type	Business Goal
Description	The system shall enable a customer moved to another country to use the service already subscribed in the former country. Maybe not all services are available, but services could be composed by existing ones or even, if no other options are available, the customer could be linked to his home country for this service (with an extra-fee).

Rationale	Satisfy the need of a customer that want to take advantage of the services of a certain CPO when moving in the countries where CPO operates.
Involved Stakeholders	CPOs and customer
Priority of accomplishment	Must have

Table 3: To satisfy national and european rules

Field	Description
Unique ID	TELCO_BG_03
Short Name	To satisfy national and european rules
Type	Business Goal
Description	European and National goverments strictly regulate some of the services provided by the CPO. As a consequence, several regulations had been approved especially for making the change of a mobile operator as simple and economic as possible. Since these rules change during the time, the system should be able to promptly react.
Rationale	Enable a quick reaction in case of new European and National regulations.
Involved Stakeholders	CPO
Priority of accomplishment	Must have

Table 4: To make the portability as much transparent as possible to the customer

Field	Description
Unique ID	TELCO_BG_04
Short Name	To make the portability as much transparent as possible to the customer
Type	Business Goal

Description	To improve the customer satisfaction the portability of services and numbers should be transparent to the user. As a consequence, the user can interact either directly to the system by a Web application or supported by a company employee to apply for the number portability. The customer should realize that the number portability occurred only at the end of the process. Only in case of problems, the customer will be contacted.
Rationale	To improve the customer satisfaction
Involved Stakeholders	CPOs and customer
Priority of accomplishment	Should have

Table 5: To provide new added-value services

Field	Description
Unique ID	TELCO_BG_05
Short Name	To provide new added-valuee services
Type	Business Goal
Description	The system shall enable CPOs to easily provide new services. Mianly regarding on the evolution of the Web and the spread of new application under the 2.0 umbrella, CPOs need to increase the number of services by considering all the users that consume this kind of applications.
Rationale	Increase the market rate.
Involved Stakeholders	CPOs and customer
Priority of accomplishment	Should have

2.1.2 Domain assumptions

Domain assumptions report all the properties and the constraint of the system. For the considered case study we could define the following domain assumptions:

- Distributed companies
- Service provisioning

Table 6: Distributed companies

Field	Description
Unique ID	TELCO_DA_01
Short Name	Distributed companies
Type	Domain assumption
Description	The CPO consists of multiple departments, possibly spread over multiple nations. In-house Services are shared among departments: e.g., CRM, Billing, and so on.
Rationale	
Involved Stakeholders	CPO
Priority of accomplishment	Must have

Table 7: Service provisioning

Field	Description
Unique ID	TELCO_DA_02
Short Name	Service provisioning
Type	Domain assumption
Description	The system shall provide some useful services the customer could access. After an authentication phase the customer should access to personal data and get information about the contracts, the billing and so on. He could decide to change his contract choosing a more suitable telephone charges, or he could have a service to pay the billings.
Rationale	Make available services to the customers.
Involved Stakeholders	CPO and customer
Priority of accomplishment	Must have

2.2 Domain Analysis

About the domain description of the case study, Strategic Dependency Diagrams (SDDs) [6] and/or Context Diagram (CD) [4] are used. In particular, the SDDs are used to model the dependencies among the actors involved in the organisational context. Dependency edges in the diagram link the actors with needs (dependers) to actors with the capability of satisfying those needs (dependees). The needs are expressed in terms of goals (positioned on the edges). In addition to the SDD it could be useful to enhance the description with the CD. In such a diagram the boxes represent all the active entities of the case study, while the direct arrows represent the phenomena between the agents. The source of the edge is the controller of the phenomena, while the destination is the agent monitoring the phenomenon. In the diagram the system under study represents a particular agent.

2.2.1 Strategic Dependency Model and Context Diagram

Figure 1 illustrates the strategic dependency diagram of the case study. The diagram puts in evidence the business goals (the ellipses in the diagram) shared among the actors (the yellow circles in the diagram) of the scenario, showing the dependencies among them. The CPOs offer their services satisfying the national and the european rules; their goal is the provisioning of services useful to the customers in order to get information about the status of their contracts and/or to change them. In particular the current case study focuses on the provisioning of the telephone number portability service; such service should be offered to the customer in a transparent way.

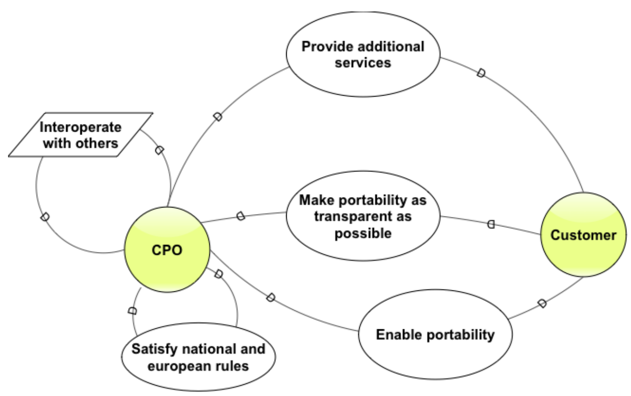


Fig. 1 Strategic Dependency Diagram

Figure 2 illustrates the context diagram of the current case study. In the context diagram, all the actors that appear in the business goals and scenarios are agents.

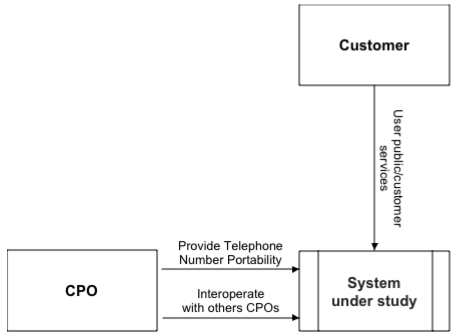


Fig. 2 Context Diagram

2.2.2 Domain model

Figure 3 illustrates the domain model of the current case study. A UML notation is used to represent the entities and the actors involved in the case study and the relationships among them. The CPO stipulates a *Contract* with a *Customer*. A contract is characterized by the particular *Fares* to pay for the service usage; moreover it could include the basic *Voice Service* or some additional service (*Value Added Service*) provided by the CPO, such as e-mail services, SMS services and so on. When a Customer requires a *Telephone Number Portability* of his telephone number moving from a CPO to another one, such portability has associated a contract. Moreover in the diagram is put in evidence the possibility of a Customer to access to different type of *Services* provided by the CPO.

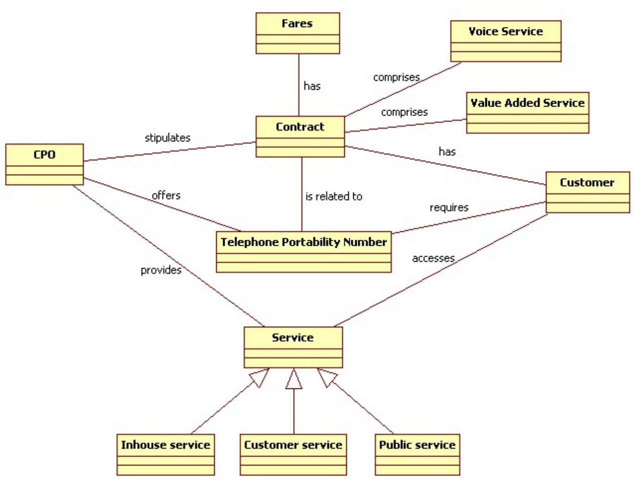


Fig. 3 Domain model

2.3 Scenarios

Finally, the scenarios description includes the phenomena shared between the world and the machine. Here, a scenario is described using a table containing information about the business goals or the domain assumptions they refer, the operational description of the scenario, the eventual problem involved and the supporting material.

The following scenarios describe the system for the management of Mobile Phone Service Portability; they assume the different departments of a CPO cooperating among them and with the other CPOs. Scenarios report the basic process behind the request of the telephone number portability among different CPOs and the portability for the same CPO in different countries.

Figure 4 shows the general use-case diagram for the case study we are considering.

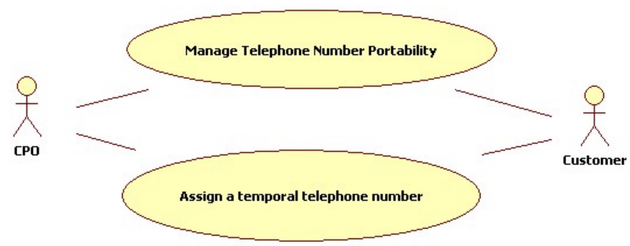


Fig. 4 Use Case Diagram

Table 8: Manage Telephone Number Portability

Field	Description
Unique ID	TELCO_S_01
Short Name	Manage Telephone Number Portability
Related to	TELCO_BG_01, TELCO_BG_02, TELCO_BG_03, TELCO_BG_04
Involved Actors	CPO, Customer
Detailed Operational Description	When a customer requires a telephone number portability, the CPO has to provide not only the porting of the number, but also the porting of the services enabled on it, when possible. After checking the portability of the number the CPO executes the porting. At the end of the process the number is activated and bound to the new CPO.
Problems and Challenges	Problems and challenges in this scenario are mainly related to the integration among the CPOs and the departments.

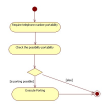
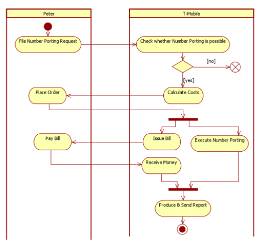
<table>
<tr><td>Additional Material</td><td>The following diagram describes the process:

</td></tr>
</table>

Table 9: Temporary mobile services portability

Field	Description
Unique ID	TELCO_S_02
Short Name	Temporary mobile services portability
Related to	TELCO_BG_01, TELCO_BG_02, TELCO_BG_03, TELCO_BG_04
Involved Actors	CPO, Customer
Detailed Operational Description	It could be happen that, for a limited period, a customer has to move to another country. During the stay, the customer would avoid to pay the roaming charges when receive or make a phone call, maintaining his CPO; so he would have assigned, from the CPO, a new temporal telephone number in the new country. When, at the end of such period, the customer comes back to his home country, he would take possession of the original telephone number and subscription. In addition to the new telephone number, the customer can also port additional services as, for instance, SMS service and UMTS service. The porting of all of these services are independent each other. So, the customer can ask to port only some of them.
Problems and Challenges	Problems and challenges in this scenario are mainly related to the integration of the departments of the same CPO, located in different countries.

<table>
<tr><td>Additional Material</td><td>The following diagram describes the process where:

In order to port a number to another subsidiary of the CPO, the customer, firstly, has to file a request.
Thr CPO checks whether it is possible to port the number and any additional services and calculates the associated costs. In this case, the CPO calculates 10 for porting the phone number and the SMS service and 50 for porting the UMTS service.
The customer decides that he only wants to port his phone number since the porting of the UMTS service seems to be too expensive for him.
In parallel to the execution of this porting a bill is issued and the money arrival is monitored.
At the end of the process, the customer receives a report, which confirms the complete-ness of this transaction.



</td></tr>
</table>

Table 10: Social network integration

Field	Description
Unique ID	TELCO_S_03
Short Name	Social network
Related to	TELCO_BG_03, TELCO_BG_05
Involved Actors	CPO, Customer

Detailed Operational Description	The innovation department of the telecom company desisers to launch a new service which offers their customers a promotion to see free pay-per view movies. Customers can send up to 10 invitations to their friends of a social network to watch one free movie per invitation. Customers can customise the invitation, selecting the friend from the social network, one movie and including some text. Friends can accept the invitation, which includes providing some marketing info. Once the friends accepts the invitation, the customer is also allowed to see the film. The service will provide valuable information about the success of the promotion, providing different reports based on the collected marketing data.
Problems and Challenges	Problems and challenges in this scenario are mainly related to the integration of CPOs and social network applications
Additional Material	none

3 Research results classification

As a second main contribution of this chapter, we introduce the S-Cube IRF as a way for organizing the research challenges and results about Service-based Applications (SBAs). Focusing on the Service Engineering perspective, this approach will be used in Section 3.3 to classify the results that are introduced in the next chapters.

3.1 S-Cube Integrated Research Framework

The IRF for SBAs of the S-Cube, aims to integrate, align, and coordinate the joint research activities undertaken within the project. To achieve this, IRF provides a coherent and holistic view on the principles and mechanisms for SBAs. In particular, the IRF should encompass those aspects of the research that are cross-cutting and defines proper interfaces among the research components of the overall conceptual network architecture. To continuously coordinate and align the research roadmap of the project as a whole, the framework is continuously validated with the help of the industrial case studies. The ultimate goal of the validation is to revise and improve the IRF.

More specifically, the IRF may be represented with the eight macro elements that are clustered into four conceptual blocks as represented in Figure 5:

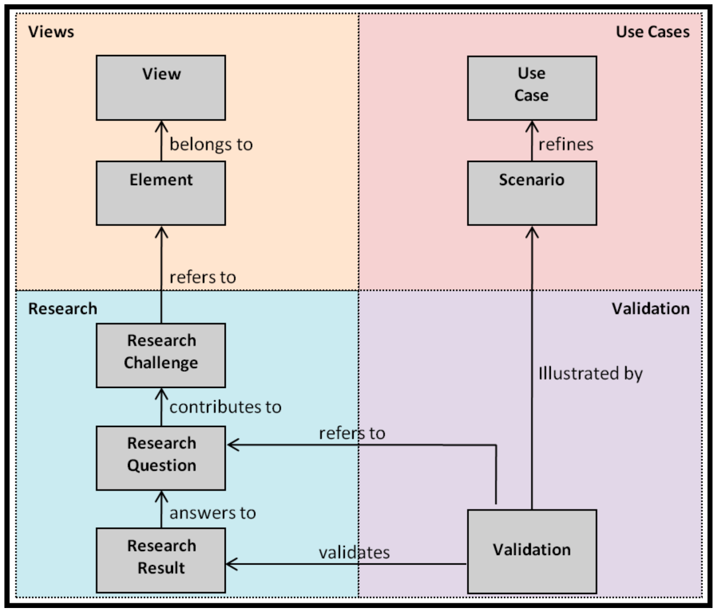


Fig. 5 Structure of the IRF

- The "Views" block represents the different perspectives that are considered by the S-Cube research activities and cover various aspects of the latter with respect to the SBAs. These views, namely "Conceptual Research Framework", the "Reference Life-cycle", "Logical Run-time Architecture", and "Logical Design Environment", are the most stable elements of the IRF. They also define the key *elements* of the framework, such as the blocks of the conceptual framework and their relations, phases of the SBA life-cycle, or modules of the logical run-time architecture.
- The "Research" block defines the objectives and results of the S-Cube research activities layered in *Research Challenges* (i.e., long-term research goals to form the S-Cube roadmap), *Research Questions* (i.e., specific short-term research objectives), and the *Research Results* (outcomes and achievements of the research efforts that aim to answer those questions).
- The "Use Cases" block defines the industrial case studies developed by the S-Cube consortium in a strong synergy with the industrial partners of the project. These case studes are the essential elements for the continuous validation of the research efforts of the project. On the one hand, the case studies are used to illustrate the research challenges of the IRF and to motivate specific research problems. On the other hand, they are exploited to validate the research outcomes. This is achieved through the definition and evaluation of the specific scenarios emerging from those use cases and industrial application domains.

- Finally, the “Validation” block defines the actual process of the IRF validation. Specifically, the validation captures the specific goal of the validation activity, the scenario used in that activity, the set up, the validated result, and the outcome of the validation process. Based on those outcomes, the research activities, as well as the IRF, are continuously revised and improved.

3.2 S-Cube Research Challenges

The research agenda of S-Cube covers a wide range of the long-term objectives in the area of the SBAs, in general, and of the Service Engineering in particular. To illustrate those objectives, namely research challenges in the terminology of the S-Cube IRF, we present the S-Cube Conceptual Research Framework that corresponds to one of the IRF views. We classify the research challenges according to the elements of this framework.

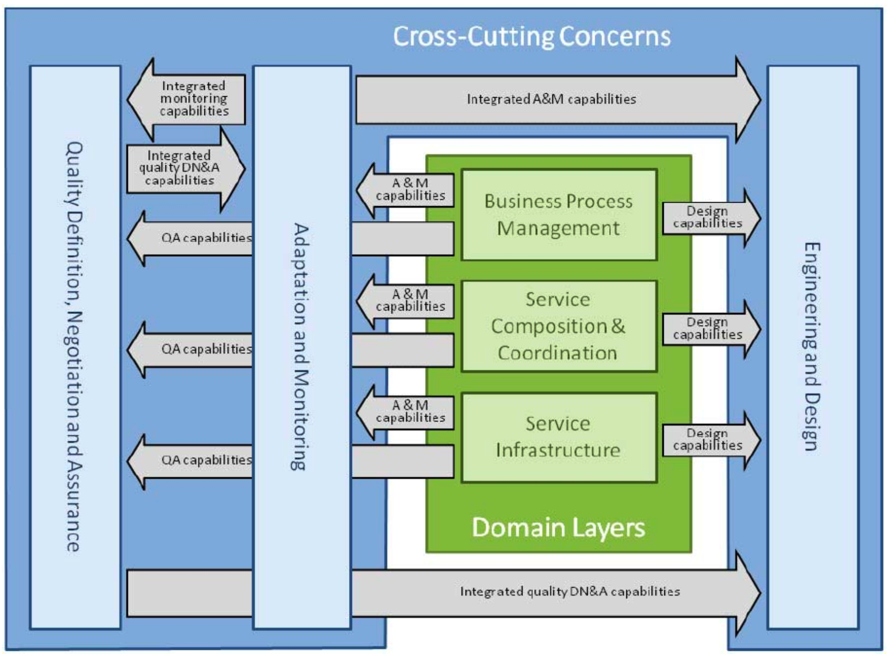


Fig. 6 S-Cube Conceptual Research Framework

As it is shown in Figure 6, the conceptual research framework consists of six related blocks, where horizontal components define “traditional” domain layers of SBAs (Service Infrastructure, Service Composition and Coordination, and Business

Process Management) and the vertical components defines the cross-cutting issues (Engineering and Design, Adaptation and Monitoring, Quality Definition, Negotiation and Assurance) crucial for the SBA research udertaken in S-Cube. In this way, the framework systematically addresses cross-cutting issues making explicit the knowledge of the horizontal layers that is relevant for these issues, and that currently is mostly hidden in specific and isolated languages, standards, and mechanisms. To do this, the used approach is that the domain layers offer (design, monitoring, adaptation, verification) capabilities that are relevant for the cross-cutting issues. The vertical components define over-arching principles and methodologies by exploiting in suitable ways the capabilities exposed by the horizontal components.

With respect to the S-Cube research agenda, these components define a wide range of the research challenges, where the Service Engineering holds a primary role:

- The "Service Infrastructure" layer studies a high-performance execution platform supporting adaptation and monitoring of SBAs (e.g., self-* mechanisms). Specifically, the challenges addressed here include the problem of *multi-level and self-adaptation*, novel mechanisms and techniques related to the *deployment and execution management*, and approaches that exploit *process mining for service discovery*.
- The "Service Composition and Coordination" layer focuses on novel service composition languages and techniques. In particular, the research here targets new *formal models and languages for QoS-aware service compositions*, the problem of *monitoring of quality characteristics*, as well as *analysis and prediction* of these characteristics, and *QoS-aware adaptation in service composition*.
- The "Business Process Management" layer addresses the aspects related to the modelling, designing, deploying, monitoring and managing of service networks, business processes and Key Performance Indicators (KPIs). The main research focus here is on the development, analysis, simulation, and management of *end-to-end processes in Agile Service Networks*, and on the novel concepts for the monitoring and validation of *business transactions in Agile Service Networks*.
- The "Engineering and Design" concern covers the issues related to the life-cycle of services and SBAs (i.e., *definition of a coherent life cycle for adaptable and evolvable SBA* and *measuring, controlling, evaluating and improving the life cycle and the related processes*), to the problem of adaptability (i.e., to *understand when an adaptation requirement should be selected*), and to the specific aspects in the design of such systems (*HCI and context aspects*, *exploiting the concept of SBAs in the internet of things setting*).
- The "Adaptation and Monitoring" concern covers the issues related to the adaptation of a SBA. Addressing the problem that is cross-cutting to the technology layers, here the research challenges include *comprehensive and integrated adaptation and monitoring principles, techniques, and methodologies*, the problem of *proactive adaptation and predictive monitoring*, studying the role of human actors in the adaptation process (*context and HCI aspects in the monitoring and adaptation*, *mixed initiative SBA adaptation*).

- The “Quality Definition, Negotiation and Assurance” concern involves principles and methods for defining, negotiating and ensuring quality attributes and Service Level Agreements (SLAs). Starting from the definition of comprehensive, *end-to-end quality reference model* and *rich and extensible quality definition language*, this component addresses such problems as *exploiting user and task models for automatic quality contract establishment*, *proactive SLA negotiation and agreement*, *run-time quality assurance techniques* and *quality prediction*.

3.3 Research Results in Service Engineering

In this section we introduce briefly the research results in Service Engineering that will be described in detail in the rest of the book, by relying on the IRF presented in the section above. The classification has been done using the research challenges introduced before and the output is depicted in Figure 7. It shows which aspect of the S-Cube research framework is covered by the approaches. In the following paragraphs we better explain this coverage.

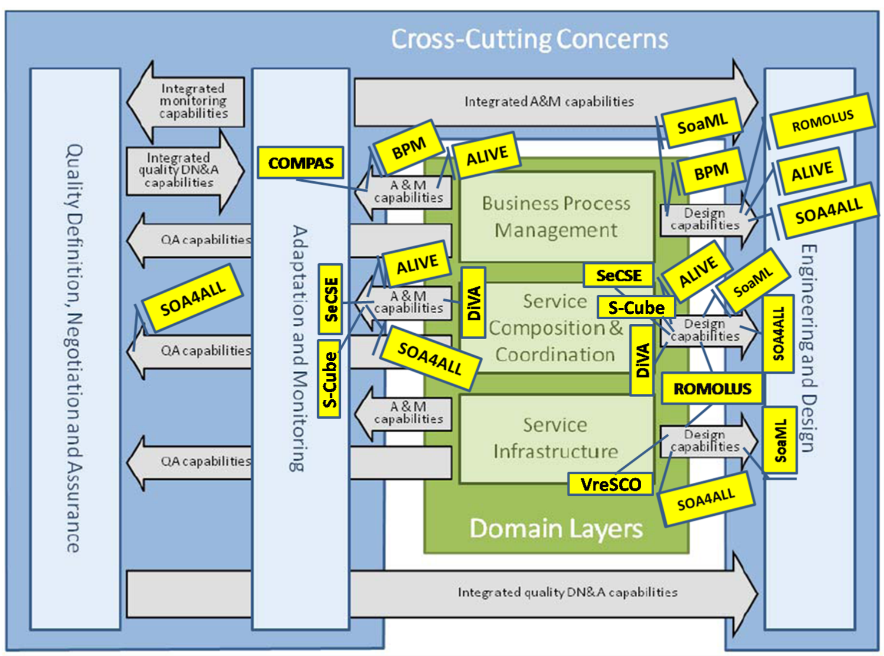


Fig. 7 S-Cube Conceptual Research Framework with mappings

Chapter 2. S-CUBE

The adaptation of a service-based application (SBA) can address various goals such as corrective and perfective adaptations. In the first case the faults contained in the SBA are repaired while in the second case the SBA is adapted to new and yet unknown requirements. In [**?**] a framework to integrate and align perfective and corrective adaptations is presented. The framework uses requirements engineering techniques to trigger perfective adaptation and online testing techniques to trigger corrective adaptations. Based on these two techniques, the framework is also able to integrate these two techniques to support the overall adaptation requirements of the Service-based applications. Both of the above techniques share the characteristics that they are pro-active in nature, i.e., both techniques lead to ”predictive” adaptation triggers. In the case of online testing, the failure of a service could point to a problem of the SBA in the future. In the case of requirements engineering, this provides the possibility to improve the SBAs and to anticipate future requirements. Finally

Chapter 3. COMPAS

In [**?**] the authors propose an approach to monitor business processes with the objective to check the conformance of an organizations business activities and practices with existing laws, regulations and its own internal policies. This is a result inside to the COMPAS project that has as main aim to design and implement novel models, languages, and an architectural framework to ensure dynamic and on-going compliance of software services to business regulations and stated user service-requirements. The main result of this project is an event-based monitoring framework for business processes at runtime. The monitoring proposed is model-aware, in the sense that it can access and reflect on process models at runtime. High-level events (that correspond to business events), containing references to the process models, are recognized from low-level process events using complex event processing techniques. The model references enable runtime retrieval and reflection on the original process models. As a consequence, the size of the events is kept small and (new) models and model elements can be considered during monitoring.

Chapter 4. SoaML

The goals of Chapter **??** are to support the activities of service modeling and design and to fit into an overall model-driven development approach. In it a UML profile has been defined to support the range of modeling requirements for service-oriented architectures, including the specification of service implementations. This is done using an automatic artifacts generation following an Model-Driven Architecture (MDA) based approach. The main objective was to provide a foundation for Model-Driven Service Engineering (MDSE) based on the MDA approach. A

MDSE supports the activities in service engineering by developing models and performing activities on them such as transformations, simulation and validation. Its main challenge was to define a consistent and comprehensive top-down approach which supports service modeling starting from higher models such as goal models, requirements models or business process models down to the modeling of services and their realization. The methodology guides developers through different phases of modeling to identify services within a service-oriented architecture, covering aspects such as service contracts, composition and integration. Starting from the Business Architecture Model (BAM) which includes goals, business processes, business informations and organization modeling the approach proceeds identifying the Service Architecture Model (SAM). The latter specifies the services architecture, the service contracts, service interfaces and message types, service behavior and component models. The mapping from BAM to SAM is done through service capabilities modeling and/or using model transformation techniques. A platform-specific model (PSM) is specified along with a model of the executable services. Finally, the mapping from SAM to PSM is support by model-to-model, model-to-text and code generators.

Chapter 5. ROMOLUS

In [**?**] a service development approach, based on Domain Driven Design (DDD), has been proposed. Principally it uses a common language which describes the application model, assisted by graphical notations, and maintaining the model very closed to its implementation. It proposes a precise method to develop services that is organized in layer where each layer depends only on the layers below. These layers are: User Interface, Service, Domain and Infrastructure. The *Infrastructure Layer* provides suppport to the rest of layers such as persistence of the domain, messaging, logging or network I/O. The *Domain Layer* represents the business logic of the application. The *Service Layer* exposes services of the domain layer while the *User Interface Layer* is responsible for showing the information to the user. Moreover a precise design approach is defined that starts from the service identification to the hend user programming and customisation of services through the domain design of services and aspect identification, development and testing. The main challenge of this project is to use the DDD approach to develop a Service and how to compose, customize and extending services using mashup technology. A Mashup Builder has been developed and provides a graphical user interface for building mashup easily, allowing the user to develop mashup faster. It allows the user to combine information obtained of different services.

Chapter 6. SeCSE and Gredia

In [**?**] authors present main results of the two projects SeCSE and Gredia. More precisely they introduce a framework to design and adapt service-based systems.

From the design point of view they show the iterative design process supported by the framework. It uses structural and behavioral design models of service-based systems to support discovery of services that can fulfill the models. The identified services are used to reformulate the design models and trigger new service discovery iterations. Finally, UML class and sequence diagrams are used to define structural and behavioral models. From the adaptation point of view it supports execution time adaptation using special servers and listeners to allow notifications of changes in services and application environment. To adapt service-based applications a process is used that allows services to be identified based on both pull and push modes of query execution. The pull mode of query execution is performed to identify services while the push mode is performed when the application is running and a service needs to be replaced.

Chapter 7. BPM

In Chapter **??** authors consider the business process management (BPM) discipline where a service is seen as an encapsulation of a business process. They provide a *refinement technique* of business processes. A refinement can be generally seen as a technique which allows to add more information to a model on a certain abstraction level while preserving the original information and constraints on a more abstract level. The refinement technique is especially useful when several people with different expertise and responsibilities work on the same business process. In this paper two kinds of refinement have been defined: horizontal and vertical. *Horizontal refinement* is a transformation from an abstract to a more specific process which contains the composition of activities. It is also called decomposition. *A vertical refinement* is a transformation from a principle behavior model (i.e., BPMN processes) of a component to a process model.

Chapter 8. ALIVE

In Chapter **??** an architecture for the deployment and management of dynamic, flexible and robust service-based applications is proposed. Its main challenge is to create a framework for services engineering addressing the new reality of open systems of active services. The proposed architecture is composed of three layers for the design and management of distributed systems, namely: *services*, *coordination* and *organization*. The *service layer* extends existing service models to make them aware of their social context and of the rules of engagement with other services. It is concerned with the description of services, the selection of appropriate services for a given task and the execution and monitoring of services associated with a given organization. The *coordination layer* provides the means to specify, at a high level, patterns of service interactions using a variety of powerful coordination techniques based on agent technology. The *organizational layer* resides above coordination, providing a social context for the coordination and service levels. This level specifies the rules that govern interaction and defines the system on the basis of goals and results, abstracting away from the specific actions used to accomplish them. This engineering approach does not provide a coherent life-cycle to design, realize and maintain services and SBA but only a set of modeling approaches for each layer with the support to transform a model in another model or models into text using the Model Driven Development principles. Unlike the organizational layer, the coordination and service layers of the system are supposed to adapt to the current situation. Regarding the adaptation challenge the framework proposes a way to monitor the service level and adapt it at run-time. Other levels react to this adaptation with re-structuring but they do not provide separate monitoring and adaptation strategies and mechanisms. No cross-layers monitoring and adaptation approaches are provided.

Chapter 9.DiVA

The DiVA [**?**] approach provides a tool-support methodology for managing dynamic variability in adaptive systems. It considers an adaptive system as a Dynamic Software Product Line (DSPL) and focuses on the variability of the system, rather than on the whole set of its possible configurations. This involves the identification and modeling of variation points in the system and the subsequent system refinement into elements of variability and commonality. Each variation is consolidate into a separate module to ease the management and subsequent application of variants. Furthermore, the DiVA approach also considers the specific context to which each variation is applicable, as well as how each service variant affects the rest of the system and its properties. To support the methodology it uses Model-Driven Engineering (MDE) techniques to model the various elements of the variability. This is complimented by Aspect-Oriented Software Development (AOSD) to aid the encapsulation of the variants identified.

Chapter 10. SOA4ALL

The main purpose of the idea proposed in Chapter **??** is to create complex and distributed systems by seamlessly combining Web services. Using semantic-based technologies, an adaptive process, able to provide adaptive late-binding capabilities within workflows has been defined. The approach is based on the use of Semantic Web Services, that is of semantic annotations of services that support the application of automated machinery in order to reason about the functional and non-functional characteristics of services. In particular, it advocates that workflow definitions use service template as internal activities instead of concrete and prefixed services whenever flexibility in service selection is desired. At run-time, these services templates can be bound to specific services selected on the basis of the existing conditions and informed by contextual knowledge which may include monitoring data, user location or other aspects that may affect which service is the most appropriate. Since that service templates are described semantically, both the required functional and non-functional properties have clear semantics. This enhances the interpretation of services by humans, and more importantly, it allows service selection and data mismatches to be resolved at runtime as supported by Semantic Web Services middleware. Non functional information about cost, QoS, trust, legal constraints, etc.. can be taken into account so that the selected service is the most suitable from a business perspective. The overall approach relies on the provisioning of semantic annotation for services and the corresponding storage and querying system, on the replacement of workflow activities by service templates, and on the adaptation of execution environments in order to take service templates into account and trigger the selection of appropriate services automatically.

Chapter 11. VreSCO

VreSCO [**?**], the Vienna Runtime Environment for Service-oriented Computing has in part been developed within the FP7 Network of Excellence project S-Cube [**?**]. It provides a solution for some issues and shortcomings that are prevalent in current SOA research and practice. First and foremost it constitutes a service registry that is used by providers to store information about services in a (meta-)data model. The distinction between abstract features (metadata model) and concrete service implementations (service model) allows to group service instances that provide an identical functionality. Moreover, it enables clients to query the stored information using the specialized query language VQL in order to dynamically select an endpoint for their invocations. The selection may be based on functional criteria which concern the interface (or service contract), but also on non-functional criteria in the form of QoS attributes. The Daios framework employs an abstracted message format and is used in VreSCO to realize dynamic and protocol-independent invocations. The VreSCO data model supports explicit visioning of service revisions, operations and parameters. User-defined and default tags describe the features of a service revision and its position in the version graph. With the aid of mapping functions, the VreSCO runtime mediates between service instances which perform the same task but differ in their technical interface.

Acknowledgements The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

Acronyms

Use the template *acronym.tex* together with the Springer document class SVMono (monograph-type books) or SVMult (edited books) to style your list(s) of abbreviations or symbols in the Springer layout.

IRF	Integrated Research Framework
CPO	Cell Phone Operator
VO	Virtual Operator
SDD	Strategic Dependency Diagram
CD	Context Diagram
SBA	Service Based Application

Index

acronyms, list of, 23

Business Goal, 3

Context Diagram, 8
CPO, 2

Domain Assumption, 3, 6
Domain Model, 9

IRF, Integrated Research Framework, 2, 13

S-Cube, 2
Scenario, 3, 10
Strategic Dependency Diagram, 8
symbols, list of, 23

VO, 2

References

1. S-Cube Consortium, "Deliverable CD-IA-2.2.2 - Collection of Industrial Best Practices, Scenario and Business Cases," May 2009.
2. ——, "Deliverable CD-IA-3.1.1 - Integration Framework Baseline," March 2009.
3. "NESSI Open Framework - Reference Architecture (NEXOF-RA): Scenarios and Requirements for Open Framework Construction," http://www.nexof-ra.eu/?q=rep/term/200.
4. M. Jackson, *Software requirements & specifications: a lexicon of practice, principles and prejudices*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1995.
5. M. S. Abdullah, I. D. Benest, R. F. Paige, and C. Kimble, "Using unified modeling language for conceptual modelling of knowledge-based systems," in *Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November 5-9, 2007, Proceedings*, 2007, pp. 438–453.
6. "i-star software formalism," http://www.ics.uci.edu/~alspaugh/software/istar.html.