Introduction to the Sensoria Case Studies

Jannis Elgner[1], Stefania Gnesi[2], Nora Koch[3,4], and Philip Mayer[3]

[1] S & N AG
[2] Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", ISTI-CNR, Pisa, Italy
[3] Ludwig-Maximilians-Universität München, Germany
[4] Cirquent GmbH, Germany

Abstract. The foundational research carried out in Sensoria has been steered by a number of case studies for ensuring applicability of Sensoria methods and meeting expectations of society and the economy. In this chapter, we introduce these case studies. Three of the case studies came from industrial applications in automotive, finance and telecommunication domains; one came from an academic application for distributed e-learning and course management. Having in mind the relevance that these areas have in society and the economy, the above case studies have been extensively used in Sensoria during the whole project.

1 Introduction

One of the challenging characteristics of Sensoria has been the use of realistic case studies in order to assess the applicability of the insights, methodologies and tools developed within the project to service applications on top of Internet-based Global Computing environments. Sensoria has used four case studies: three from the industrial domains of automotive, finance and telecommunication; the fourth from the academic domain of distributed e-learning and course management. These case studies have been extensively used during the whole project for developing intuitions that could feed and challenge the research process according to the expectations of society and its economy, discussing and communicating ideas among partners, communicating research results to and getting feedback from the research community at large, and for dissemination and training activities.

The development activities related to the case studies have been carried out in parallel with the foundational activities of technical Sensoria work packages. In this way, the case studies have driven the results produced in those work packages, giving feedback that has been considered in the refinement of the methodologies and tools developed in the project.

In the following sections, we introduce the four case studies, presenting their main characteristics that have been investigated during the project together with some scenario descriptions for each of them. More detailed descriptions of some of scenarios will be presented later (see chapter 7-1).

2 Finance Case Study

The finance domain plays an important role in the worldwide economy and wrong decisions have a great influence on nearly all aspects of our society, even when they seem not to be directly connected. So, any wrong decisions and simple individual mistakes can be a disaster for the economy, as the world wide financial crisis has shown in the recent past. A bank has to handle critical information of customers and has a deep impact of their progression, e.g. when they need to borrow money from the bank. The finance case study is set in this context. In particular, we investigated the process of credit approval [1] as an example of a typical finance application.

Credit Request. This scenario, also known as *Credit Portal*, models a loan workflow including interaction with customers and employees. Normally, several employees with different authorisations facilities are involved in the communication process with the customer. The credit approval process is error prone and time consuming due its complexity in information collecting, customer and employee interaction, and risk management. Thus, the aim of IT in this context is to create a software system for simplifying and automating the credit request process in order to minimise mistakes and speed up the approval process.

In our scenario, the credit request workflow is implemented using a SOA-based system: Several specialised services are orchestrated to realise the process. Sensoria tools and methods have been used to verify properties of this system.

A web-based system is employed to handle the tasks involved for approving a credit request. This system is available to and used by both the customers of the bank as well as employees, the latter of which additionally may have different access rights.

The overall workflow in the credit request scenario works as follows:

- The customer logs in to the *Credit Portal* website. In order to start the process, the customer enters the necessary data like the amount of money and intended use. Furthermore, the customer has to enter his securities and balances data.
- This information is used for the calculation of a rating which is automatically started after the information is provided. The rating is based on the uploaded information and may also use the information stored from older credit requests. Depending on the rating, the approval can be given automatically or must be delegated to a clerk or supervisor.
  - In case an automated decision is possible, an answer is given immediately to the user without involving any bank employees.
  - In case of the non-automatic decision either a clerk or supervisor gets a notification and has to approve or decline the request. The choice of the employee is made on the risk calculated during the rating phase.
- In case of an approval, an offer is created and the customer needs to decide whether to accept or to decline the offer.

Automating the credit approval process in this way has huge benefits for the bank as the process is mostly automated any only requires human interaction if absolutely necessary. Of course, the process needs to be implemented correctly; this will be the topic of some of the following chapters.

3 Automotive Case Study

Much of the research and development costs in the vehicle production is due to the complexity of automotive software, which leads to an increased importance of software engineering in the automotive domain. By nature, automotive software is service-oriented as a modern vehicle consists of many individual parts which need to be orchestrated to perform as a whole. Therefore, there is great potential for service-oriented computing in the automotive world.

A vehicle that leaves the assembly line today is equipped with a multitude of sensors and actuators that provide the driver with services that assist driving the vehicle more safely, for example ABS systems or vehicle stabilisation systems. Driver assistance systems kick in automatically when the vehicle context renders it necessary, and more and more context is taken into account (road condition, vehicle condition, driver condition, weather conditions, traffic conditions etc.). Due to the advances in mobile technology it is possible to take connectivity to the car: telephone and internet access in vehicles are possible today, giving rise to a variety of new services for the automotive domain. The automotive case study is set in this context.

In Sensoria, we assume a SOA-based software infrastructure such as the one shown in Fig. 1 to be present in vehicles. In the figure, architectural modules of the vehicle are represented as nodes in an UML deployment diagram. This type of diagram has been selected to show the distribution of the components within the different platforms and devices of the vehicle, and those belonging to the vehicle environment.

The vehicle contains sensors and actuators and is able to determine its geographical position. Sensors are used to observe e.g. the vehicle's status. Actuators trigger fully-automatically the on-vehicle (low-level) driving assistance systems like ABS, anti-slipping and stability assistance. Another element of the architecture is the orchestrator of services that is in charge of achieving a goal by means of a composition of services. The discovery selects services based on established criteria. Services can be provided by local or remote service providers; they may be discovered locally or by external discovery services. The driver interface enables communication between driver and vehicle. The driver receives information from the active services and can enter commands to trigger, stop or customise them.

Several scenarios can be imagined in the context of the architecture defined above. We discuss the ones most used and referred to in the project below.

On Road Assistance. It is also called *On Road Repair* and *Low Oil Level.* In this scenario [3,5] the in-vehicle diagnostic system of the vehicle is triggered

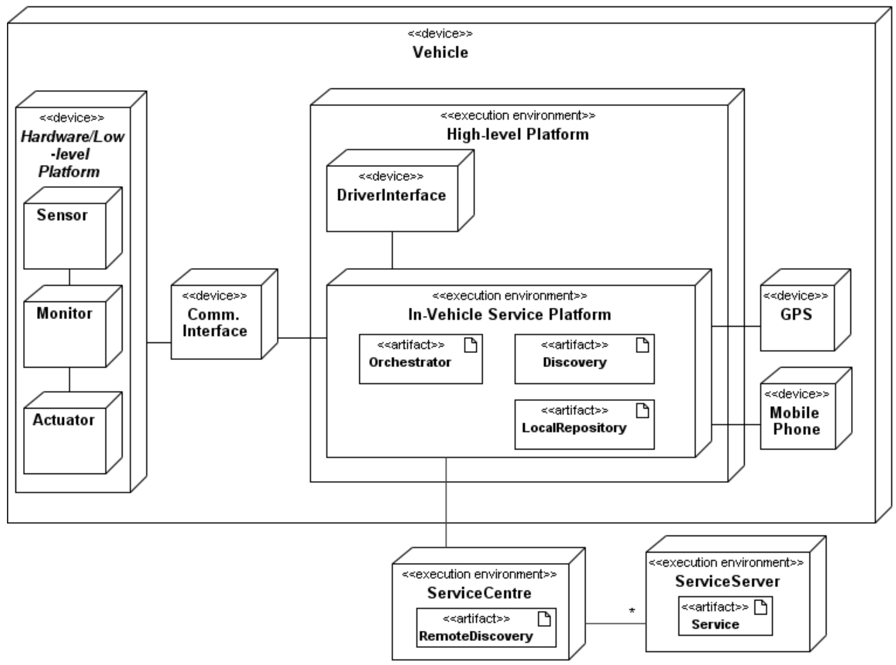

Fig. 1. Architecture of the vehicle in the automotive case study

by a failure in the car engine, for example, a low oil level and performs an analysis of the sensor values. The diagnostic system reports e.g. a problem with the pressure in one cylinder head, indicating that the driver will not be able to reach the planned destination.

The diagnostic system sends a message to the assistance system, which starts to orchestrate a set of services. Based on availability and the driver's preferences, the service discovery system identifies and selects the appropriate services in the area: repair shops (garage), towing truck and rental car stations. The selection of services takes into account personalized policies and preferences of the driver to find the *best services*. We assume that the owner of the car has to deposit a security payment before being able to order services.

Accident Assistance. This scenario (also know as *Airbag*) is concerned with road traffic accidents and dispatch of medical assistance to crash victims. Drivers interested in such a service must have an in-car GPS location tracking device with communication capabilities and have pre-registered their mobile phone information with the service. If a road traffic accident occurs, the deployment of the car airbag causes the on-board safety system to report the current location (obtained by GPS) to a pre-established accident report endpoint, which in turn attempts to call the registered drivers' mobile phone. If there is no answer to the call then medical assistance is dispatched to the reported location of the car (presuming that the driver has been incapacitated by injuries sustained in the accident).

Route Planning. It is also called *Emergency* [3]. In this scenario a vehicles' navigation system reacts to external events such as a broken car which is blocking the road. For example, two vehicles operating in a convoy mode may be forced to continue in an autonomous way due to the road block; this requires a reconfiguration of goals in both vehicles.

Road Sights. In this scenario, the driver has subscribed to a dynamic landmark service offered by the car company. The vehicles' GPS coordinates are automatically sent to the dynamic sights server at regular intervals, so the vehicles' location is known within a specified radius. Based on the drivers preferences, the dynamic sights server searches a landmark database for appropriate places of interest and displays them on the map of the vehicles' navigation system. The driver clicks on sights he would like to visit which results in more detailed information being displayed about a specific sight (e.g. opening times, guidance to parking etc.).

4 Telecommunication Case Study

The telecommunication case study has focused on the development of applications combining two global computing infrastructures, namely the Internet and next generation telecommunication networks. In this context, telecommunication services and capabilities including call and session control, messaging features, and presence and location features are integrated with the SOA-based computing infrastructure available in computing environments. In particular, issues concerning semantic and dynamic composition as well as orchestration to define, create and execute telecommunication services are addressed. Furthermore, secure and controlled interaction between application components deployed in different domains (e.g., an enterprise domain and a network operator domain) were considered in the case study.

The telecommunication case study mainly addressed issues concerning the evolution of the service infrastructure, not specific telecommunication services. However, some specific services were used in the description to exemplify the infrastructural issues. This case study was carried out during the first two years of the project and was hence used as test bed for the preliminary Sensoria research results (see [4]).

Telecommunication services, i.e., the services that are provided by a telecommunication infrastructure managed by a public network operator, are evolving by considering several aspects of *convergence*: convergence of media, convergence of terminals, combination of service features, and convergence of telecommunications and Internet worlds. Moreover, the services should be *user-centric*, that is, their behaviour should be personalised according to the requirements of single end-users. The possibility of personalisation should be uniform and cover all the features of the service. In particular, the end-users should be seen as single entities even across several different networks, terminals, communication media, and applications.

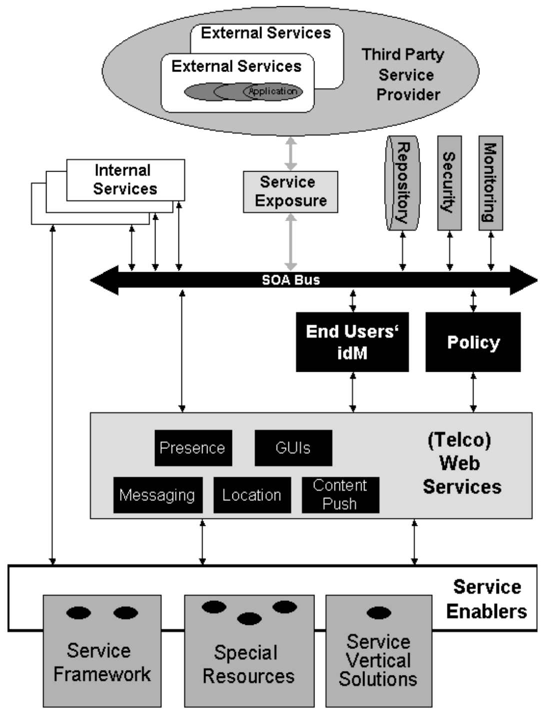

Fig. 2. Horizontal Telco architecture

Most of the current services in the telecommunication area are realised as a set of *vertical platforms*, each of them specialised to provide services involving a specific telecommunication feature and a specific network. Usually, such platforms integrate – in a single system – the service execution environments with the telecommunication features and some supporting functions (e.g., payment, authentication, profiles). Unfortunately, in general, such vertical systems are not connected at all.

This organisation of services introduces several problems in dealing with the realisation of *converged services*. In order to improve this situation, telecommunication services are evolving towards a *horizontal approach* based on the integration of systems for service delivery (see Fig. 2).

In Sensoria, we have investigated the possibility of evolving telecommunication services according to a horizontal approach with the help of service-oriented architectures. Some of the areas where the SOA approach can be adopted for the evolution of such services are:

- adoption of SOA-based principles and techniques to organise the internal structure of the service layer, and to define the communication bus among the different services and macro-functions;
- evolution of composition mechanisms to create and execute telecommunication services by adopting solutions based on orchestration or choreography and by introducing the possibility to handle semantic and dynamic service compositions;
- adoption of SOA technology to expose telecommunication capabilities to third party applications on the Internet, and to assemble services provided by third parties;
- adoption of SOA technology to introduce a uniform interaction model of services delivered by the service layer and terminal applications.

An interesting setting for evaluating the benefits of SOAs in the telecommunication context is the *Call-And-Pay Taxi* scenario.

Call-and-Pay Taxi. This scenario concerns the retrieval and purchase of goods or services via a mobile terminal. The *Call-and-Pay Taxi service* provides a user with the possibility to call a taxi by sending an SMS to a specific SMS service number and to pay the taxi service by sending another SMS. The service automatically debits the charging amount for the taxi ride to the end-users credit card and transfers the money to the taxi company. From the point of view of the involved human actors (i.e., end-user, call center agent, taxi driver) the service behaves in the following way:

- The end-user sends an SMS to the SMS service number which is associated to the Call-and-Pay Taxi service; in this way the end-user asks to call the taxi company of the town where he is currently located.
- The end-user receives an incoming call on his mobile phone in order to be connected to the call center of the local taxi company.
- The end-user talks to the taxi company agent; the call center agent contacts a taxi driver and confirms to the user the selected taxi number and the expected waiting time.
- The end-user receives an SMS which includes the taxi number and a *call-code* to identify the ride. The taxi driver receives a similar SMS.
- After the taxi ride, the user sends another SMS in order to authorise the payment using the preferred means of payment (stored in the user profile).
- In case of a successful transaction, the taxi driver and the end-user receive a confirmation of the payment with an SMS. In case of failure, an SMS with such an indication is sent to them (and the end-user has to pay in the traditional way).

5 The eUniversity Case Study

The administration of a university is a complicated task. Student applications, enrolment, course management, theses, and examination management all pose individual problems and, in general, a lot of paperwork. Nowadays, many of these tasks can and are being automated using computer systems. As universities are often large organisations with autonomous sub-organisations, a promising approach for this is the use of SOA-based software, in which the individual parts of a university as well as (external) students can work together with respective back- and frontends of a web-based system.

To investigate the problem of developing SOA-based university management systems, we have created a case study based on a set of university scenarios that make use of the specific features of SOAs [2]. In particular, we consider eUniversities, i.e., universities in which at least all of the paperwork, if not the courses themselves, are handled online. Scenarios in such an environment include:

- *Management of curricula*, i.e., providing information on which courses are offered, which requirements and how many credit points each course has, where a course takes place during a certain semester, etc.
- *Management of students*, in particular, enrolment and progress during a course of studies.
- *Thesis management*, i.e. the process of offering, accepting, working on, and submitting bachelor, master, or diploma theses.
- *E-Learning*, i.e., a system which provide students with (additional) training material, and enables the integration (embedding) of courses into other universities to enable students to participate in courses remotely.

During the course of the project, we have investigated the following three scenarios in more detail.

Thesis Management. In this scenario, we have considered the management of a thesis – bachelor, master, or diploma – from the announcement of a thesis topic by a tutor to the final assessment and student notification, including regular updates of the student to the thesis as well as status messages to the tutor, but also considering deadlines imposed by the examination office. This scenario is typical for a service orchestration – a central coordinator service uses several other services in combination to achieve a certain goal, and during this task, needs to be aware of problems, undoing previous work.

Submitting Coursework. We consider services to provide e-learning courses which can be shared between universities and services which enable several universities to jointly provide e-learning courses. Sharing courses in this way enables students to pick from a greatly increased number of courses; however, it also means that the number of students taking part in a course might be rather large. The scenario is that the students inscribed in a class all need to submit their coursework via uploads by the deadline. The question to ask here is how the system scales with respect to increasing student numbers and increasing file sizes.

Student Enrolment. In this scenario, students may apply for a certain course of studies online, providing the necessary documents and certificates via a website.

The functionality for handling the enrolment is provided by two service orchestrations which interact with each other and other services to verify the application. The consulted services include the student office, an admission checking service, and a service for the upload of documents. Requirements for this scenario (and for the verification) include that the client and the services should communicate via a secure and reliable connection, that the services are proven to perform as expected up to a certain workload, that messages sent between the services are accountable for, and that no deadlocks occur in the communication.

6 Conclusion

This chapter has introduced the four case studies used in Sensoria for feeding and steering the foundational research carried out in the project: Three case studies from the industrial domains of finance, automotive, and telecommunications, and one from the academic domain of university management. All case studies have been used extensively in Sensoria for validating the techniques, methods and languages developed in the project against the requirements discussed in this chapter.

The specification and implementation of demonstrators for the finance, automotive and eUniversity case studies is further discussed in Chapter 7-1. In Chapter 7-2, a detailed overview of the application of the results obtained in Sensoria (i.e., techniques, methods and languages developed in the technical work packages) to the case studies is be presented.

Acknowledgements. This work has been supported by the EU FET-GC2 IP project Sensoria (IST-2005-016004).

References

1. M. Alessandrini and D. Dost. Finance Case Study: Requirements, Specification and Modelling of Selected Scenarios (D8.3.a). Technical report, S&N AG, 2007.
2. M. Hölzl. Distributed E-University Management and E-Learning System: Requirements modelling and analysis of selected scenarios (D8.4.a). Technical report, LMU München, 2007.
3. N. Koch and D. Berndl. Requirements Modelling and Analysis of Selected Scenarios: Automotive Case Study (D8.2.a). Technical report, FAST GmbH, 2007.
4. C. Moiso, L. Ferrari, E. Thuegaz, N. Buscemi, U. Monanari, P. Bertoli, M. Pistore, R. Kazhamiakin, R. Bruni, S. Gnesi, M. ter Beek, M. Petrocchi, F. Mazzanti, J. Fiadeiro, and L. Bocchi. Telecommunication Case Study (D8.1.a). Technical report, TILab, 2007.
5. R. Xie and N. Koch. Automotive CASE Study: Demonstrator. Technical report, Cirquent GmbH, 2009.