SEVENTH FRAMEWORK PROGRAMME

ICT PPP

Future Internet

# The Environmental Observation Web and its Service Applications within the Future Internet

FP7-284898

Collaborative project

## Environmental Requirements I

**Fraunhofer IOSB**

Deliverable due date: 30/09/2011

Actual submission date: 30/09/2011

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 284898

## Document Control Page

| | |
|---|---|
| **Title** | Environmental Requirements I |
| **Creator** | Thomas Usländer, Fraunhofer IOSB |
| **Description** | This deliverable describes the methodology and the tool support applied in ENVIROFI to collect, analyze and consolidate environmental requirements, especially those from WP1 (water), WP2 (land) and WP3. Furthermore, it contains the result of this analysis process in terms of abstract (i.e. application-independent) use cases as a basis for the identification of requirements on FI-PPP generic and specific enablers. |
| **Publisher** | ENVIROFI Consortium |
| **Contributors** | Sven Schade, EC-JRC<br>Thomas Usländer, IOSB<br>Denis Havlik, AIT<br>Mihai Bartha, AIT<br>Arne J. Berre, SINTEF<br>Jose Lorenzo, ATOS |
| **Creation date** | 18/07/2011 |
| **Type** | Text |
| **Language** | en-GB |
| **Rights** | copyright "ENVIROFI Consortium" |
| **Audience** | ☒ internal<br>☐ public<br>☐ restricted |
| **Review status** | ☐ Draft<br>☐ WP leader accepted<br>☐ Technical Manager accepted<br>☒ Coordinator accepted |
| **Action requested** | ☐ to be revised by Partners<br>☐ for approval by the WP leader<br>☐ for approval by the Technical Committee<br>☐ for approval by the Project Coordinator |
| **Requested deadline** | |

## Revision history

| Version | Date | Modified by | Comments |
|---|---|---|---|
| 0.1 | 18/07/2011 | Thomas Usländer (Fraunhofer IOSB) | Initial table of contents |
| 0.2 | 24/08/2011 | Thomas Usländer (Fraunhofer IOSB) | Draft of Section 6 |
| 0.3 | 15/09/2011 | Arne J. Berre (SINTEF) | Input on agile methods and lifecycle and architectural analysis – section 5, 8 and 9 |
| 0.4 | 19/09/2011 | Sven Schade (EC-JRC) | Input to section 7 and placeholder for section 12: lessons learned |
| 0.5 | 25/09/2011 | Thomas Usländer (Fraunhofer IOSB) | Revised structure and editorial changes |
| 0.6 | 30/09/2011 | Thomas Usländer (Fraunhofer IOSB) | Edition of missing sections, finalization |
| 1.0 | 30/09/2011 | Jose Lorenzo (ATOS) | QA and Final Revision |

# Table of Contents

Glossary....6
Abbreviations and acronyms....6
1 Introduction....8
2 Agile Requirements Analysis....9
3 ENVIROFI Use Case Analysis Methodology....12
3.1 Relationship to the ISO Reference Model....12
3.2 Description of the Use Case Analysis Methodology....12
3.2.1 Use Case and Requirements Modelling....13
3.2.2 Use Case Analysis Process....14
3.2.3 Use Case Template....15
3.2.4 Requirements template....18
3.2.5 Relation types between use cases and requirements....22
3.2.6 UML Representation....23
3.3 Tool Support....25
4 Thematic Areas of the Use Case Analysis....27
5 Categorization of Enablers....29
5.1 Lifecycle-based Approach....29
5.1.1 Lifecycle Components....29
5.1.2 Value Chain of Environmental Knowledge Generation....30
5.1.3 Overview of Stakeholders....30
5.1.4 Requirements for a Next Generation of Environmental Services....31
5.2 Architecture-based Enabler Classification....32
5.2.1 Architectural Classification according to ISO 19119....32
5.2.2 Architectural Classification according to Cloud Computing Service Categories....33
6 Overview about Generic Use cases and Requirements....35
7 Lessons Learnt and Conclusions....36
8 References....37
9 Annexes....38
9.1 Annex A: List of Generic Use Cases....38
9.2 Annex B: List of Requirements for "Generic Enablers"....38

# Index of figures

Figure 2-1. Overall approach of Agile Software Requirements [05 – Leffingwell, D. (2011)] ............................9
Figure 2-2. ENVIROFI Interpretation of the Agile Methodology Approach for Use Cases ............................ 10
Figure 2-3. ENVIROFI Interpretation of the Agile Methodology Approach for Requirements......................... 11
Figure 3-1. Overall Idea of the ENVIROFI Use Case Analysis ......................................................................... 13
Figure 3-2. Procedure of the ENVIROFI Use Case Analysis ............................................................................ 15
Figure 3-3. Relationships between Use Cases, Requirements, Actors and Information Resources ............. 23
Figure 3-4. Schema for the Graphical Representation of a Use Case ............................................................ 24
Figure 3-5. Screenshot of the Use Case Server (source: Fraunhofer IOSB)................................................... 26
Figure 4-1. Overview of the ENVIROFI WP1 Use Cases on "Biodiversity" ..................................................... 27
Figure 4-2. Overview of the ENVIROFI WP2 Use Cases on "Air Quality" ....................................................... 28
Figure 4-3. Overview of the ENVIROFI WP3 Use Cases on "the Marine Environmental Area" ..................... 28
Figure 5-1. Core Components of the SDI Reference Model (derived from [07 – CEN (2011)]). .................... 29
Figure 5-2. Added value chain of environmental knowledge generation [07 – CEN (2011)] .......................... 30
Figure 5-3. Relationships of enablers in both a layered and a bus architecture ............................................ 32
Figure 5-4. Generic and specific enabler types for SaaS, PaaS and IaaS ...................................................... 34

# Index of Tables

Table 1. Abbreviations and Acronyms...........................................................................................................6
Table 3-1: Description of the Use Case Template ......................................................................................... 18
Table 3-2: Description of the FI-PPP Requirements Template ...................................................................... 22
Table 5-1. Added-value chain of environmental knowledge generation [6]. ................................................... 31
Table 8-1. References ................................................................................................................................... 37

## Glossary

The glossary of terms used in this deliverable can be found in the public document "ENVIROFI_Glossary.pdf" available at http://www.envirofi.eu/.

## Abbreviations and acronyms

| Term | Explanation |
|---|---|
| CEC | Commission of the European Communities |
| CEN | European Committee for Standardisation |
| FI | Future Internet |
| FI-PPP | Future Internet Public Private Partnership |
| GEOSS | Global Earth Observation System of Systems |
| GMES | Global Monitoring for Environment and Security |
| I2ND | Interface to Network and Devices |
| IaaS | Infrastructure as a Service |
| INSPIRE | Infrastructure for Spatial Information in the European Community |
| IoT | Internet of Things |
| OGC | Open Geospatial Consortium |
| PaaS | Platform as a Service |
| PSI | Public Sector Information |
| SaaS | Software as a Service |
| SDI | Spatial Data Infrastructure |
| SERVUS | Design Methodology for Information Systems based upon Geospatial Service-oriented Architectures and the Modelling of Use Cases and Capabilities as Resources |
| SOA | Service-Oriented Architecture |
| SOAD | Service-Oriented Analysis and Design |
| TC | Technical Committee |

**Table 1.** Abbreviations and Acronyms

## Executive Summary


The present deliverable D4.1.1 is the result of the first iteration of ENVIROFI task 4.1 - Requirements Consolidation and Interaction with the core FI Core Platform Consortium. A second version will be edited in month 15 (June 2012) in form of the deliverable D4.1.2. This task shall investigate existing requirements identification methodologies and shall provide guidance and best practices in collecting environmental requirements from various sources. Furthermore, it shall properly categorize and document them for further processing, including presentation to the FI Core Platform. In order to support this activity, this task shall develop the necessary infrastructure for managing the requirements, with the aim of providing support for their proper analysis. The result shall also serve as input to WP5 for system requirements elicitation and to task 4.2 which aims at specifying the ENVIROFI architectural approach.

This document acknowledges the fact that a methodology for a coherent identification of requirements is needed in order to identify and document requirements coming from various sources, including both internal (water, land, and air pilot case studies in WP1-3) and external (various projects, initiatives, e.g. INSPIRE, GMES, GEO/GEOSS via WP6).

This deliverable provides an overall description of agile software requirements analysis being pursued in FI-WARE and its interpretation for ENVIROFI. Based upon this, the methodology for the ENVIROFI Use Case Analysis is described, including also a description of the use case and enabler requirements template used. It summarizes the use cases as having been described in the WP1-3, respectively for the three thematic areas biodiversity, air and marine environment. Furthermore it provides an overview about the generic use cases (WP4) and the requirements (WP4 and WP5) on generic and specific enablers having been identified so far.

As an input to the respective discussion in the FI-PPP community it contains a discussion about how to categorize requirements for enablers following different approaches of ISO, CEN work on spatial data infrastructures and structuring approaches in cloud computing.

The document finishes with some lessons learnt during this first iteration cycle and resulting conclusions. Note that the deliverable encompasses two annexes: Annex A contains the descriptions of the generic use cases, annex B the descriptions of the enabler requirements. Both annexes mirror the contents of the ENVIROFI use case server and have been automatically generated from its contents.

# 1 Introduction

The architecture of environmental software applications is a design artifact that results from a dialogue between experts in thematic domains such as hydrology or air quality and information technology experts [15 – Usländer, T. et al (2010)]. As in traditional software engineering it is a dialogue between those who express their requirements in terms of which information and functions they need, with which level of quality and dependability, and those who know about the capabilities and constraints of software systems and architectural styles. Today, it is the style of a service-oriented architecture that is more and more used for the design of environmental software applications. Hence, the architecture shall reuse as much as possible standard services and capabilities of the existing systems in order to get an open and cost-effective solution. The solution shall be flexible such that future needs may be satisfied, e.g. it shall be possible to easily integrate new monitoring systems, new environmental models or new sensors. The architectural approach shall even be applicable to other application domains, e.g. transfer of a water-related information system to air quality monitoring. The following two questions arise:

1. Is there a reference architecture, a kind of template for the design artifact, that may be used for this type of systems?
2. And if yes, what methodology is available that supports the design process for it, specifies requirements and explicitly considers architectural side conditions?

As an answer to the first question we refer to the architectural approach of the Sensor Web Enablement (SWE) that has been developed under the auspices of the Open Geospatial Consortium (OGC) and further elaborated, validated and implemented by research projects such as the European FP7 project SANY (Sensor Anywhere). This approach is one of the candidates for the FI-PPP community to be considered for the environmental usage areas.

The answer to the second question is in the focus of this deliverable. There are currently numerous methodologies for service-oriented analysis and design (SOAD), however, none of them has reached up to now the status of an agreed methodology in the SOA community [16 - Kohlborn et al (2009)]. One of the biggest challenges is the transition between the business and organizational aspects of a service-oriented approach and their realization on the technical level, e.g. as Web services, especially, when aiming at reusing as much as possible the capabilities of existing platforms as it is the case in the FI-PPP context where the aim is to re-use the (emerging) enablers of the FI-WARE platform.

Furthermore, the design of software applications in the environmental usage area poses specific needs that are not yet sufficiently addressed. This needs a SOAD methodology that is tailored to the geospatial domain, which means that it contains and exploits knowledge about information and service models, guidelines and constraints of geospatial architectures, e.g., interfaces of OGC service types. Hence, the ENVIROFI project relies upon the basic ideas of the SERVUS design methodology that specifically addresses the needs for a service-oriented design of environmental information systems [15 – Usländer, T. et al (2010)]. The acronym SERVUS denotes a Design Methodology for Information Systems based upon Geospatial Service-oriented Architectures and the Modeling of Use Cases and Capabilities as Resources. It is based upon an extended use case approach and describes individual design activities interconnecting the Enterprise, Information and Service Viewpoints of the ISO Reference Model for Open Distributed Processing (RM-ODP), see section 3.1.

This deliverable is structured as follows: Section 2 starts with an overall description of agile software requirements analysis being pursued in FI-WARE and its interpretation for ENVIROFI. Based upon this, the methodology for the ENVIROFI Use Case Analysis is described in section 3, including also a description of the use case and enabler requirements template used. Section 4 summarizes the use cases as having been described in the WP1-3, respectively for the three thematic areas biodiversity, air and marine environment. A discussion about categorization approaches for enablers is contained in section 5, followed by an overview about the generic use cases (WP4) and the requirements (WP4 and WP5) having been identified so far. The document finishes with some lessons learnt during this first iteration cycle and resulting conclusions in section 7. Note that the deliverable encompasses two annexes: Annex A contains the descriptions of the generic use cases, annex B the descriptions of the enabler requirements. Both annexes mirror the contents of the ENVIROFI use case server and have been automatically generated from its contents.

# 2 Agile Requirements Analysis

This section provides a short overview and a description of the state-of-the-art in agile requirements analysis. It motivates the application of an agile and iterative approach triggered by use cases which are originally derived from scenario descriptions. The requirements analysis is being harmonised with the agile approach chosen by FI-WARE. The challenge here is to relate a use case analysis methodology as described in section 6 with an agile development approach.

A good source for this may be found in the recent methodology work by the agile community and the recommendations of a recent book on agile software requirements [05 – Leffingwell, D. (2011)].

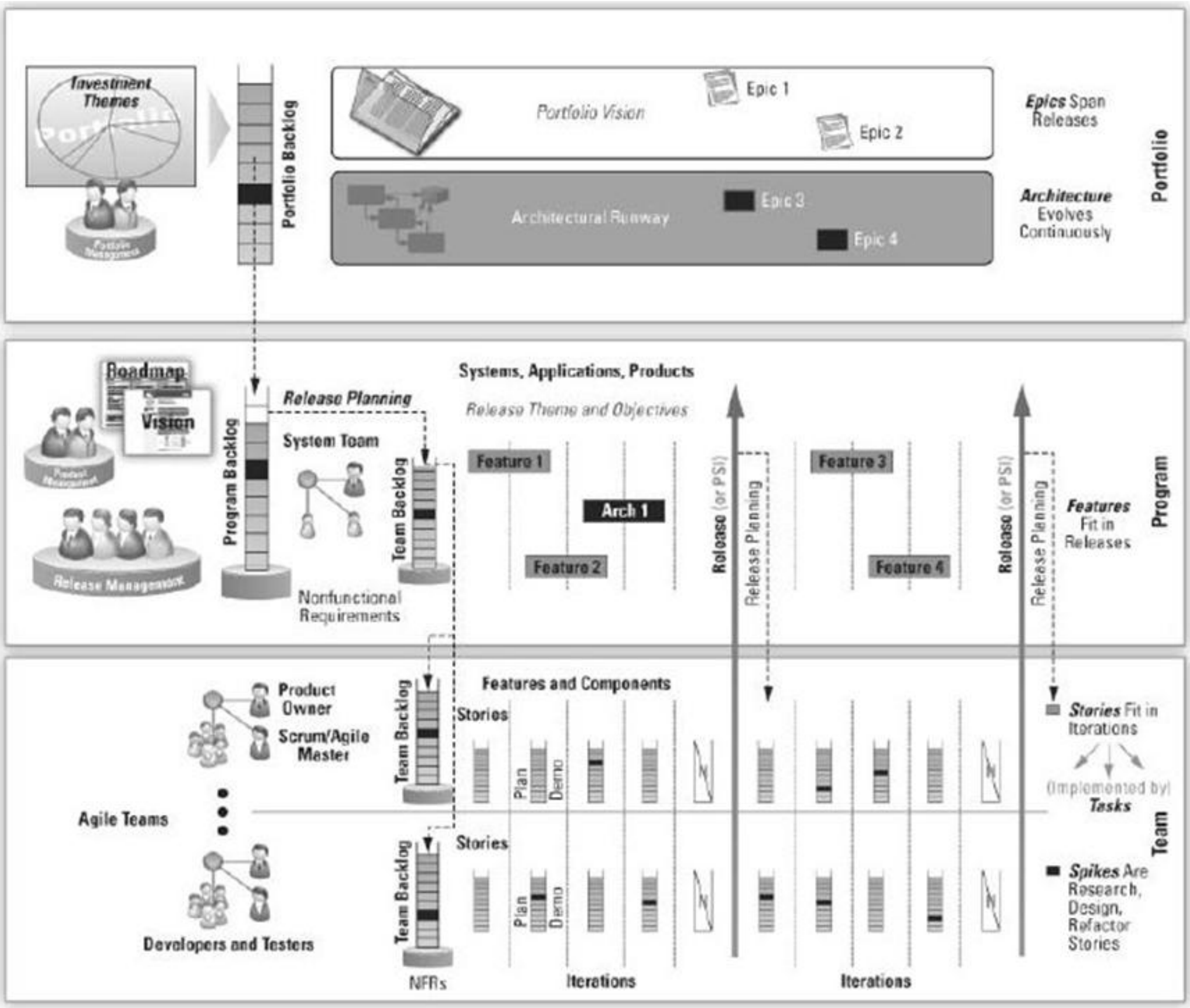


**Figure 2-1.** Overall approach of Agile Software Requirements [05 – Leffingwell, D. (2011)]

Figure 2-1 illustrates the suggested expansion in terms of three levels **team**, **program** and **portfolio** and related **backlog items** as results of the requirements analysis process. The team level refers ot the level of the classical agile teams and focuses on the provision of **user stories**. The next abstraction level is the program level and defines **features** which refer to requirements on systems, applications and products. Finally, the portfolio level is defined in terms of **investment themes** as well as **portfolio** and **architectural epics**.

 

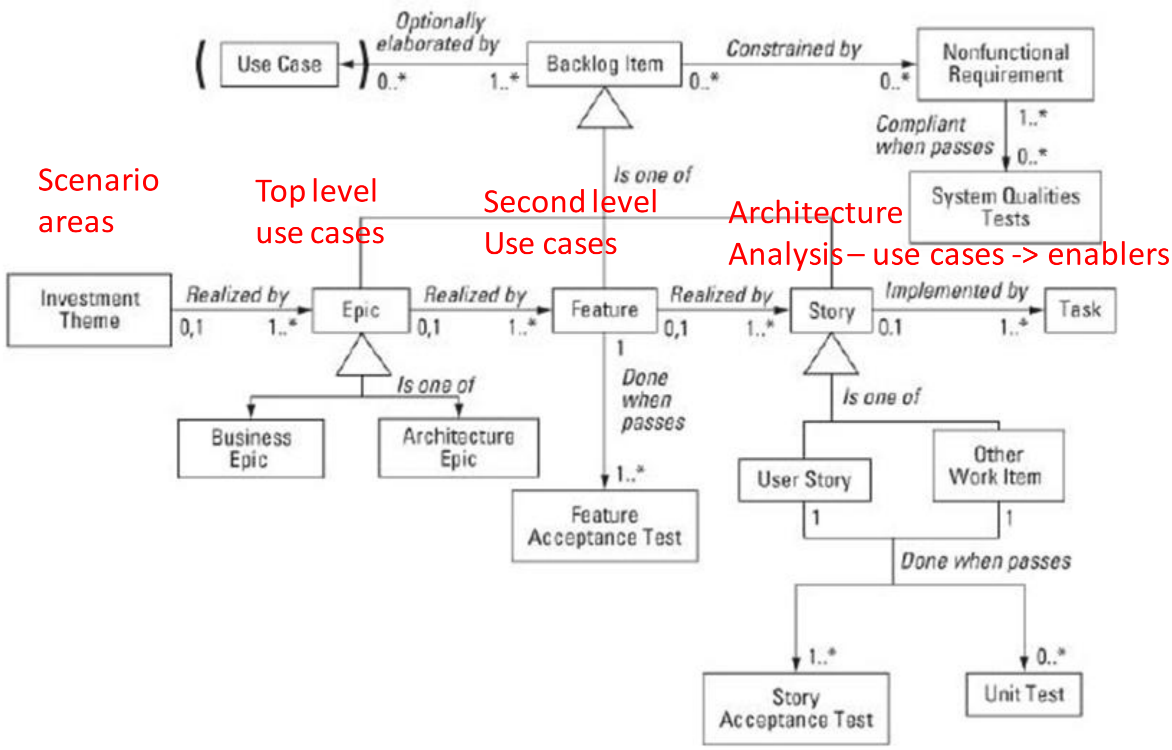


**Figure 2-2.** ENVIROFI Interpretation of the Agile Methodology Approach for Use Cases

Now, these ideas have to be interpreted and mapped to the FI-PPP requirements analysis process.

Figure 2-2. ENVIROFI Interpretation of the Agile Methodology Approach for Use Cases

shows a meta-model of different kinds of so-called backlog items. As can be seen by the inheritance relationships, backlog items may be either epics, features or stories. It also shows that backlog items may be elaborated by use cases, however, just as an option. In ENVIROFI, this option is leveraged as use cases are considered to be an important step in formalizing the user requirements.

In red it is shown how the corresponding use case types that are addressed within ENVIROFI can be related to backlog items in the agile methodology approach. ENVIROFI applies the following interpretation:

- Investment themes are interpreted as **scenario areas**
- Epics are interpreted as **top-level use cases**.
- Features are interpreted as **second-level use cases**.
- Stories are interpreted as architecture-level use cases, which correspond to **requirements** on what is being called **enablers** in FI-PPP.

Now, this scheme may be recursively applied to the requirements on enablers, too. The ENVIROFI interpretation is illustrated in Figure 2-3. Note that enablers may be generic (to be provided by FI-WARE) or specific (to be provided by the usage are projects). However, this distinction is not considered here.

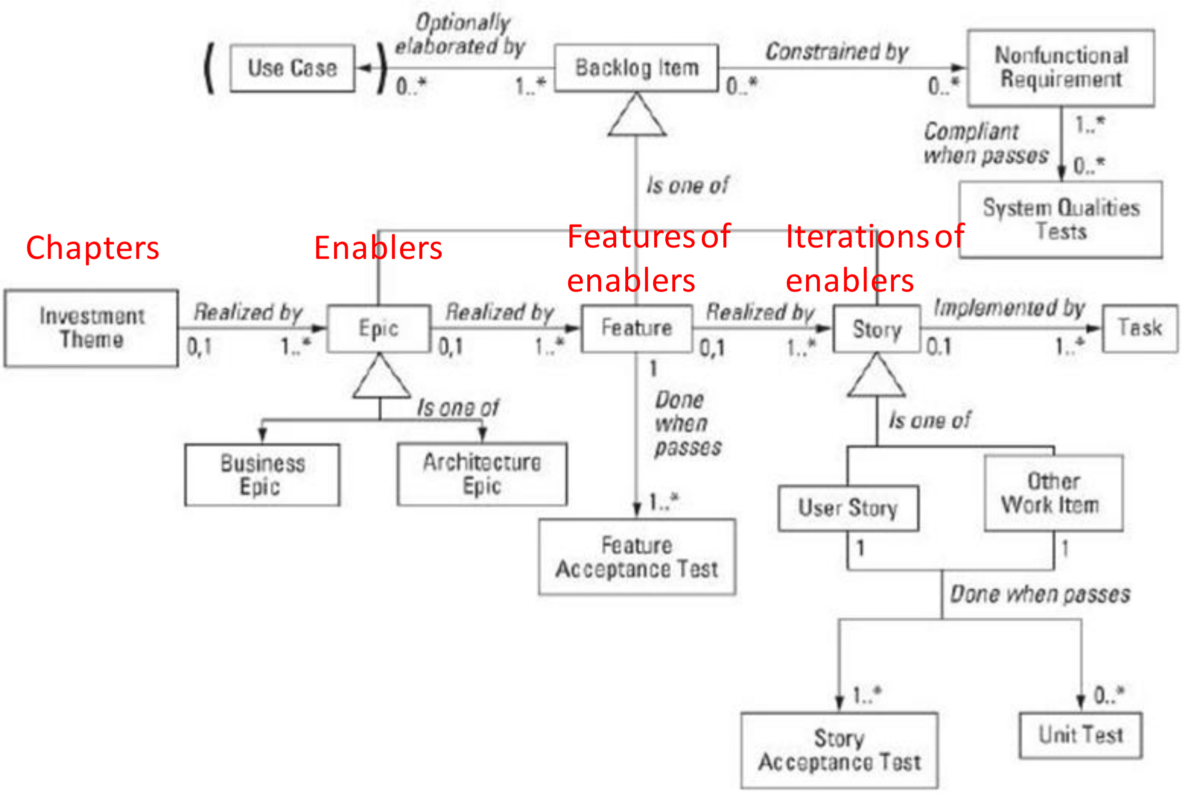


**Figure 2-3.** ENVIROFI Interpretation of the Agile Methodology Approach for Requirements

This figure shows again the meta-model for different kinds of backlog items and annotated in red are the corresponding concepts that are related to the chapters and enablers in the FI-WARE structure.

ENVIROFI applies the following interpretation:

- Investment themes are interpreted as **chapters of enablers.**
- Epics are interpreted as the **enablers** themselves..
- Features are interpreted as **features (characteristics) of enablers**.
- Stories are interpreted as iterations (i.e. different versions) of **enablers**.

# 3 ENVIROFI Use Case Analysis Methodology

## 3.1 Relationship to the ISO Reference Model

The analysis of user requirements and the derivation of requirements on enablers cannot take place without having in mind a common reference model of a Future Internet (FI) system architecture. Here, ENVIROFI proposes to rely upon agreed international standards such as ISO RM-ODP. Inspired by “distributed processing systems based on interacting objects”, ISO defined the Reference Model for Open Distributed Processing (ISO/IEC 10746-1:1998).The RM-ODP standards have been adopted widely. They constitute the conceptual basis for the ISO 191xx series of geospatial standards from ISO/TC211.

The viewpoints of RM-ODP are applied as follows:

- The **Enterprise Viewpoint** describes the purpose, scope and policies of that system and contains the use cases described above.
- The **Information Viewpoint** describes the semantics of information and information processing and contains the information resources identified as the use case extension.
- The **Computational Viewpoint** describes the functional decomposition of the system into components and objects which interact at interfaces. In ENVIROFI, this viewpoint is also referred to as **Service Viewpoint** acknowledging its application in (geospatial) service-oriented architectures as predominant architectural style [01 – Usländer, T. (2010)].
- The **Engineering viewpoint** describes the mechanisms and functions required to support distributed interaction between objects in the system.
- The **Technology viewpoint** describes the choice of technology in that system.

The use case analysis methodology described in the following sections comprises the activities of the ISO RM-ODP Enterprise Viewpoint. Hence, the specification of use cases and the requirements are artefacts of this viewpoint and constitute the ENVIROFI Enterprise Viewpoint specification. Following the RM-ODP model this specification is the foundation for the abstract system design resulting in the information and service viewpoint, and, further on, for the concrete system design in the technology and engineering viewpoint.

## 3.2 Description of the Use Case Analysis Methodology

This section provides an overview about the ENVIROFI Use Case Analysis Methodology. This methodology is positioned as a prelude of the SERVUS methodology [01 – Usländer, T. (2010)] that aims at a Design Methodology for Information Systems based upon Geospatial Service-oriented Architectures and the Modelling of Use Cases and Capabilities as Resources.

The purpose of the SERVUS methodology as applied in ENVIROFI is to capture and analyze the requirements of the three usage examples (application pilots) of ENVIROFI. These requirements are elaborated in a first step as use cases (UC) b the experts of the application pilots (i.e., in the work packages WP1-WP3) and documented in the deliverables Dx.1, respectively. Applying an iterative approach, the use cases are matched in a second step with the capabilities of the emerging Future Internet platform, encompassing generic enablers (to be provided by the FI-WARE project as part of the core platform) and environmental enablers (to be provided by ENVIROFI). This general idea is illustrated in Figure 3-1.

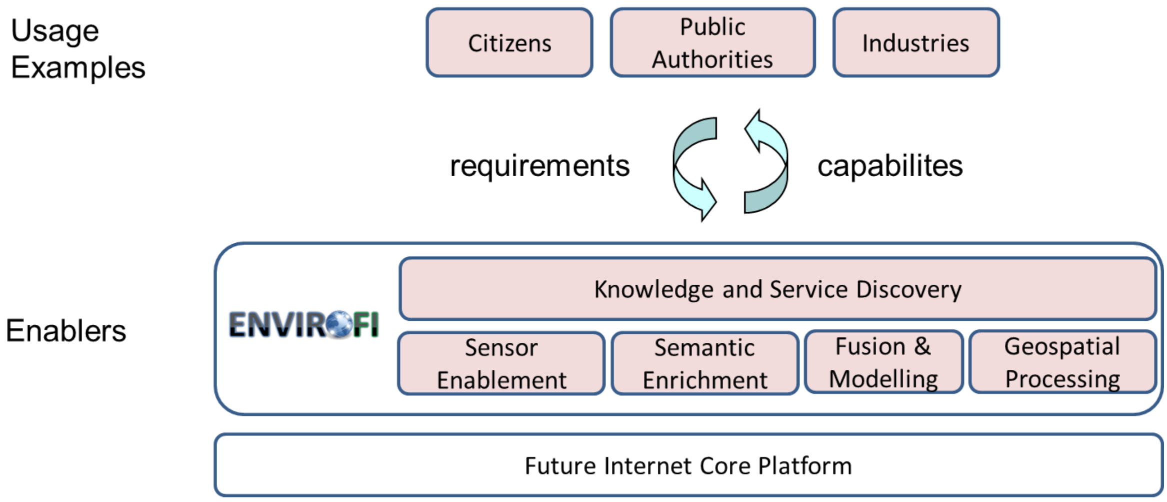


**Figure 3-1.** Overall Idea of the ENVIROFI Use Case Analysis

The next sub-sections describe the instantiation of this idea in more detail. In particular, they comprise

- the description of the analysis process,
- the description of the UC template including an explanation of the UC elements,
- the description of the requirements template including an explanation of the requirements elements, and
- the description of the possible relations between use cases and requirements, hence a kind of UC meta-model.

### 3.2.1 Use Case and Requirements Modelling

Use case modelling has been proven to be an efficient and powerful approach to reach a common understanding of the system itself and its behaviour. In interdisciplinary projects, involving thematic experts from different domains (e.g., air and water) as well as IT-experts, it is as challenging as essential to reach consensus on a common terminology. Otherwise, the consequences would include different interpretations and assumptions about the systems to be developed. Thus to avoid misunderstandings, use case descriptions shall be based on a common vocabulary, stemming from the glossary and the thesaurus whenever possible.

The description of use cases is necessary to capture all functional and non-functional requirements of the system. The use cases also describe the interaction between the users and the system. Use cases are the most common practices for capturing and deriving requirements. The requirements of the system are described in a narrative way with minimal technical jargon. In a nutshell: “a use case describes who can do what with the system and for what” [02 – Cockburn, A. (2001)].

Those quotes indicate that the most important basis to implement the systems foreseen case studies is use case modelling. Cockburn states that use cases are the central aspect in a software development project. The descriptions should be clear and sophisticated.

In this project use cases are described in a semi-formal way, based on a structured textual description in tabular form derived from a template initially proposed by [02 – Cockburn, A. (2001)]. Other recent European research projects (such as SANY[1], EO2HEAVEN[2] and TRIDEC[3]) based the description of

[1] EU FP7 project no. 033564 Sensors Anywhere (SANY) - http:///www.sany-ip-eu

[2] EU FP7 project no. 244100 Earth Observation and ENVironmental modeling for the mitigation of HEAlth risks (EO2HEAVEN) - http://www.eo2heaven.org/

their use cases on this template, too.

Based upon this approach, [01 – Usländer, T. (2010)] proposes in his SERVUS design methodology that additional information about the requested information resources (e.g. type and format of needed data) is necessary to completely describe a use case from both a user's and system's point of view. Furthermore, the requirements should be derivable from the use cases. Three types of requirements can be identified:

- Functional requirements,
- Informational requirements,
- Non-functional requirements.

Functional requirements can be derived from the sequence of actions (main success scenario, extensions and alternative paths). The informational requirements address data that is exchanged between two communication partners, i.e. between users and the system or between system components. The non-functional requirements cover all requirements that do not alter the foreseen functionality of the system, e.g. the quality of data and results.

### 3.2.2 Use Case Analysis Process

Figure 3-2 illustrates the analysis phase as a prelude of the SERVUS Design Methodology [03 - Usländer, T., Batz, T. (2011)]. As part of the project planning there needs to be some agreement of how to document use cases. For this continuous activity a project space has to be created which preferably should be supported by a use case server that is accessible by all participants of the analysis process. In ENVIROFI, this use case server is provided by Fraunhofer IOSB as described in section 3.3.

As a first step of an analysis iteration loop a set of preliminary use cases (UC) is identified, mostly by those thematic experts who drive the project. For each of them an entry in the project space has to be generated. The methodology proposes that use cases are initially described in structured natural language but already contain the list of requested resources.  This small extension with respect to the approach of [02 – Cockburn, A. (2001)] heavily facilitates the transition to the abstract design step (here: the specification of the information model in the Unified Modelling Language UML) but is still very easy to understand by thematic experts. It is described in more detail in section 3.2.3. Hence, this description is the language which is used in the UC discussion that takes place in workshops that are facilitated by the system analyst. Depending on the level of agreement that can be reached the iteration loop is entered again in order to refine or add new use cases.

In order to identify inconsistencies and check the completeness of the UC model, the system analyst may transform the semi-structural UC description into formal UML specifications. However, these UML diagrams should still be on a high abstraction level such that a discussion with the end-user is possible. It is the advantage of this formal transition step already in an early analysis phase to detect inconsistencies and missing information as quickly as possible. The UML specification helps to (re-)discuss and check the use cases together with the thematic experts.

However, in addition to the usual UML use cases they already comprise the links to the set of requested (information) resources, their representation forms and the requirements to create, read, write or delete them. Guidance about these UML diagrams is provided in section 3.2.4. Once an agreement is reached about the set of use case descriptions and related UML specifications it is then up to the system analyst to specify the resulting information model taking the resource model as a first guidance.

[3] EU FP7 project no. 258723 Collaborative, Complex and Critical Decision-Support in Evolving Crisis (TRIDEC) - http://www.tridec-online.eu

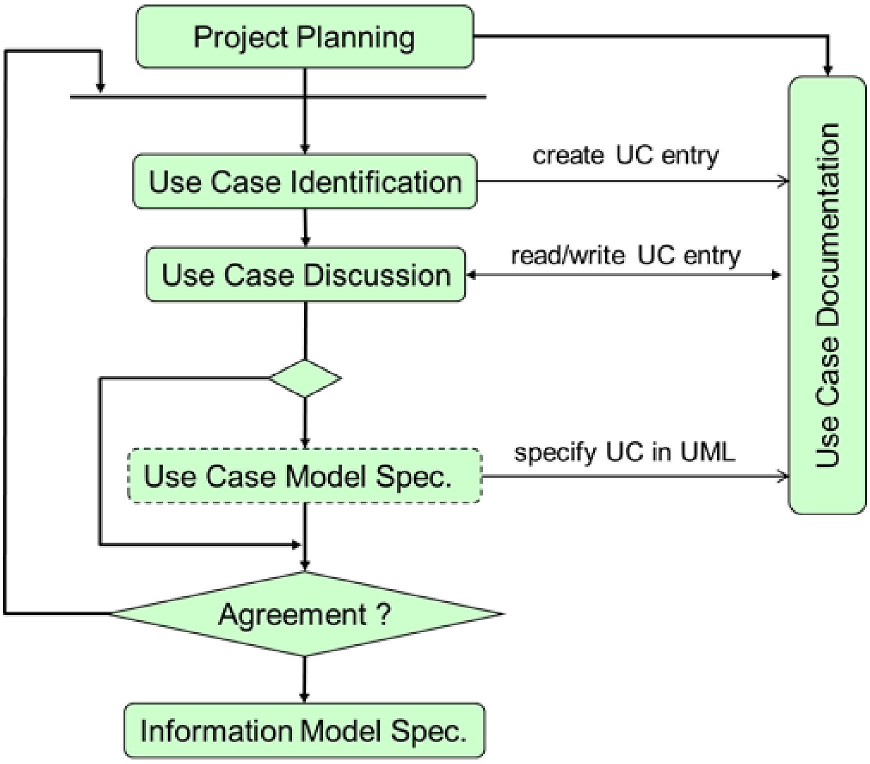


**Figure 3-2.** Procedure of the ENVIROFI Use Case Analysis

### 3.2.3 Use Case Template

Based on the state-of-the-art analysis of the previous section, this section describes the adapted approach for the use case description in this project and also the guideline to fill out this template.

As mentioned above, the approach of [02 – Cockburn, A. (2001)] provides the basis for the use case development in this project. However, the SERVUS methodology proposes that beside the functional and non-functional requirements the informational requirements are very important to complete the use case description. For a more detailed analysis (esp. for IT-experts) and as a first step towards information modelling it is necessary to consider input data, data format, data type, data encoding, and the desired format of the output data, too. Thus the template contains additional issues like ‘Requested Information Resources’.

The common form of a use case description is to describe it from the user’s point of view where only the external perceivable behaviour is reflected. The described system is a black box for the user. This template should be used by both sides, the users and the system developers and operators. Both sides and all involved experts have to understand the use cases in the same way. Especially the IT experts should understand the user’s requirements because they have to develop the IT-components on the basis of the descriptions.

It is foreseen to describe each use case in a semi-formal way. A form was created to structure the textual description. The table represents the use case template and is shown in Table 1. The methodology describes the use case template items, explains what each item mean, instructs how to fill them out and includes additional examples and tips.

Generally, to avoid getting lost in details, [04 – Balzert,H. (2004)] proposes to concentrate on the standard cases for each Use Case (what most likely will happen) in the first iteration of developing a specific Use Case. In the next iteration, extensions and alternative paths are modelled that only happen under certain conditions like optional, seldom or alternative steps.

| Use Case Template | Description | Examples |
| --- | --- | --- |
| Use Case Name | Nam of the use case | Visualise proposed water height after the tsunami event |
| Use Case ID | Unique identifier of a use case according to the following scheme:<br>UC-ENV<workpackage>.<scenario>-<category>-<use case no.>.-V<version no.><br>* workpackage number, i.e. biodiversity (WP1), atmosphere (WP2) and marine environment (WP3)<br>* scenario: The number of the scenario within the WP<br>* category: no particular scheme has to be applied. One can either specify it e.g. with 'mob' (for mobile application) or use 'any' if no specification needed.<br>* (sub) use case number: two digits for each separated by a dot<br>* version number: two digits with V prefixed | UC-ENV1.2-mob-05.06-V02 |
| Revision and Reference | Revision = version number of use case ID<br>Reference = URL of the use case (you get the URL by right-clicking on the entry in the index column) | V02, http://envirofi.server.de/servlet/is/4900/ |
| Use Case Diagram | Description of the UML use case diagram for the actual use case. The diagram should include extend and include relationships if there is any.<br>The actual UML diagram figure may be added at the bottom of the template by uploading a bitmap generated from a UML editor, e.g. Enterprise Architect of SPARX systems. | |
| Status | Status of the use case development | One of the following:<br>• Planned<br>• in progress |
| Priority of accomplishment (optional) | The priority of the use case to be considered when assessing its importance for a development cycle. | One of the following:<br>• Must have: The system must implement this goal/ assumption to be accepted.<br>• Should have: The system should implement this goal/ assumption: some deviation from the goal/assumption as stated may be acceptable.<br>• Could have: The system should implement this |

| | | |
|---|---|---|
| | | goal/assumption, but may be accepted without it. |
| Goal | Short description (max. 100 characters) of the goal to be achieved by a realization of the use case. | System generates alerts based on user observations |
| Summary | Comprehensive textual description of the use case. | The user opens the browser which shows map-window with the water height after the tsunami event in the affected area |
| Category | Categorisation of use cases according to over-all reference architecture. | *To be defined* |
| Actor | List of users of the use case (actors) | Examples may be citizen, administrator or employee of a environmental agency |
| Primary Actor (initiates) | Actor that initiates the use case execution. | |
| Stakeholder (optional) | Company, institution or interest group concerned by the execution of the use case | |
| Requested Information Resources<br><br>(optional) | Information category or object that is required to execute the use case or is being generated during the course of the use case execution.<br><br>The requested information resource shall be listed together with its requested access mode (create, read, update or delete) or "manage" which encompasses all access modes. | • user observation (read)<br>• user-specific effect (read, update)<br>• alert (manage) |
| Preconditions | Description of the system/user status statement) that is required to start the execution of the use case.<br><br>Note that use cases can be linked to each other via „preconditions". This means, a precondition for a use case can be either an external event or another use case. In this case the use case ID should be provided in the field „preconditions". | The user has opened the portal successfully. |
| Triggers (optional) | (External) event that leads to the execution of the use case.<br><br>Note that use cases can be linked to each other via „triggers". This means, a trigger for a use case can be either an external event or another use case. In this case the use case ID should be provided in the field „triggers". | The user chooses water height forecast. |
| Main success scenario | Numbered sequence of actions (use case workflow) to be carried out during the execution of the use case. | 1. User chooses assessment report.<br><br>2. He specifies one or more components (default should be all).<br><br>3. He sets a time-frame (last 24 hours, last week, last month)<br><br>4. The system shows a report as graphical visualisation. |

| | | |
|---|---|---|
| Extensions | Extension of an action of the main success scenario. The action to be extended shall be referred to by its number (e.g. 1) appended by a letter (e.g. 1a). | 1a. The user defines the temporal extent b. The user defines an unavailable temporal extent. A new dialogue window opens and requires a new temporal extent. |
| Alternative paths (optional) | Alternate path through the main success scenario w.r.t. an identified action. | 4a. User can select to view report in different formats, e.g. tabular or graphical map |
| Post conditions | Description of the system/user status (statement) that holds true after the successful execution of the use case. | Report is displayed on the screen. |
| Non-functional requirements | Description of non-functional requirements for this use case w.r.t. performance, security, quality of service or reliability. | Display of report expected after 20 seconds at the latest. |
| Validation statement | List of statements that indicate how to validate the successful realization of the use case. | |
| Notes | Additional notes or comments (also by other users). | |
| Author and date | Author of use case, date of last edition. | Fraunhofer IOSB, 2011-10-11 |

**Table 3-1:** Description of the Use Case Template

### 3.2.4 Requirements template

In the following the current template for the analysis of the requirements for generic and specific enablers is shown. It applies the agile software requirements analysis methodology based upon so-called backlog items [05 - Leffingwell, D., 2011].

Note that this table represents the current status and may change during the course of the project !

| Field | ENUM | Description | Comments |
|---|---|---|---|
| Id | No | Used to uniquevocally identify the backlog entry | We should agree in a meaningful format, e.g.: <category>.<scope>.<id-number> (EPIC.UC.16) where:<br><category>::= "Theme" \| "EPIC" \| "Feature" \| "User story" \| "Task"<br><scope>::= "UC" \| "FIWARE" \| "OTHER" \| …<br>Some people would prefer acronym instead of <id-number><br>Some people mentioned to have name of the project in the <scope> instead of "UC"<br>It is not clear whether scope should be part of the Id, given the fact that scope changes over time |
| Name | No | Descriptive name of the entry | This will be an acronym. |
| Goal | No | Short phrase describing the goal for this entry | |
| Version | No | Version associated to the entry. Helpful to monitor progress and follow-up modifications | Need a version scheme !<br>FI-WARE to come with a proposal |
| Source | Yes | Project or organization who identified the use case / feature | |
| Source contact | No | Contact point in Source project or organization (was "Author") | Contact point of the source |
| Stakeholder | Yes (per item in list) | List of additional projects or organizations interested in coverage of the use case / feature (the source is considered to be a stakeholder) | The value of this field may change over time |
| Scope | Yes | Could be:<br>"Platform" = it relates to a functional or non-functional feature required at platform level (not yet agreed whether "common" or "generic")<br>"Platform Generic" = It relates to a feature required at platform level and general purpose<br>"Platform Common" = It relates to a functional or non-functional feature required at platform level but whose applicability is restricted to applications in a few number of domains (Usage Areas)<br>"Application" = It relates to a user story related to some functional o non-functional feature required at application level<br>"Global" = It relates to some functional or non-functional feature required both at platform (generic or common) and application level | The value of this field may change over time |

| | | | |
|---|---|---|---|
| | | "Not Yet Determined" = when decision still has not taken place | |
| Status | Yes | Should be "Pending" (still not revised), "Planned" (is in the roadmap), "Under execution" (being developed in current sprint), "Done" (has been already developed) or "Deprecated" in which case the Description field should explain why and the list of Ids of entries replacing it when applicable. | Maybe we need to establish here a link to some ticket in a ticket management tool like bugzilla or the FusionForge tracker |
| MoSCoW priority | Yes | MUST - Features that absolutely have to be done are categorized as Must. If any of these features are not done, the project will be considered a failure.<br><br>SHOULD - Features that are important to the success of the project, but are not absolute musts (they have a workaround or will not cause the project to fail) are categorized as Should.<br><br>COULD - Features that are nice to have but are not core features are categorized as Could.<br><br>WONT - Features that are not going to be implemented this time are marked as Won't. | In the first place, while "Owner" has not been identified, this priority will be assigned by the Source. Stakeholders may agree to change it. Final value will be assigned by the "Owner", although a new entry may be created, keeping the previous one for the sake of history (which would change to "Deprecated" status pointing to the new entry).<br>Clarification on WONT priority: Why should we have won't features in the backlog? There are two reasons. One is that feature priorities can change as the project goes on. These features could have started as Should and been re-prioritized to Wont, and they may be re-prioritized back again. The second is that these features are a starting point to the second version.<br>Remember that features prioritized as "Must" should be only those features without which the project cannot be put into production, and will cause the project to fail. When you decide to mark a feature as "Must", ask yourself if the project will have to be canceled if this feature is not implemented. Only if the answer is yes should you go ahead and prioritize it as Must. |

| | | | |
|---|---|---|---|
| Relative priority | No | Priority number relative to the same MoSCoW priority | In the first place, while "Owner" has not been identified, this priority will be assigned by the Source. Stakeholders may agree to change it. Final value will be assigned by the "Owner", although a new entry may be created, keeping the previous one for the sake of history (which would change to "Deprecated" status pointing to the new entry).<br>We may use maybe a number of 2 digits, with the first one linked to the MoSCoW priority and the second to the relative priority. That would allow to order entries just based on this number. Thus, we may have:<br>Priority 35 = "Should" with priority 5 among those labeled as "Should" |
| Chapter | Yes | Will refer to chapters in FI-WARE, when dealing with entries related to FI-WARE GEs:<br>- "Cloud"<br>- "Data/Context"<br>- "Apps"<br>- "IoT"<br>- "I2ND"<br>- "Security"<br>When dealing with entries with "Application" scope, may contain value of chapters relevant to the application. | |
| Enabler | No | Only applicable to Platform features. It identifies the Enabler to which this entry (feature) in the backlog applies. | |
| Category | Yes | Would be "Theme", "EPIC" or "user story"/"feature" (depending on how we want to name fine-grained items in the backlog) | Still pending on number of categories we want to consider and names for the categories. |
| Description | No | Description of the feature. FINEST proposed the following information but this will not be mandatory:<br>- Actors<br>- Primary Actors<br>- Facades<br>- Preconditions<br>- Triggers<br>- Main success scenario ("How to demo")<br>- Extensions<br>- Alternative paths<br>- Postconditions<br>- Notes | |

| | | | |
|---|---|---|---|
| Rationale | No | Rationale of why the feature is needed | (Similar to goal ?) |
| Owner | Yes | Name of project taking care of it | |
| Owner contact | No | Name of person in Owner project taking care of it | |
| Complexity | Yes | Number describing how complex supporting the use case / feature will be. Will map to the following categories:<br>- XXL: Costs quite a lot<br>- XL: Costs a lot<br>- L: Has a significant cost<br>- M: Medium cost<br>- S: Doesn't take that much<br>- XS: It's almost trivial | |
| Creation Date | No | Date of creation | |
| Last modified | No | Last date at which it was modified (any field) | |

**Table 3-2:** Description of the FI-PPP Requirements Template

### 3.2.5 Relation types between use cases and requirements

The following types of relations have been implemented to link **use cases** to other use cases or to link use cases to **requirements** (to be defined by WP5 and WP4 as requirements for generic and specific enablers in the context of the Future Internet project cluster):

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.
- Example: The "generic use cases" (or also called “abstract use cases”) provided by WP4 are abstracted from the use cases provided by WP1-3. Generic use cases are abstract in the sense that they are independent of the application domains (e.g. water and air) and hence may be re-applied to other application domains. This approach facilitates the mapping to requirements and decreases the number of “*is derived from*” links (see next item) from requirements to use cases. The generic use cases identified and described in WP4 are attached to this deliverable as annex A (section 9.1).

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

UC to REQ:

- *maps to* (inverse relation: *is derived from*): a UC is mapped to a REQ defined by WP4 (--> generic enabler requirements) or WP5 (--> specific enabler requirements).

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.

These relations are illustrated in Figure 3-3.

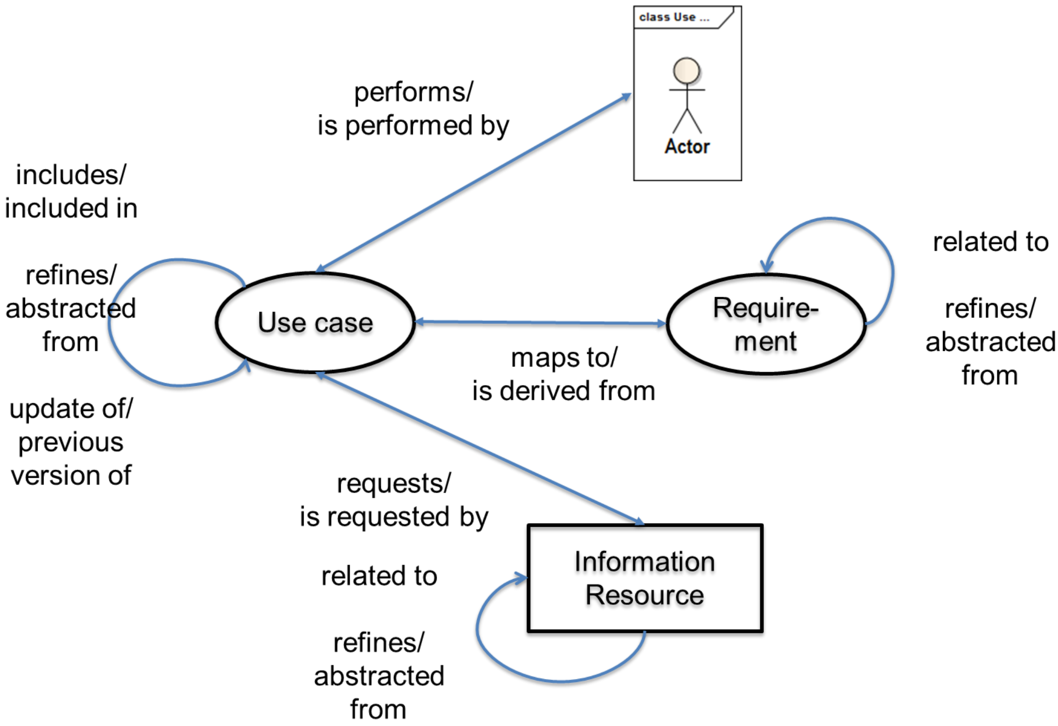


**Figure 3-3.** Relationships between Use Cases, Requirements, Actors and Information Resources

Further important concepts in the use case modelling are **actors** and **(requested) information resources**. For completeness, they are also contained in Figure 3-3 together with their relationships although they are not yet implemented as distinct identifiable concepts in the UC server (see section 3.3). The possible relations are as follows:

UC to Actor:

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

UC to Information Resource:

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

Information Resource to Information Resource:

- *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.

### 3.2.6 UML Representation

After having described the use cases in a semi-formal way using the template, a more formal step is recommended to represent the use cases. They are modelled in a formal graphical way by using a UML tool, e.g. the Enterprise Architect of SPARX Systems. Figure 3-4 shows the general schema for such UML representations. In addition to the conventional UML use case diagrams they comprise the links to the requested information resources.

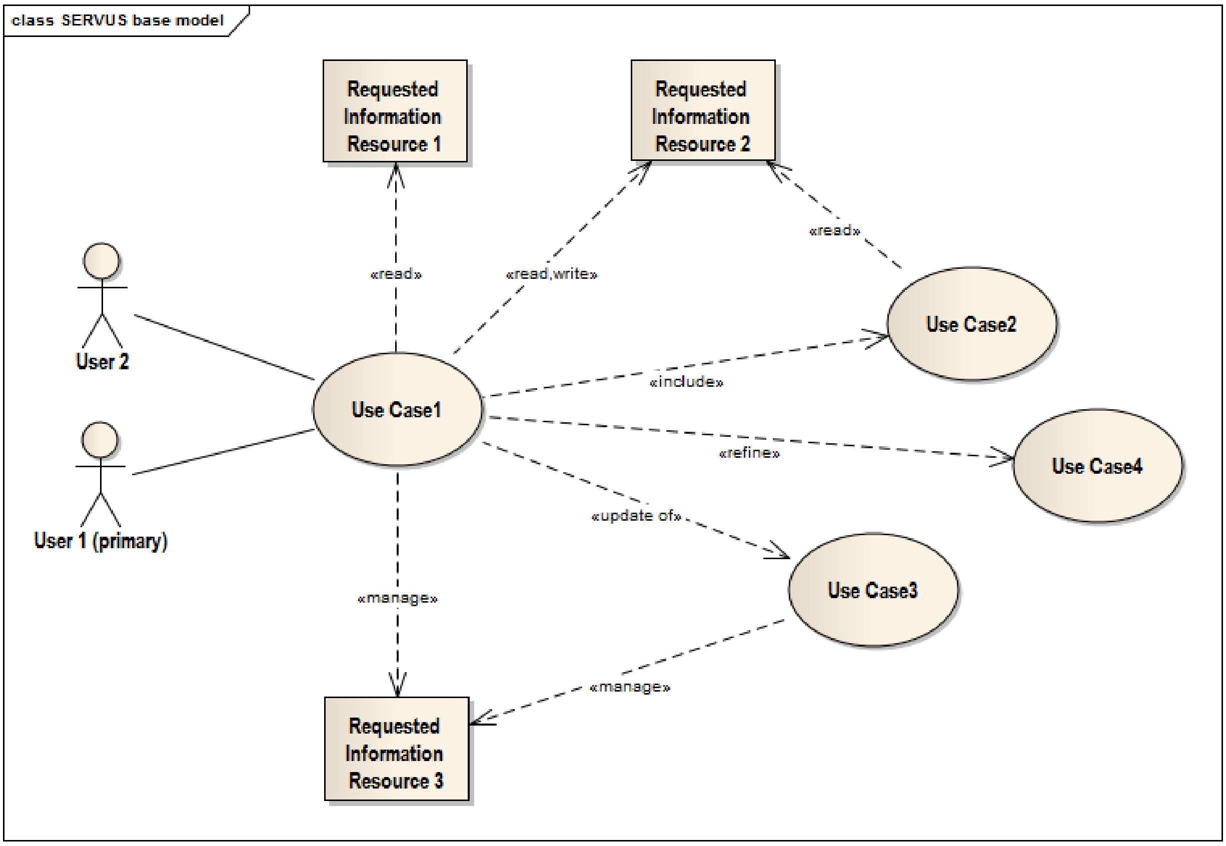


**Figure 3-4.** Schema for the Graphical Representation of a Use Case

The graphical description is divided into three parts:

- the upper region comprising
  - all information objects (requested information resources) necessary for the execution of the use case,
- the middle section comprising
  - all actors, and
  - the main use case and all related use cases, and
- the lower section comprising
  - the additional result objects (also in terms of requested information resources).

According to section 3.2.4 we distinguish the following relation types between these graphical elements[4]:

a) Relation types between actors and use cases:
   - Here the relation type “uses” is applied, i.e. an actor <<uses>> a use case.

b) Relation types between use cases:

[4] In UML marked as <<stereotypes>> with the associations.

- The relation type <<include>> may be applied.
- The relation type <<refine>> may be applied.
- The relation type <<update of>> may be applied.

c) Relation between Use Case and Information Object:

- This relation expresses the involved information objects (requested information resources). In most cases a distinction is made between a <<creates>>, <<reads>>, <<updates>> or <<writes>> relation, expressing the access rights to create an instance of the related object, to read its attributes, to write (update) its attributes or to delete an instance of the related object. All these relations together may be encompassed by the relation <<manages>>.

d) Relation between result objects and further information objects.

- The relation “depends on” describes that the existence or the contents of the result object depends on the existence or contents of another information object.

## 3.3 Tool Support

This section describes the UC server that is available for ENVIROFI team members at http://envirofi.server.de. It enables the edition and management of the UC and requirements descriptions including their interdependencies between these concepts. The UC server has been implemented by Fraunhofer IOSB using the basic functionality of its own content and knowledge management system WebGenesis®.

In order to use this server a password-protected account is required which identifies the user of the server. There are two types of users in two different roles:

- Use case editors: These users may edit use cases and requirements.
- Administrators: These users may edit use cases and requirements, too, but in addition have access to all administrative functionalities of the UC server.

The objective of the UC server is to provide a Web-based collaborative tool to share the edition of use cases. Due to the possibility to link use cases with other use cases (see the “refines use case” link illustrated in Figure 3-5) and to requirements, it also enables the easy navigation and browsing through the use cases and requirements.

Furthermore, the UC server allows to automatically generate documents in pdf format out of the use case and requirements descriptions. This facility is being used in order to generate the following reports:

- list of generic (abstract) use cases summarized in section 6 and provided as annex A to this deliverable (section 9.1)
- list of requirements provided as annex B (section 9.2) to this deliverable.

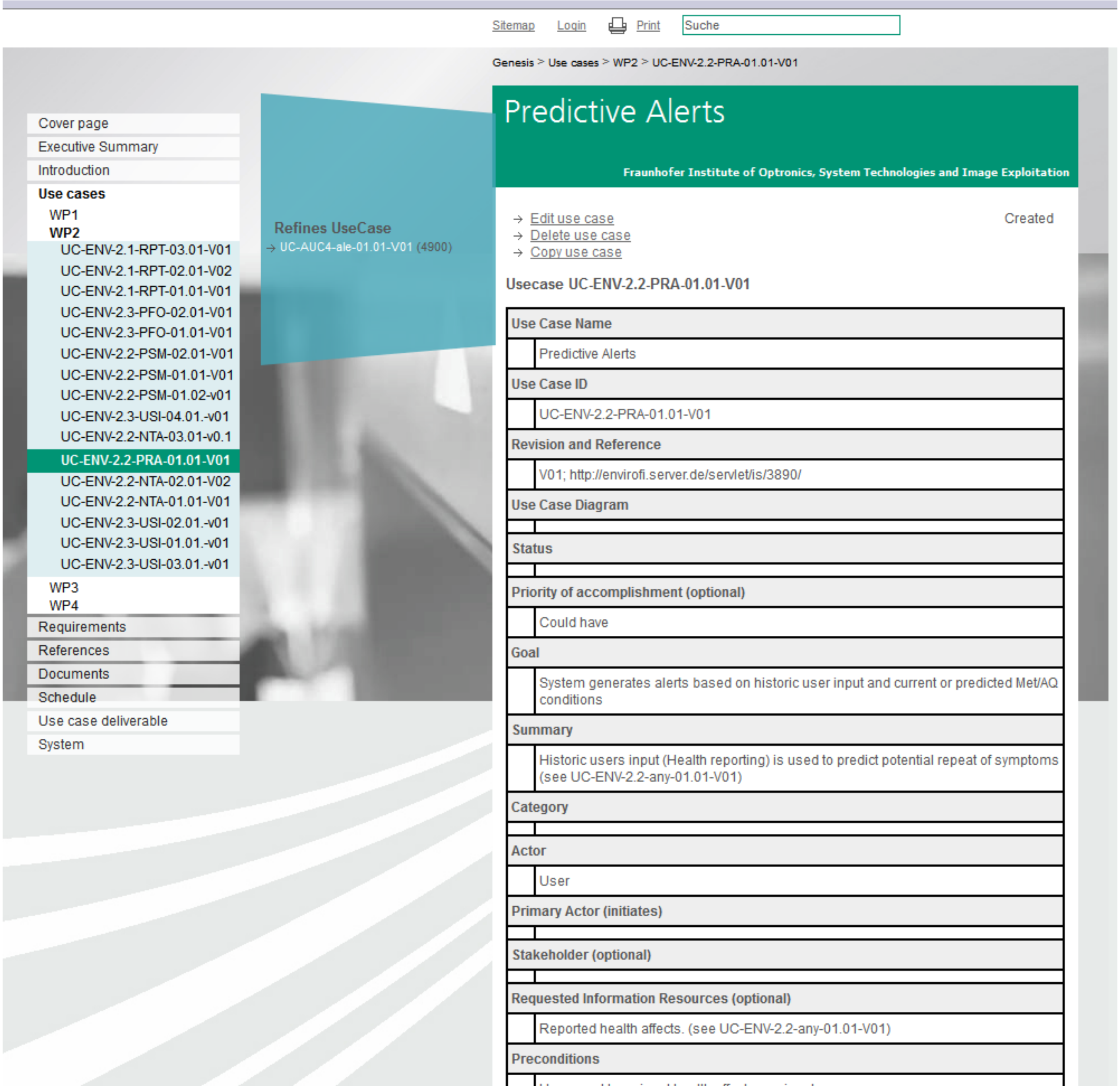


**Figure 3-5.** Screenshot of the Use Case Server (source: Fraunhofer IOSB)

# 4 Thematic Areas of the Use Case Analysis

The thematic areas for which a use case analysis has been carried out in ENVIROFI are the biodiversity, air quality and marine area. These are handled by the three thematic work packages WP1, WP2 and WP3, respectively. They provided an initial set of use cases via the ENVIROFI deliverables D1.1, D2.1 and D3.1. These use cases have then been revised, refined and harmonized during the preparation of the second set of deliverables D1.2, D2.2 and D3.2. The use cases describing the application-specific scenarios can be briefly summarized as follows:

- **WP1** (Bringing **Biodiversity** into the Future Internet) (see Figure 4-1), detailed information is available from ENVIROFI deliverable D1.1.
- **WP2** (Personal Information System for **Air** Pollutants, Allergens and Meteorological Conditions) (see Figure 4-2), detailed information is available from ENVIROFI deliverable D2.1.
- **WP3** (Future Internet Collaborative Usage of **Marine** Environmental Assets), (see Figure 4-3), detailed information is available from ENVIROFI deliverable D3.1.

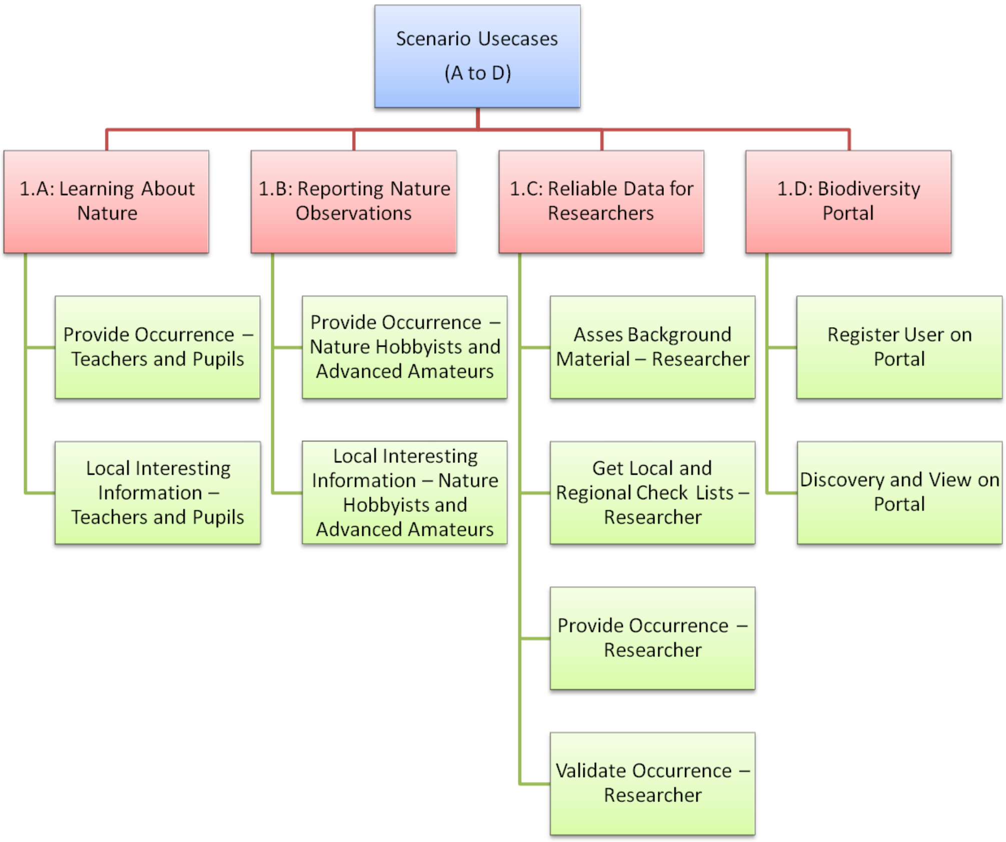


**Figure 4-1.** Overview of the ENVIROFI WP1 Use Cases on “Biodiversity”

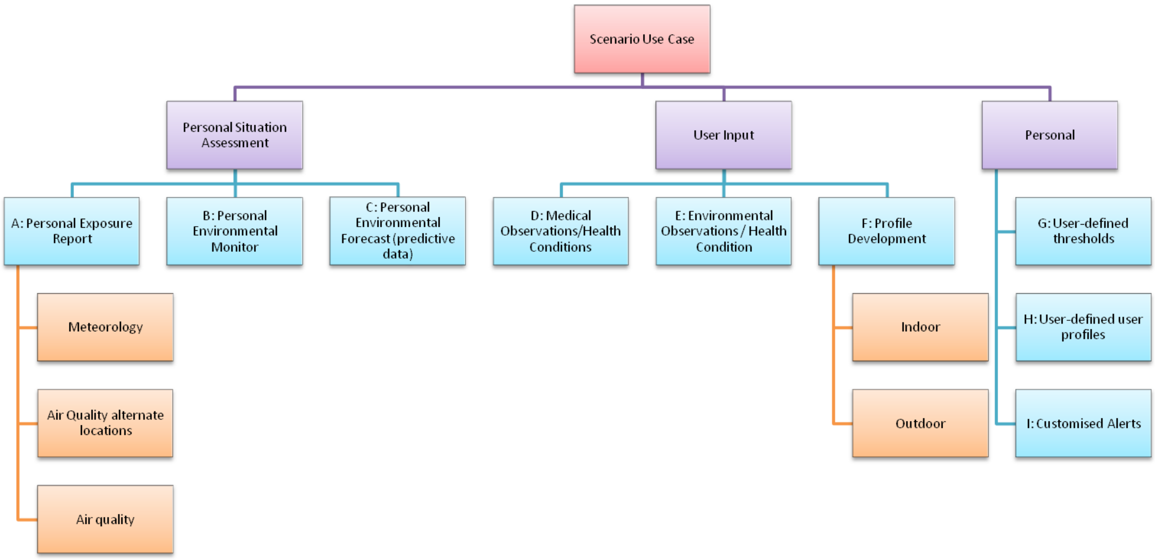


**Figure 4-2.** Overview of the ENVIROFI WP2 Use Cases on “Air Quality”

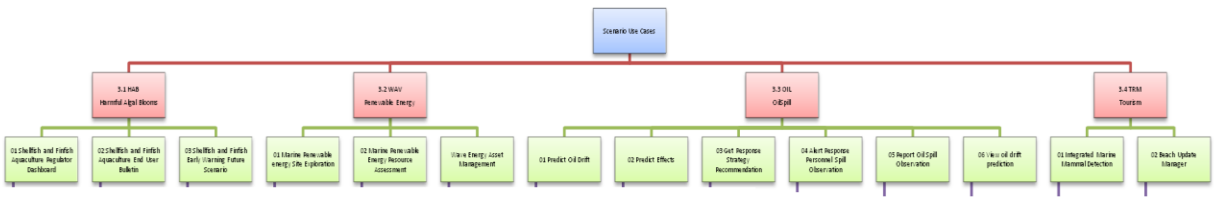


**Figure 4-3.** Overview of the ENVIROFI WP3 Use Cases on “the Marine Environmental Area”

# 5 Categorization of Enablers

The identification of generic and specific enablers is performed based on a combination of top-down and bottom-up analysis. There is no unique approach of categorizing the resulting enablers.

Basically, ENVIROFI distinguishes between the following two approaches:

- Lifecycle-based approach according to the generation of environmental knowledge (section 5.1), and an
- Architecture-based approach considering ISO classification schemes of services as well as cloud computing terminologies (section 5.2).

## 5.1 Lifecycle-based Approach

### 5.1.1 Lifecycle Components

Components which have been identified in a recent activity of the European Committee for Standardisation (CEN), Technical Committee (TC) 287 for building a reference model for spatial data infrastructures (SDI) [07 – CEN (2011)] are re-used, see also Figure 5-1.

The lifecycle-based perspective for the identification of enablers comprises both a service-centric and a data-centric view. Notably, the service-centric view could be applied to any service-oriented system. Only the Data Centric View contains instantiations, which are specific for the geospatial and environmental domains. Likewise, GeoPortals are a specific type of geospatial applications.

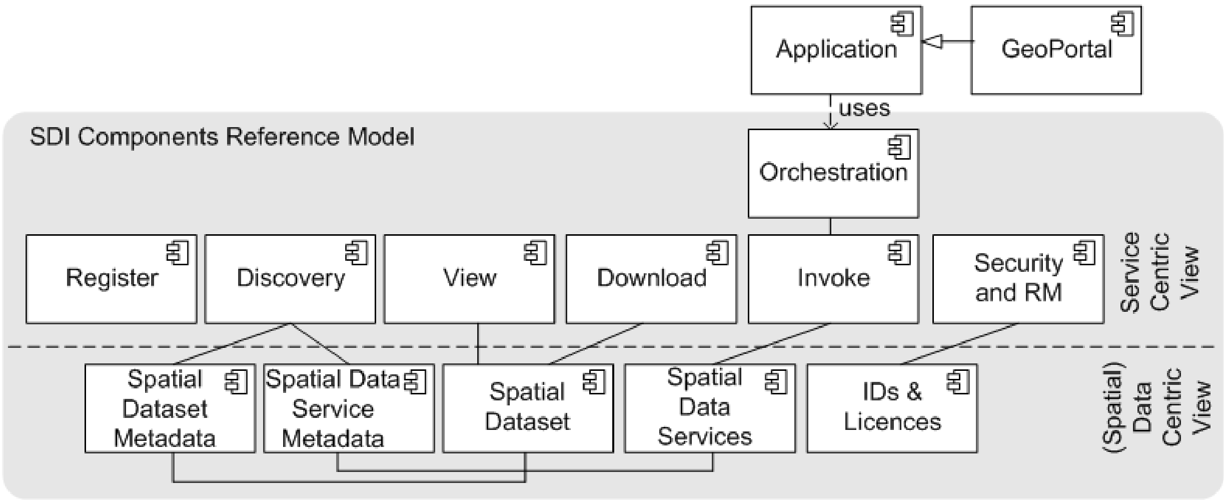


**Figure 5-1.** Core Components of the SDI Reference Model (derived from [07 – CEN (2011)]).

The primary organizing structure is determined by the following generic core lifecycle components (corresponding to the service centric view in Figure 5-1):

- **Register**: for describing and publishing resources.
- **Discovery**: for searching for and discovery of resources.
- **View**: for visualising of resources.
- **Download**: for downloading and exchanging resources.
- **Invoke**: for interacting with resources.
- **Orchestration and Composition**: for providing aggregated resources including in particular workflows for service composition.

- **Security and Rights Management**: for managing access rights to resources.

On a secondary level, these components encompass both a data-centric and service-centric view as shown in Figure 5-1.

First, we introduce the roles, which are involved in the generation of knowledge about the environment and define the overall added-value chain. In a second step, we present common requirements for future environmental services. In doing so, we provide a bridge between practical environmental applications and the wider political framework. These findings could equally be applied to other geospatial and non-geospatial domains beyond the environmental domain.

### 5.1.2 Value Chain of Environmental Knowledge Generation

When analyzing the requirements of environmental services for the terrestrial, atmospheric and marine sphere, six roles may be identified each of which contributes to the generation of environmental knowledge and are therefore part of the value chain [07 – CEN (2011)].

- **Observer**, being the initial source of information about the environment. This may reach from sensors measuring weather conditions to citizens observing species occurrences.
- **Publisher**, making a resource, such as an observation, discoverable to a wider audience, e.g. by providing required resource descriptions (metadata).
- **Discoverer**, being the entity that finds a resource, e.g. species occurrence data, based on all available descriptions.
- **Service Provider**, making information or an environmental model accessible to (and usable by) the wider audience, e.g. by offering a standard based service for data download.
- **Service Orchestrator**, being responsible for combining existing services in a way that they create information for a distinct purpose, i.e. an environmental application focusing on a particular sphere, such as terrestrial biodiversity.
- **Decision Maker**, consuming an environmental application in order to retrieve decision supporting material and making a final decision based on the information available, e.g. designating a new protected area.

Consequently, the process workflow can be summarized as in Figure 5-2. Note that workflow services may themselves get published in order to serve as building blocks for more complex environmental solutions.

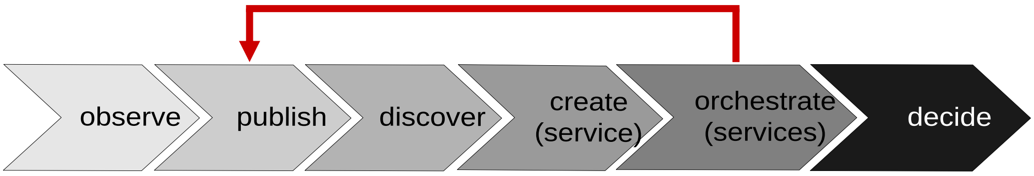


**Figure 5-2.** Added value chain of environmental knowledge generation [07 – CEN (2011)]

### 5.1.3 Overview of Stakeholders

The roles identified above (section 3.1) are played by a variety of individuals and organizations. In a nutshell, those can be defined as:

- **Citizens** of a particular social, political, or national community;
- **Environmental agencies** on sub-national, national and European level;
- **Public authorities** of national and regional and other level;
- **Industries** from the primary, secondary and service sector;
- **Platform providers** offering frameworks on which applications may be run;

- **Infrastructure providers** offering physical components and essential services;
- **Sensor network owners** holding the sensor and basic communication hardware.

| | observe | provide | discover | create | orchestrate | decide |
|---|---|---|---|---|---|---|
| Citizens | x | x | x | x | x | x |
| Environmental agencies | x | x | | x | | x |
| Public authori-ties | | x | | x | | x |
| Industries | | | x | x | x | x |
| Platform pro-viders | | | | x | | |
| Infrastructure providers | | | | x | | |
| Sensor network owners | x | (x) | | (x) | | x |

**Table 5-1.** Added-value chain of environmental knowledge generation [6].

Table 1 provides an overview of the manifold mappings between these stakeholders and the different roles in the value chain of environmental knowledge generation. Notably, citizens can play all roles, they may even discover available information and provide new services (mash-ups). The decisions they may take are on individual level, such as “Should I travel through an area with bad air quality?”.

### 5.1.4 Requirements for a Next Generation of Environmental Services

Given the above, we can now identify the requirements for a next generation of environmental services in Europe. They can be summarized as follows:

- publication, discovery, access and visualization of environmental data sets;
- planning, publication, discovery, access and visualization of measurements;
- publication, discovery, access and visualization of objective, semi-objective and subjective observations by end users;
- transformation of data sets and fusion of observations;
- publication, discovery and access to environmental models and simulations;
- composition and invocation of workflows;
- support and enforcement of data and service policies based on identity, licenses, trust chains, etc.;
- publication, discovery, access, visualization and annotation support for controlled vocabularies, taxonomies, and ontologies;
- integration with the Semantic Web and Web 2.0; and
- interoperability with existing and planned infrastructures in the context of:
  - the most relevant initiatives at international level, such as INSPIRE [06 – European Parliament (2007)], GMES [08 – CEC (2009)], SEIS [09 – CEC (2008)], GEOSS[14 – GEO (2008)],
  - relevant well-established communities, including research and e-government infrastructures [10 – CEC (2010)], and

- the mode relevant policies on international level, above all related to Public Sector Information (PSI) [11 – European Parliament (2003)].

Dedicated components (environmental enablers) should support these requirements. They should be designed and developed leveraging existing architectural approaches and technical specifications, and re-using/extending existing tools. Particular attention should be paid to open international standards and communities-of-practice specifications, and to open source components in order to make the resulting system more flexible and scalable, see also [12 – isa (2010)].

## 5.2 Architecture-based Enabler Classification

The lifecycle-based enablers and relevant applications can further be described in terms of their architectural components and enablers/services. ENVIROFI distinguishes between two classifications schemes: one based upon ISO 19119 and another based upon terms used in the cloud computing domain.

### 5.2.1 Architectural Classification according to ISO 19119

The following figure shows how the different types of enablers can be related in the context of a complete end-to-end ICT architecture.

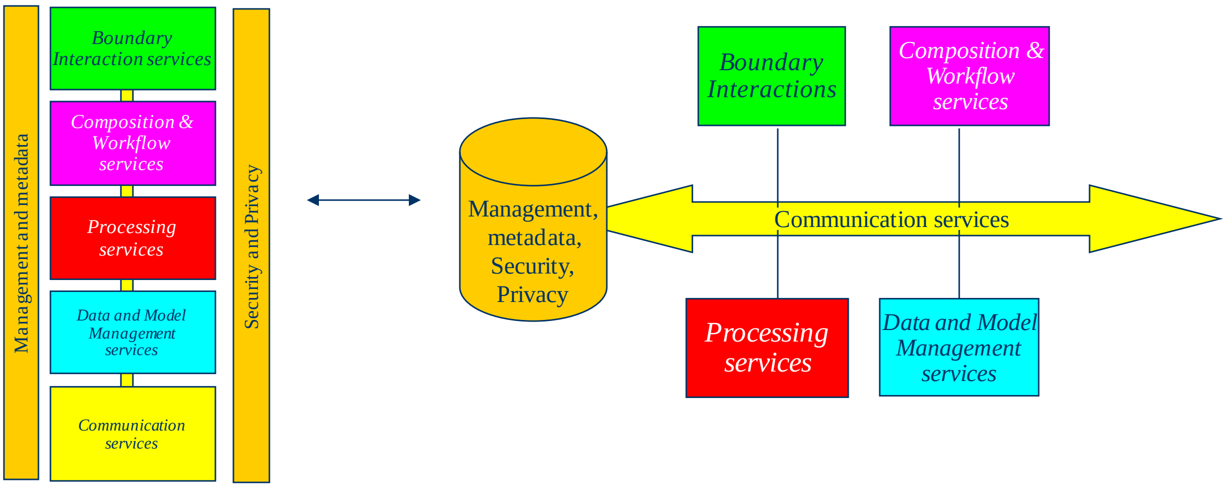


**Figure 5-3.** Relationships of enablers in both a layered and a bus architecture

Figure 5-3 shows the relationship of different enabler categories both as a layered architecture and as a bus architecture. Here, the taxonomy of the enabler types is in accordance with ISO 19119 Geographic information – Services, clause 8.3. [13 – ISO (2005)]. It follows the approach to define both generic domain-independent and specific enablers, such as geospatial and environmental specific enablers, in each of the following six groups which are colour-coded in Figure 5-3:

- **Boundary Interaction Enablers** are enablers for the management of user interfaces, graphics, multimedia and for the presentation of compound documents. Boundary Interaction services have been defined to not only include human interaction services, but also other system boundaries like sensor and actuator services. Specific enablers focus on providing capabilities for managing the interface between humans and Geographic Information Systems and location-based sensors and actuators. This class includes also graphic representation of features, as described in ISO 19117.
- **Workflow/Task Enablers** are services for support of specific tasks or work-related activities conducted by humans. These enablers support use of resources and development of products involving a sequence of activities or steps that may be conducted by different persons. The specific enablers focus on workflow for tasks associated with geographic and environmental

information — involving processing of orders for buying and selling of geographic information and services. These services are described in more detail in ISO 19119.

- **Processing Enablers** perform large-scale computations involving substantial amounts of data. Examples include enablers for providing the time of day, spelling checkers and services that perform coordinate transformations (e.g., accepting a set of coordinates expressed using one reference system and converting them to a set of coordinates in a different reference system). A processing service does not include capabilities for providing persistent storage of data or transfer of data over networks. Specific enablers focus on processing of geographic information. ISO 19116 is an example of a processing service. Other examples include services for coordinate transformation, metric translation and format conversion.
- **Model/Information Management Enablers** are enablers for management of the development, manipulation and storage of metadata, conceptual schemas and datasets. The specialization of this class of enablers focuses on management and administration of geographic information, including conceptual schemas and data. Specific services within this class are identified in ISO 19119. These services are based on the content of those standards in the ISO 19100 series that standardize the structure of geographic information and the procedures for its administration, including: ISO 19107, ISO 19108, ISO 19109, ISO 19110, ISO 19111, ISO 19112, ISO 19113, ISO 19114 and ISO 19115. Examples of such services are a query and update service for access and manipulation of geographic information and a catalogue service for management of feature catalogues.
- **Communication Enablers** are enablers for encoding and transfer of data across communications networks. The specific enablers focus on the transfer of geographic information across a computer network. Requirements for Transfer and Encoding services are found in ISO 19118.
- **System Management and Security Enablers** are enablers for the management of system components, applications and networks. These services also include management of user accounts and user access privileges. The specific enablers focus on user management and performance management, and on Geo Right Management

These six categories of enablers have been identified through an end-to-end architectural analysis. Since the initial version of this approach in the ISO 19101 and 19119 standards around 2001 they have been considered to be sufficient for most of the identified service types and enablers, with the escape mechanism that many new instances will be put into the processing category. There are also situations where tools and applications are composite and contain components that will span multiple categories, and also for this reason the lifecycle-based classification has been found useful as an additional classification.

### 5.2.2 Architectural Classification according to Cloud Computing Service Categories

The different service types can also be categorized according to their relevance for emerging cloud services, starting with a classification for the application level and software as a service (SaaS), but also further down to platform as a service (PaaS) and infrastructure as a service (IaaS).

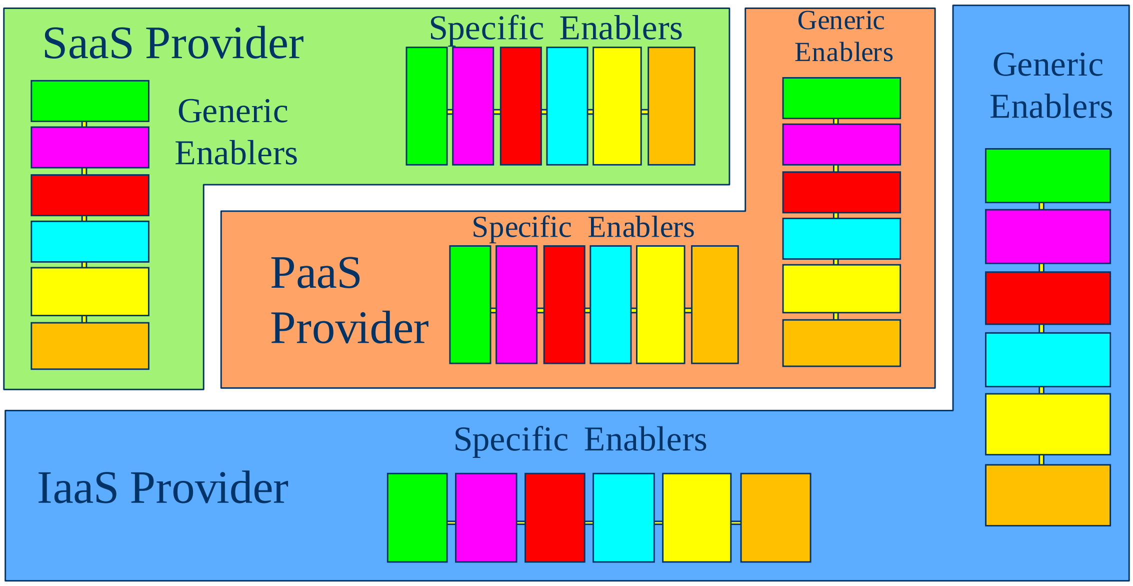


**Figure 5-4.** Generic and specific enabler types for SaaS, PaaS and IaaS

The initial generic enabler areas identified by the FI-WARE project is targeted at providing further support in many of the areas identified through the lifecycle-based perspective and the architectural perspective here. The initial six areas can be mapped to the architectural areas as follows:

- **Cloud hosting** on the infrastructure level (IaaS) is addressing generic enablers in particular related to processing and model/information management.
- **Data/Context management** (with intelligent services) is related to model/information management enablers on the SaaS and PaaS level.
- **Application Services framework** is related to processing and system management enablers on the PaaS level.
- **IoT Service enablement** is related to boundary enablers on the SaaS and PaaS level
- **Interface to Network and Devices** (I2ND) is related to communication enablers on the PaaS and IaaS levels.
- **Security** is related to system management/security enablers on the SaaS and PaaS level.

In the course of the FI-PPP activities on the identification of further generic and specific enablers it is assumed that more enablers will be found for all of the different enabler areas across all of the cloud computing levels from SaaS to PaaS and IaaS.

# 6 Overview about Generic Use cases and Requirements

This section provides an overview about the generic use cases and the requirements that are derived from the domain-specific UCs as selected for this phase of the analysis (see section 6). Generic UCs are abstract in the sense that they are independent of the application domains (e.g. water and air) and hence may be re-applied to other application domains.

The detailed description of the generic UC is automatically generated from the WP4 contents of the UC server and provided as annex A to this deliverable. Currently, 19 generic use cases have been identified and linked to their origins in WP1-3.

Analogously, the detailed description of the requirements for generic and specific enablers is also automatically generated from the WP4 and WP5 contents of the UC server and provided as annex B to this deliverable. Currently, 54 requirements have been identified and linked to the generic use cases of WP4 and/or to the use cases of WP1-3.

This approach enables a full traceability of the use cases and requirements in both directions.

These generic use cases as well as the requirements are categorized as follows:

- functional requirements (FUN) as seen by the user
- requirements on performance (PER)
- requirements to data and meta-information (DAT)
- requirements to new knowledge (KNO)
- requirements on connectivity (CON)
- requirements on tools (TOO)
- requirements on User-Interface (XUI)
- requirements on Security (SEC)
- requirements on Privacy (PRI)
- requirements on Trust (TRU)
- other (OTH), i.e. everything not covered in the other requirement categories.

# 7 Lessons Learnt and Conclusions

In its use case analysis approach and the derivation of associated requirements the ENVIROFI project follows an agile approach. This means that both use cases and requirements are developed in an iterative way instead of specifying them in "one shot", exploiting the knowledge that is being gained in each iteration cycle. However, this approach also applies to the methodology itself as there is no standard SOAD methodology fitting to the purposes of the project upon the project could rely upon.

Hence, having carried out the use case analysis and requirements for this first iteration cycle in the ENVIROFI project, some shortcomings in the application of the use case and requirements templates have been revealed. These were largely caused by the high degree of freedom within the template descriptions. As a consequence, some individual elements of the template were interpreted differently such that there is no consistent content of these elements across the whole set of use cases and requirement descriptions. Although agility means adaptions according to new knowledge gained, the application of changes is not possible at all times but also at well-defined milestones in order not to endanger the achievements of the ongoing work, here the existing use case and requirements descriptions.

In order to improve this situation in the follow-up work, i.e. the second iteration cycle, we suggest to formalize both templates as well as the possible relations between use cases and requirements. Among others, this encompasses the following issues:

- Definition of well-defined lists (code lists) for those template elements where no free text is allowed. This leads to a restriction in the entries that are allowed. Example: The element "actor" of the use case element could be restricted to those actors that are being considered relevant for a given thematic area, such as "citizen", "administrator" or "expert".
- Definition of rules that apply to the elements across the template entries. This helps in improving the level of consistency within one use case or requirements description.
- Definition of a formal model the exactly specifies the relations between use cases and requirements.

Finally, following such a formalization will lead to a kind of "meta-model" for the ENVIROFI use case and requirements analysis methodology. It may be specified using a formal language such as UML or even an ontology specification language. As a consequence, it will be possible to specify consistency rules and to apply plausibility checks across the specifications which will help in the quality assurance process of the user requirements specification.

However, usability and user-friendliness for non-IT experts as one of the major objectives of the use case analysis approach, must not be compromised. Hence, there is a need for associated tool support, i.e. to accompany the formalization by an evolution of the existing tool following the rules of the meta-model.

As a conclusion it may be said that the analysis of the user requirements of the three ENVIROFI thematic areas in terms of use cases as well as the derivation and identification of requirements would not have been possible without a rigid application of the methodology described in this deliverable and the Web-based use case server used to support their shared edition and documentation having in mind the huge amount of information to be gathered and harmonized. Furthermore, the automatic generation of a written report out of the tool content following the layout guidelines of ENVIROFI heavily reduces the editing effort.

The ENVIROFI project will analyse the shortcomings and investigate how they may be improved for the second iteration cycle in close collaboration with the methodology used to exchange and discuss the requirements for generic and specific enablers in the FI-PPP community.

# 8 References


| | References |
|---|---|
| 01 | Usländer, T., 2010. Service-oriented Design of Environmental Information Systems. PhD thesis of the Karlsruhe Institute of Technology (KIT), Faculty of Computer Science, KIT Scientific Publishing. ISBN 978-3-86644-499-7. |
| 02 | Cockburn, A., 2001. Writing Effective Use Cases. Addison-Wesley, Reading, ISBN-13 978-0-201-70225-5 |
| 03 | Usländer, T. , Batz, T., 2011. How to Analyse User Requirements for Service-Oriented Environmental Information Systems. In: Hřebíček, J., Schimak, G., Denzer, R. (eds.) ISESS 2011. IFIP AICT, vol. 359, pp. 165–172. Springer, Heidelberg. |
| 04 | Balzert, H., 2004. Lehrbuch der Objektmodellierung. Spektrum-Akademischer Verlag. |
| 05 | Leffingwell, D., 2011. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise (Agile Software Development Series). Addison-Wesley Professional; 1 edition (January 6, 2011). ISBN-13: 978-0321635846. |
| 06 | The European Parliament and of the Council, 2007. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE). Official Journal on the European Parliament and of the Council. |
| 07 | European Committee for Standardization (CEN), 2011. TR15449 Geographic information — Spatial data infrastructures — Part 1: Reference model. Technical Report. |
| 08 | Commission of the European Communities (CEC), 2009. Communication COM(2009) 223 - GMES and its initial operations (2011 - 2013). |
| 09 | Commission of the European Communities (CEC), 2008. Communication COM(2008)0046 - Towards a Shared Environmental Information System (SEIS). |
| 10 | Commission of the European Communities (CEC), 2010. Communication COM(2010)743 - The European eGovernment Action Plan 2011-2015: Harnessing ICT to promote smart, sustainable & innovative Government. |
| 11 | The European Parliament and the Council, 2003. Directive 2003/98/EC of the European Parliament and of the Council of 17 November 2003 on the re-use of public sector information (PSI Directive). |
| 12 | Interoperable Solutions for European Public Administrations (isa), 2010. European Interoperability Strategy (EIS) for European public services. |
| 13 | International Organization for Standardization (ISO), 2005. ISO 19119 Geographic information – Services. ISO Standard. |
| 14 | Group on Earth Observations (GEO), 2008. The Global Earth Observation System of Systems (GEOSS) 10-Year Implementation Plan. |
| 15 | Usländer, T.; Jacques, P.; Simonis, I.; Watson, K. (2010). Designing environmental software applications based upon an open sensor service architecture. *Environmental Modeling and Software* 2010, *25*, pp. 977–987. |
| 16 | Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009). Service Analysis - A Critical Assessment of the State of the Art". European Conference of Information Systems (ECIS 2009), Verona, Italy.. |


**Table 8-1.** References

# 9 Annexes

## 9.1 Annex A: List of Generic Use Cases

*Automatically generated from the UC server (theme WP4)*

## 9.2 Annex B: List of Requirements for "Generic Enablers"

*Automatically generated from the UC server*