SERVUS – Collaborative Tool Support for Agile Requirements Analysis

Thomas Usländer[1], Thomas Batz[1], Hylke van der Schaaf[1]

[1]Fraunhofer IOSB, Germany
thomas.uslaender@iosb.fraunhofer.de
thomas.batz@iosb.fraunhofer.de
hylke.vanderschaaf@iosb.fraunhofer.de

Abstract. Agility in software and service engineering usually needs to encompass the requirements analysis phase. This paper describes a multi-lingual Web-based tool that allows multiple users and stakeholders of different, possibly geographically dispersed organizations to perform requirements analysis activities in an agile and collaborative manner, and in close cooperation with software architects and engineers. The tool relies upon the identification and semi-formal description of use cases, e.g. following Cockburn's approach, and the resource-oriented extensions proposed by the SERVUS Design Methodology. It supports to define relationships between use cases, their representation in UML, their step-wise refinement and mapping to existing or emerging capabilities of service platforms as well as the specification of associated test cases. The tool is illustrated by its use in the European research project ENVIROFI. This project aims at analyzing and specifying the generic and specific enablement of the Future Internet core platform for the environmental information space.

Keywords: Requirements analysis, agility, use cases, service-oriented analysis and design, Future Internet, environment information space, SERVUS

1 Motivation

Agility in software and service engineering usually needs to encompass the requirements analysis phase. The reason is that, only in rare cases, the requirements are fully available and fixed when software design and developments starts. In contrary, especially in the domain of environmental information systems, requirements analysis is an agile process including a multi-step dialogue between the user(s), the stakeholders and the software architects who know about the technological capabilities and constraints, and who may also estimate the effort to realize the expectations of the user. The resulting discussion often leads to reconsiderations and/or refinements of the user requirements. If multiple users of one or even multiple organizations are involved, such dialogues are usually carried out during requirements analysis workshops facilitated by experienced systems analysts or architects. The crucial aspects are a solid methodology underpinning this process as well as an associated flexible documentation of the requirements during this process, also taking into account capabilities and constraints of underlying geospatial architectures [9]. This paper presents a Web-based collaborative tool that supports the documentation according to the SERVUS design methodology [1].

The paper is structured as follows: Section 2 provides an overview about the SERVUS Design Methodology, before the structure of the use case model, i.e. its meta-model, is presented in section 3. The architecture of the tool supporting the methodology in form of a so-called use case server is described in section 4. As an illustrative example, section 5 explains how it is used in the ENVIROFI project, before the paper concludes with references to other tool applications and ideas about future developments.

2 SERVUS Design Methodology

The SERVUS design methodology [1] describes individual design activities interconnected by a common modelling environment that is common to the Enterprise, Information and Service Viewpoint of geospatial architectural frameworks [4]. The common modelling environment follows a resource-oriented approach according to the Representational State Transfer (REST) architectural style [5]. Hereby, a resource is considered to be an information object that is uniquely identified, may be represented in one or more representational forms (e.g. as a diagram, XML document or a map layer) and support resource methods that are taken from a limited set of operations whose semantics are well-known (uniform interface). A resource has own characteristics (attributes) and is linked to other resources forming a resource network. Furthermore, resource descriptions may refer to concepts of the domain model (design ontology) using the principle of semantic annotation, yielding so-called semantic resources

SERVUS considers requirement analysis to be part of the Enterprise Viewpoint. It results in use case models that describe the behavior of a system [6] whereby “a use case is a sequence of actions performed by the system to yield an observable result that is typically of value for one or more actors or other stakeholders of the system”.

When designing an (environmental information) system, a use case expresses the functional, informational and qualitative requirements of a user (i.e. an actor or a stakeholder) with respect to the system. Usually, use cases do not describe the user interactions themselves. The requirements for the user interface, i.e. a description of how the system functions are accessed by and presented to the user depending on his end-user device, may be captured in separate task models and then mapped to use cases [10]. It is essential for the use case description that the level of abstraction, the type of formalism as well as the language should be such that it is adequate to the domain of expertise of the users. In order to serve as a kind of contract with the user, a use case shall be both understandable to the user but also precise enough. Very often this means that use cases shall be specified in a non-technical way, normally achieved using plain text in natural language. However, in order to reduce the ambiguities and impreciseness of descriptions in natural language, structured textual descriptions are preferred. This means that use case descriptions are structured according to a given template, e.g. an application form comprising identifier and description fields or thematic domain references associated with code lists.

The SERVUS design methodology recommends a semi-formal description of use cases, e.g. according to the template proposed by Cockburn [8], but extends it in order to include references to requested resources, e.g. a time-series of water gauge values represented as a diagram. Optionally, it may be accompanied by a UML use case diagram. The templates of the ENVIROFI project are contained in [3].

3 Use Case Meta-Model

The use case meta-model describes all the element types in a use case model and its relationships to each other. In addition to use cases as main element types, the SERVUS use case meta-model comprises the following element types:

- **actors**: describes the roles of users that initiate use cases.
- **test cases**: describes a possible instantiation of a use case that is decisive for the system test with respect to this use case.
- **requirements**: describe functions of the system under design that may support the execution of a use case. In the ENVIROFI project also called enablers (see below).
- **information resources**: describes the information elements including its basic operations (create, read, update, delete) that are required to carry out the use case (see the resource-oriented approach of SERVUS introduced in section 2). Note that the early identification and linkage to information resource is the key idea that aims at reducing the gap between thematic experts as users and software architects. Hence, it distinguishes SERVUS from other use case-based approaches.

Each element type has its own structure and template, i.e. its own set of text elements. There are the following relations between these element types: **use cases** (UC) are linked to **actors**, to other use cases, to **test cases (TC)**, to **requirements (REQ)** and **information resources (IR)**. These relations are illustrated in **Fig. 1** and will be explained below.

UC to Actor

- *performs* (inverse relation: *is performed by*): a UC is initiated and performed by an actor.

UC to UC

- *includes* (inverse relation: *is included in*): one UC is included in another UC, i.e. one UC is included as a whole in the main success scenario, extension or alternate path of another UC.

- *refines* (inverse relation: *abstracted from*): one UC is a refinement of another UC, e.g. it provides more details in its main success scenario, adds an extension or interprets a more abstract UC in the context of a thematic domain.

  Note: The relation type <<*refine*>> corresponds to the relation type <<*extend*>> that is typically used in conventional UML use case diagrams.

UC to TC

- *is tested by* (inverse relation: *tests*): one UC may be tested by one or more test cases.

  Note: As use cases may be included in other use cases, one test case may also be linked to several use cases (which are then included in each other).

UC to REQ

- *maps to* (inverse relation: *is derived from*): a UC is mapped to one or more requirements. This means that the system under design should provide function (may be in terms of a Web service) that fulfills each requirement..

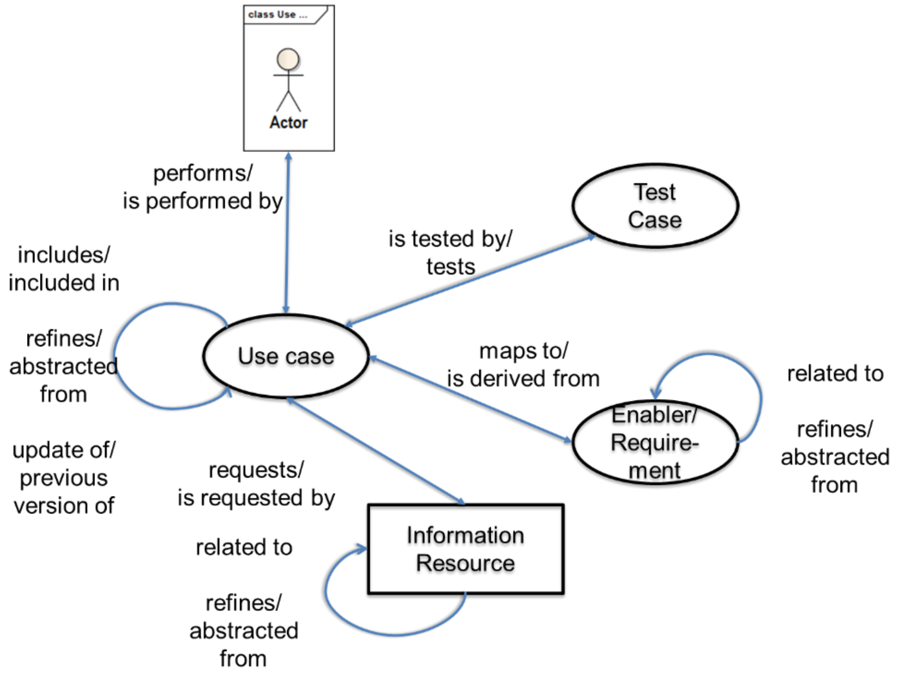

Fig. 1. Relationships between Use Cases, Requirements, Actors and Information Resources

REQ to REQ

- *related to* (bijective relation): one REQ is related to another REQ, i.e. there is some relationship between the requirements. This relation has to be better qualified in the future. It could be a unilateral or bilateral dependency but also some similarity in terms of concepts, design pattern or technology.

UC to IR

- *requests* (inverse relation: *is requested by):* a UC requests an information resource in a defined access mode (create, read, update, delete).

IR to IR

- *refines* (inverse relation: *abstracted from*): an information resource is a refinement of another information resource (in the sense of inheriting all properties of the more abstract information resource).
- *related to* (bijective relation*):* an information resource is related to another information resource. The meaning of the relation may be defined during the information modelling design step

4 Tool Support – The SERVUS Use Case Server

Non-trivial projects and related system analysis and design activities require a tool that supports the edition and documentation of the use cases. For the SERVUS design methodology a dedicated tool has been implemented that allows the users to work in a collaborative and distributed manner. Furthermore, it supports an agile approach, i.e. use cases and the other elements in the model may be iteratively specified which allows the users to refine and change them according to the knowledge that is typically gained in the analysis and design process.

The architecture of the SERVUS Use Case Server is illustrated in **Fig. 2**. There are the following basic architectural decisions and characteristics of the SERVUS Use Case Server:

- The server is realized as an application of the Fraunhofer IOSB WebGenesis® content and community management framework that supports the implementation of Web-based collaborative applications by rich generic functions.
- Use cases and all other elements following the meta-model (e.g. test cases, information resources) are persistently stored in a relational database as WebGenesis® entries. This enables the immediate use of built-in search functions for use cases and the other elements.
- The database structure is organized according to the use case meta-model which acts in this case as an ontology. Due to the built-in ontology support of WebGenesis® this design decision increases the flexibility of the system and facilitates the linkage of these core elements to ontologies representing thematic domains.

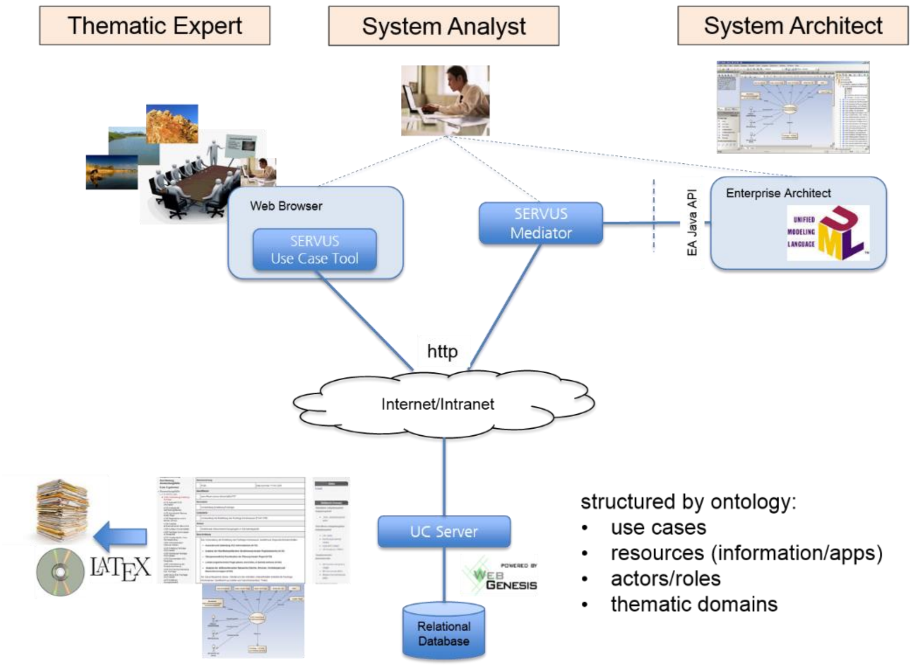

Fig. 2. System Architecture of the SERVUS Use Case Server

- At any time, there is the possibility to generate document reports out of the content of the use case server. In most projects, there is a need to deliver written use case specifications as soon as a given milestone has been reached. By using LATEX as document generation tool, the structure and the layout of the report may be easily changed and adapted to the requirements of a given project (e.g., addition of project or company logos or compliance with corporate designs).
- The face to the thematic expert as one of the user roles is the Web browser, hence, no installation on the user site is necessary. Furthermore, due to WebGenesis® as underlying framework, the layout of the Web-based application may be tailored to the corporate or project design constraints. Furthermore, user management and collaboration utilities between users (e.g. calendar functions, etc) are already available, and may be easily integrated in order to support the analysis process within and across organizational boundaries.
- On the other side of the user role spectrum there is the system architect whose objective is to formalize the requirements as UML use case diagrams (e.g. using a UML tool such as the Enterprise Architect) and map the use case scenarios to call sequences of interfaces and services, possibly also specified in UML.
- The system analyst has to mediate between both communities: On the one hand, he/she shall facilitate the requirements analysis process within the community of the

thematic experts aiming at getting consolidated semi-structured use case models. On the other hand, he/she shall guarantee the consistency with the UML models required for the system design activities led by the system architect. For this purpose the SERVUS use case server provides a mediator tool that, among others, extracts UML use case diagrams from the Enterprise Architect and feeds them as associated figures into the use case models.

5 Example: Use in the ENVIROFI Future Internet Project

The European Commission (EC) launched the Future Internet Public-Private Partnership programme (FI-PPP) with the idea of enabling a broad range of Internet applications, including those of the environmental knowledge domain. The FI-PPP programme shall advance a shared vision for common pan-European technology platforms and their implementation, as well as the integration and harmonization of the relevant policy, legal, and regulatory frameworks.

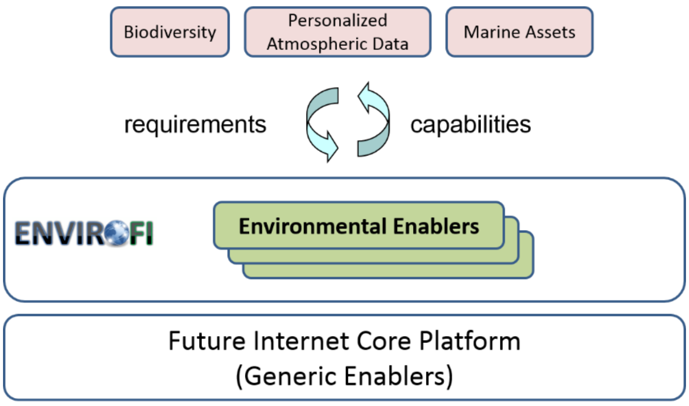

Fig. 3. ENVIROFI Approach to Design an Environmental Observation Web

As illustrated in **Fig. 3** the FI-WARE platform is an ICT platform based on generic functional building blocks: the so-called Generic Enablers (GE). These are used by the usage area projects in order to validate and demonstrate the new generation FI enabled environmental applications. ENVIROFI represents the environmental usage area within phase 1 of FI-PPP (see http://www.envirofi.eu/). It explores the so-called environmental enablers, i.e. reusable building blocks for collecting and processing environmental data, and provides environmental sector requirements to FI-WARE [7]. Thus, ENVIROFI lays the foundation for an FI-enabled Environmental Observation Web, which will help Europe to tackle the grand challenges of climate change, socio-economic pressures on the environment and sustainable development.

ENVIROFI's vision is to establish an Environmental Observation Web in which all environmental data from sensors, citizens and models become available through the

Internet in a standardized and usable format [13]. ENVIROFI specifically works on three environmental application areas: biodiversity, personalized atmospheric data and marine assets.

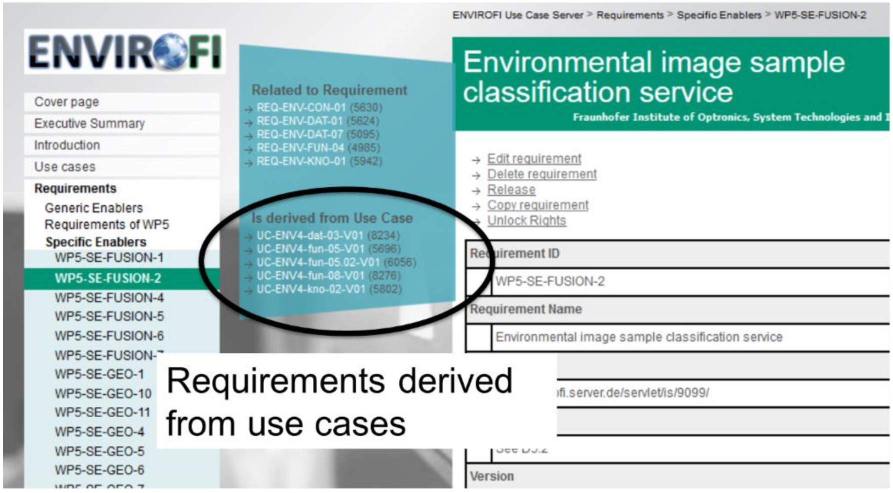

Fig. 4. Requirements linked to use cases

The setting of the FI-PPP encompassing both usage area projects collecting requirements and a core platform project providing a capability platform under development requires a sophisticated and transparent analysis and design process. The ENVIROFI project has selected the SERVUS methodology including its tool support. The SERVUS use case meta-model has been interpreted such that the entry category of “requirements” represents requirements for FI-PPP generic or specific enablers, hence, generic functions to be integrated as reusable components into the platform with a well-defined, if possible, standard interface [3].

Fig. 4 shows a screenshot of the use case server as it was applied and configured in the ENVIROFI project. It illustrates on the left side a list of requirements for specific enablers and in the middle the list of those use case entries from which a selected requirement was derived. These links are realized as bijective hyperlinks in the use case server, hence, the user can browse through the entries in both directions and therefore get full transparency.

6 Further Application Examples and Next Steps

In addition to European research projects such as ENVIROFI, the SERVUS methodology and its use case server is used in the following environmental projects in Germany:

- WIBAS 5.0 - Architectural Re-Engineering of the Integrated Environmental Information System WIBAS [11]. In this project, user requirements for a functional upgrade in the context of a new virtually centralized system architecture in the German Federal State of Baden-Württemberg have to be collected among several dozens of multi-disciplinary experts of regional environmental authorities.
- FLIWAS 3.0: Requirements analysis of the German-Dutch Flood Information and Warning System along the Rhine river basin. FLIWAS is a Web-based flood information and warning system for local and regional decision support [12]. Originally developed and installed as a central database application for the German Federal State of Baden-Württemberg it will now undergo an essential architectural and functional upgrade along the Rhine river basin encompassing other German federal states and cities (e.g. the city of Cologne) as well as water authorities in the Netherlands. For this new analysis process, the SERVUS methodology is used. Hence, one of the utmost new functions in the SERVUS use case server will be a multi-lingual support of the use case edition and documentation process.

All these example projects demonstrate that an agile requirements analysis methodology urgently needs a powerful software tool to support its edition and documentation. However, beyond technical tool support, there is also a confirmation that requirements analysis is a social process and needs didactical and facilitation skills, especially for the system analyst as the "man or the woman in the middle". Further developments will focus on investigations how to enrich the SERVUS use case server to better support such group facilitation techniques.

7 References

1. Usländer, T.: Service-oriented Design of Environmental Information Systems. PhD thesis of the Karlsruhe Institute of Technology (KIT), KIT Scientific Publishing. ISBN 978-3-86644-499-7 (2010) http://digbib.ubka.uni-karlsruhe.de/volltexte/1000016721
2. Usländer, T., Batz, T.: How to Analyse User Requirements for Service-Oriented Environmental Information Systems. In: Proceedings of the ISESS 2011, Brno, Czech Republic, pp. 161-168, Springer Verlag (2011)
3. ENVIROFI Consortium (Ed.). D4.1.2 Environmental Requirements II. Deliverable D4.1.2 of the FP7 project ENVIROFI Work Package 4 (2012) http://www.envirofi.eu/Downloads/PublicDeliverables/tabid/4983/Default.aspx
4. Usländer, T. (ed.). "Reference Model for the ORCHESTRA Architecture Ver-sion 2.1". OGC Best Practices Document 07-097, 2007. http://portal.opengeospatial.org/files/?artifact_id=23286
5. Fielding, R.T. "Architectural Styles and the Design of Network-Based Software Architectures". Doctoral dissertation, University of California, Irvine, 2000. http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
6. Jacobson, I. and Ng, P.-W. "Aspect-Oriented Software Development with Use Cases". The Addison-Wesley Object Technology Series, ISBN 0-321-26888-1, 2005.
7. Berre, A.-J., Usländer, T., Schade, S.: Identification and Specification of Generic and Specific Enablers of the Future Internet – Illustrated by the Geospatial and Environmental Domain. Lecture Notes in Computer Science, Volume 6994, pp. 278-289 (2011)

8. Cockburn, A. Writing Effective Use Cases. ISBN-13: 9780201702255. Addison-Wesley (2001)
9. Usländer, T., Denzer, R. “Requirements and Open Architecture for Environmental Risk Management Information Systems”, Chapter 15 of Van de Walle, B., Turoff, M. and Hiltz, S.R. (eds.): Information Systems for Emergency Management. In the Advances in Management Information Systems monograph series (Editor-in-Chief: Vladimir Zwass). Armonk, NY: M.E. Sharpe Inc., ISBN 978-0-7656-2134-4, 2009.
10. Sinnig, D., Chalin, P. and Khendek, F. “Use case and Task Models: An Integrated Development Methodology and Its Formal Foundation”. ACM Transactions on Software Engineering and Methodology, Vol. 22, No. 3, Article 27, 2013.
11. Batz, T., Rudolf, M., Usländer, T., Braun von Stumm, G., Schulz, K.-P., Uhrig, W., Scherrieble, T., Schillinger, W., Spandl, H., Frenzl, R. and Wiechmann, R. “WIBAS 5.0 Modellierung von Anwendungsfallen in WIBAS 5.0 unter Nutzung von SERVUS“. In: Mayer-Föll, R. (ed.): Umweltinformationssystem Baden-Württemberg : F+E-Vorhaben KEWA; kooperative Entwicklung wirtschaftlicher Anwendungen für Umwelt, Verkehr und benachbarte Bereiche in neuen Verwaltungsstrukturen; Phase VI, 2010/11, KIT Scientific Publishing, KIT Scientific Reports 7586, ISBN: 978-3-86644-674-8, pp.87-97, 2011.
12. Sartorius, M. “Bewältigung der Hochwasserlage im Mai/Juni 2013 mit FLIWAS“. In: Die Gemeinde (BWGZ). Organ des Gemeindetags Baden-Württemberg, Issue 18, 2013.
13. Usländer, T., Berre, A. J., Granell, C., Havlik, D., Lorenzo, J., Sabeur, Z. and Modafferi, S.. “The Future Internet Enablement of the Environment Information Space». In: J. Hřebíček et al. (Eds.): ISESS 2013, IFIP AICT 413, pp. 109–120, 2013.