CEN/TC 287

Date: 2011-03

CEN/TR –15449

WI 00287030

CEN/TC 287

Secretariat: BSi

Geographic information — Spatial Data Infrastructures — Part 4 : Service Centric View

Geoinformation —Geodateninfrastrukturen

Information géographique —Infrastructures d'information spatiale

ICS: 07.040; 35.240

Descriptors:

Document type: Technical Report
Document subtype:
Document stage: Working Document
Document language: E

C:\PRODUCTI\00287\030\51_e_stw.doc STD Version 2.1c

Contents

Page

1 SCOPE 7

2 REFERENCES 8

3 TERMS AND DEFINITIONS 9

4 ABBREVIATED TERMS 12

5 SERVICE-CENTRIC VIEW ON SDI 14

5.1 INTRODUCTION 14
5.2 USE OF RM-ODP VIEWPOINTS 15
5.3 SERVICE MODELS, PROCESSES AND SERVICE ORIENTED ARCHITECTURES 17
5.4 THE MODEL-DRIVEN APPROACH 18
5.5 SYSTEM-OF-SYSTEMS ENGINEERING 18

6 ENTERPRISE VIEWPOINT 20

6.1 OVERVIEW 20
6.2 RELEVANT STANDARDS 21
6.3 EXAMPLE AND TOOLS 21

7 COMPUTATIONAL VIEWPOINT 22

7.1 OVERVIEW 22
7.2 RELEVANT STANDARDS 22
7.3 EXAMPLES AND TOOLS 23
7.3.1 Service Modeling with SoaML 23
7.3.2 Lifecycle Service Components 23
7.3.3 Value Chain of SDI Knowledge Generation 24
7.3.4 Overview of Stakeholders 25
7.3.5 Requirements for a next generation of SDI Services 26
7.4 ARCHITECTURE-BASED ENABLER CLASSIFICATION 27
7.4.1 Architectural Classification according to ISO 19119 27

8 INFORMATION VIEWPOINT 28

8.1 RELEVANT STANDARDS 31
8.2 EXAMPLES AND TOOLS 31

9 ENGINEERING VIEWPOINT 33

9.1 OVERVIEW 33
9.2 MULTI-STYLE SOA 33
9.3 RELEVANT STANDARDS 35
9.3.1 Service-Oriented-Architectures (SOA) 35
9.3.2 Representational State Transfer (REST) 36
9.3.3 Web 2.0 36

10 TECHNOLOGY VIEWPOINT 37

10.1 OVERVIEW 37
10.2 ARCHITECTURAL CLASSIFICATION ACCORDING TO CLOUD COMPUTING SERVICE CATEGORIES 38

11 SERVICES AND SECURITY 39

11.1 SERVICES - SECURITY AND RIGHTS MANAGEMENT 39
11.1.1 Managing access and re-use rights *39*
11.1.2 Key drivers behind effective rights management *40*
11.1.3 Challenges of implementing rights management *40*
11.1.4 Securing OGC Web Services *40*
11.1.5 Security Requirements *41*

ANNEX A RM-ODP VIEWPOINTS AND THE PARTS OF TR 15449 44

ANNEX A EXAMPLE – USE CASE BASED METHODOLOGY 45

11.1.6 Example of the application of a methodology based on use cases *45*
11.1.7 Use Case and Requirements Modelling *46*

ANNEX B EXAMPLE – USE CASE TEMPLATE 49

ANNEX C SERVICE MODELING - SOAML 52

Foreword

This document has been prepared by Technical Committee CEN/TC 287 “Geographic information”, the secretariat of which is held by BSi.

Introduction

Spatial data infrastructure (SDI) is a general term for the computerised environment for handling data that relates to a position on or near the surface of the earth. It may be defined in a range of ways, in different circumstances, from the local up to the global level.

This Technical Report focuses on the technical aspects of SDIs, thereby limiting the term SDI to mean an implementation neutral technological infrastructure for geospatial data and services, based upon standards and specifications. It does not consider an SDI as a carefully designed and dedicated information system; rather, it is viewed as a collaborative framework of disparate information systems that contain resources that stakeholders desire to share. The common denominator of SDI resources, which can be data or services, is their spatial nature. It is understood that the framework is in constant evolution, and that therefore the requirements for standards and specifications supporting SDI implementations evolve continuously.

SDIs are becoming more and more linked and integrated with systems developed in the context of e-Government. Important drivers for this evolution are the Digital Agenda for Europe, and related policies (see part 1). By sharing emerging requirements at an early stage with the standardization bodies, users of SDIs can help influence the revision of existing or the conception of new standards. A number of useful recommendations are made within the Eye on Earth White Paper[1] which provide additional context and background to the service centric view as described here.

The users of an SDI are considered to be those individuals or organisations that, in the context of their business processes, need to share and access geo-resources in a meaningful and sustainable way. Based on platform- and vendor-neutral standards and specifications, an SDI aims at assisting organisations and individuals in publishing, finding, delivering, and eventually, using geographic information and services over the internet across borders of information communities in a more cost-effective manner.

Existing material about SDIs abounds. The criteria used for determining if a given standard or specification is referred to in this report are that the publication addresses an aspect of SDI, and that it is non-proprietary in nature.

Based on these considerations, the following reports have been taken into account:

- legal texts and guidelines produced in the context of INSPIRE;
- documents produced by ISO/TC 211 (and co-published by CEN);
- documents produced by the Open Geospatial Consortium (OGC), including the OpenGIS Reference Model (ORM) (OGC, 2003);
- the European Interoperability Framework and related documents;
- deliverables from the European Union-funded projects (e.g. ORCHESTRA, GIGAS, SANY, ENVISION, ENVIROFI, EO2HEAVEN)[2].

Considering the complexity of the subject and the need to capture and formalize different conceptual and modelling views, TR15449 is comprised of multiple parts:

- Part 1: Reference model: this provides a general context model for the other Parts, applying general IT architecture standards;

[1] Recommendations for the Technical Design of a Global Interoperable Information Network - http://www.eyeonearthsummit.org/sites/default/files/WG3_WP1_Formatted_Dec4_Final check_.pdf

[2] A definitive list of EU Funded projects is given in Part 2 and further reference to these projects is outlined in Annex A.

- Part 2: Best Practice: This provides best practices guidance for implementing SDI, through the evaluation of the projects in the frame of the European Union funding programmes.
- Part 3: Data centric view: This addresses concerns related to the data, which includes application schemas and metadata.
- Part 4: Service centric view: This Part.

Further parts may be created in the future.

1 Scope

This Part of TR 15449 describes a service-centric view of a Spatial Data Infrastructure (SDI).

The Service Centric view addresses the concepts of service specifications, the methodology for developing service specifications through the application of the relevant International Standards, and the content of such service specifications described from the perspective of the five Reference Model of Open Distributed Processing (RM-ODP) viewpoints. The enterprise viewpoint addresses service aspects from an organisational, business and user perspective. The computational viewpoint addresses service aspects from a system architect perspective. The information viewpoint addresses service aspects from a geospatial information expert perspective. The engineering viewpoint addresses service aspects from a system designer perspective. The technology viewpoint addresses service aspects from a system builder and implementer perspective.

The intended readership of this Technical Report are those people who are responsible for creating frameworks for SDI, experts contributing to INSPIRE, experts in information and communication technologies and e-government that need to familiarize themselves with geographic information and SDI concepts, and standards developers and writers.

2 References

This Technical Report contains no normative references. Informative references are given in the annexes and bibliography. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10181 Information technology - Open Systems Interconnection - Security frameworks for open systems

EN ISO 19119 Geographic information – Services

ISO/IEC 10746-1:1998 Open Distributed Processing -- Reference model: Overview

ISO/IEC 10746-2:2009 Open Distributed Processing -- Reference model: Foundations

ISO/IEC 10746-3:2009 Open Distributed Processing -- Reference Model: Architecture

ISO/IEC 10746-4:1998 Open Distributed Processing -- Reference Model: Architectural semantics

ISO/IEC 19793: Information technology - Open distributed processing - Use of UML for ODP system specifications

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
architecture
fundamental organization of a system embodied in its components, their relationship to each other and the environment, and the principles guiding its design and evolution

[IEEE 1471-2000].

3.2
architectural style
coordinated set of architectural constraints that restricts the roles/characteristics of architectural elements and the allowed relationships among those elements within an architecture that conforms to that style. derived from [Fielding, 2000][3]

3.3
conceptual formalism
set of modelling concepts used to describe a conceptual model

[EN ISO 19101:2005]

EXAMPLE UML meta model, EXPRESS meta model.

NOTE One conceptual formalism can be expressed in several conceptual schema languages.

3.4
conceptual model
model that defines concepts of a universe of discourse

[EN ISO 19101:2005]

3.5
conceptual schema
formal description of a conceptual model

[EN ISO 19101:2005]

3.6
conceptual schema language
formal language based on a conceptual formalism for the purpose of representing conceptual schemas

[EN ISO 19101:2005]

EXAMPLE UML, EXPRESS, IDEF1X.

NOTE A conceptual schema language may be lexical or graphical. Several conceptual schema languages can be based on the same conceptual formalism.

3.7
conformance
fulfilment of specified requirements

---

[3] 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

[EN ISO 19113:2005]

3.8
component
physical, replaceable part of a system that packages implementation and provides the realization of a set of interfaces

[ISO/TS 19103:2005]

3.9
identifier
linguistically independent sequence of characters capable of uniquely and permanently identifying that with which it is associated

[ISO/IEC 11179-3:2003]

3.x
interface
Named set of operations that characterize the behaviour of an entity.
(ISO 19119:2005).

3.10
interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units

[ISO/IEC 2382-1:1993]

3.11
multi-style service-oriented architecture[4]
Service-oriented Architecture in which the service-oriented architectural style coexists with other architectural styles. A service-oriented architectural style is an architectural style that restricts the roles, characteristics and allowed relationships of services and service consumers.

3.12
reference frame
aggregation of the data needed by different components of an information system

3.13
resource
asset or means that fulfils a requirement

[EN ISO 19115:2005]

3.14
Service

Distinct part of the functionality that is provided by an entity through interfaces (ISO 19119:2005).

Service is defined as the delivery of value to another party, enabled by one or more capabilities. [OMG SoaML].

---

[4] See 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. http://digbib.ubka.uni-karlsruhe.de/volltexte/1000016721

3.14
spatial data infrastructure (SDI)
metadata, spatial data sets and spatial data services; network services and technologies; agreements on sharing, access and use; coordination and monitoring mechanisms, processes and procedures, established, operated or made available in an interoperable manner

NOTE In the context of this report the term SDI is restricted to a platform- and implementation-neutral technological infrastructure for geospatial data and services, based upon standards and specifications.

3.15
system of systems
large-scale integrated systems that are heterogeneous and consist of sub-systems that are independently operable, but are networked together for a common goal

3.16
use case
specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system

[OMG UML Specification]

4 Abbreviated terms

API Application Programming Interface

AJAX Asynchronous JavaScript and XML

BPMN Business Process Model and Notation

BMM Business Motivation Metamodel

BSi British Standards Institute

CEN European Committee for Standardization or Comité Européen de Normalisation

EIF European Interoperability Framework

EN European Standard (CEN deliverable)

INSPIRE Infrastructure for Spatial Information in Europe

ISA Interoperability Solutions for European Public Administrations

ISO/TC 211 Geographic information/Geomatics

GIGAS GEOSS , INSPIRE and GMES an Action in Support

GEOSS Global Earth Observation System of Systems

GMES Global Monitoring for Environment and Security

GML Geography Markup Language

HTTP Hyper-text Transfer Protocol

IaaS infrastructure as a service

ISO International Organization for Standardization

IT Information Technology

JSON Javascript Object Notation

OASIS Organization for the Advancement of Structured Information Standards

ORCHESTRA Open Architecture and Spatial Data Infrastructure for Risk Management

ODP Open Distributed Processing

OGC Open Geospatial Consortium

OMG Object Management Group

ORM OpenGIS Reference Model

PaaS Platform as a Service

REST Representational State Transfer

RM-ODP	Reference Model of Open Distributed Processing
RPC	Remote procedure call
SANY	Sensors Anywhere
SaaS	Software as a Service
SDI	Spatial Data Infrastructure
SEIS	Shared Environmental Information System
SOA	Service Oriented Architecture
SoaML	Service oriented architecture Modeling Language
SOAP	Simple Object Access Protocol[5]
SoS	System of systems
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
USDL	Universal Service Description Language
UUID	Universally Unique Identifier
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WRS	Web Registry Server
WSA	Web Service Architecture
WSDL	Web Service Description Language
W3C	World Wide Web Consortium
WWW	World Wide Web
XHTML	eXtensible HyperText Markup Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

[5] Original meaning of the acronym SOAP, however, its use has been deprecated

5 Service-centric view on SDI

5.1 Introduction

This Part of TR 15449 addresses the concepts of service specifications, the methodology for developing service specifications through the application of the relevant International Standards. It is based on a system architecture approach to system design known as the Reference Model of Open Distributed Processing[6] (RM-ODP). Architecture is a set of components, connections and topologies defined through a series of viewpoints. The spatial data infrastructure of interest for this report will have multiple users, developers, operators and reviewers. Each group will view the system from their own perspective. The purpose of architecture is to provide a description of the system from multiple viewpoints. Furthermore, architecture helps to ensure that each viewpoint will be consistent with the requirements and with the other viewpoints.

According to RM-ODP, the content of such service specifications is described from the perspective of the five RM-ODP viewpoints:

- **The enterprise viewpoint** addresses service aspects from an organisational, business and user perspective.
- **The computational viewpoint** addresses service aspect from a system architect perspective.
- **The information viewpoint** addresses service aspects from a geospatial information expert perspective.
- **The engineering viewpoint** addresses service aspects from a system designer perspective.
- **The technology viewpoint** addresses service aspects from a system builder and implementer perspective.

This document focuses on platform independent descriptions of services, from the viewpoints of different stakeholders and concerns. References are given to relevant standards that can be used further for this, both for platform independent and platform dependent descriptions related to services. These concepts are illustrated in Figure 1.

6 see ISO/IEC 10746

Figure 1 This document covers 'what' might be needed and points existing standards for 'how' it might be implemented

5.2 Use of RM-ODP viewpoints

This Part of TR 15449 is following the approach of using ISO RM-ODP for description of distributed systems. Figure 2[7], - illustrates the main focus of each of the five RM-ODP viewpoints, and the main question that each viewpoint addresses.

[7] This is adapted from "GIGAS Methodology for comparative analysis of information and data management systems" OGC Best Practice, 10-028r1

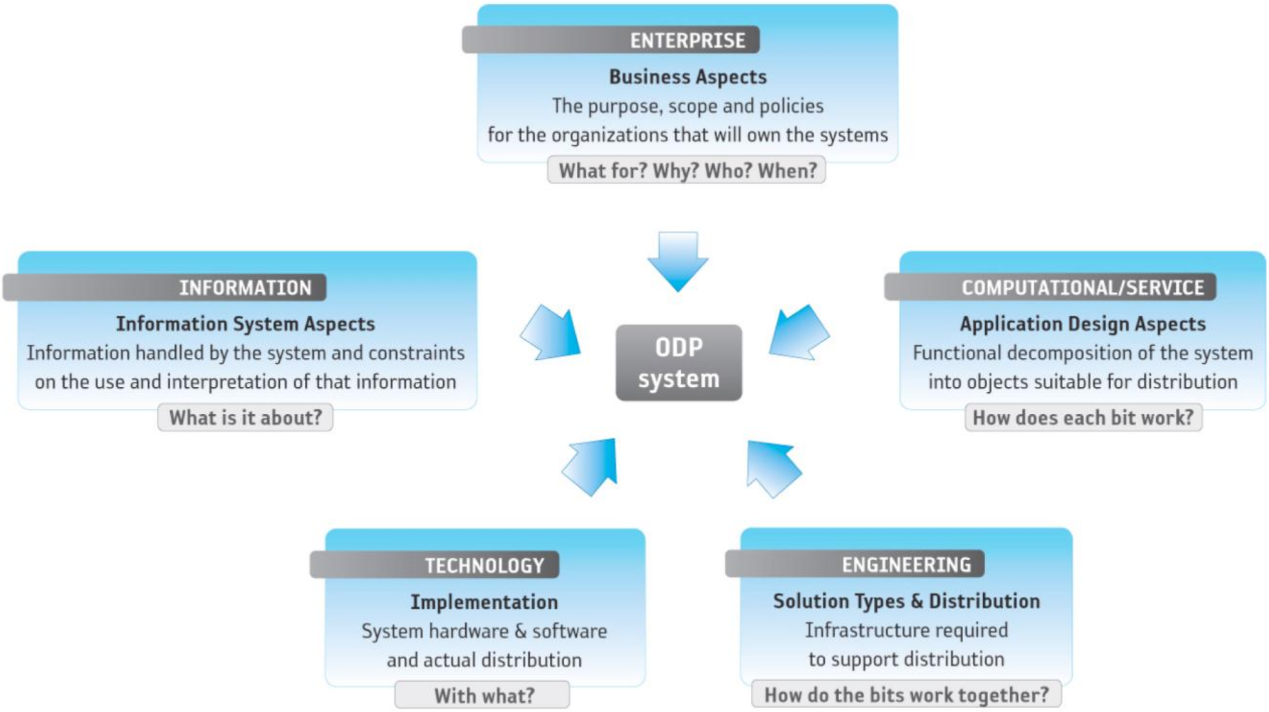


Figure 2 ISO RM-ODP viewpoints

The enterprise viewpoint is concerned with the purpose, scope and policies of an enterprise or business and how they relate to the specified system or service. An enterprise specification of a service is a model of that service and the environment with which the service interacts. It covers the role of the service in the business and the human-user roles and business policies related to the service. *In the context of the service centric view there is a particular focus on the use cases and external functionally related to the particular services.*

The computational viewpoint is concerned with the interaction patterns between the components (services) of the system, described through their interfaces. A computational specification of a service is a model of the service interface seen from a client, and the potential set of other services that this service requires to have available, with the interacting services described as sources and sinks of information.

The information viewpoint is concerned with the semantics of information and information processing. An information specification of an ODP system is a model of the information that it holds and of the information processing that it carries out. *In the context of the service centric view there is a particular focus on the information being used and provided by the particular services.*

The engineering viewpoint is concerned with the design of distribution-oriented aspects, i.e., the infrastructure required to support distribution. An engineering specification of an ODP system defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it defines. ODP defines the following distribution transparencies: access, failure, location, migration, relocation, replication, persistence and transaction. Security may also be a mechanism.

The technology viewpoint describes the implementation of the ODP system in terms of a configuration of technology objects representing the hardware and software components of the implementation. It is constrained by cost and availability of technology objects (hardware and software products) that would satisfy this specification. These may conform to platform-specific standards that are effectively templates for technology objects.

There are, however, important relationships between the viewpoints. In particular for the service-centric view on SDIs, it is important to see the relationship to the use of services from the enterprise viewpoint, the information being provided as input and output to services from the information viewpoint, the logical service architecture itself from the computation viewpoint, the different mechanisms and architectural patterns used for distribution and different architectural styles around services in the engineering viewpoint, and the actual technologies being used in the technology viewpoint. For this reason the structure of the whole of this Part is created according to the various ODP viewpoints – and describe the relevant service-centric aspects within each of the viewpoints.

The different Parts of TR 15449 have an emphasis anchored in one of the RM-ODP viewpoints. These are illustrated in Annex A. Figure 3 shows how the following clauses are dedicated to describing the service-centric for SDIs from the various ODP viewpoints

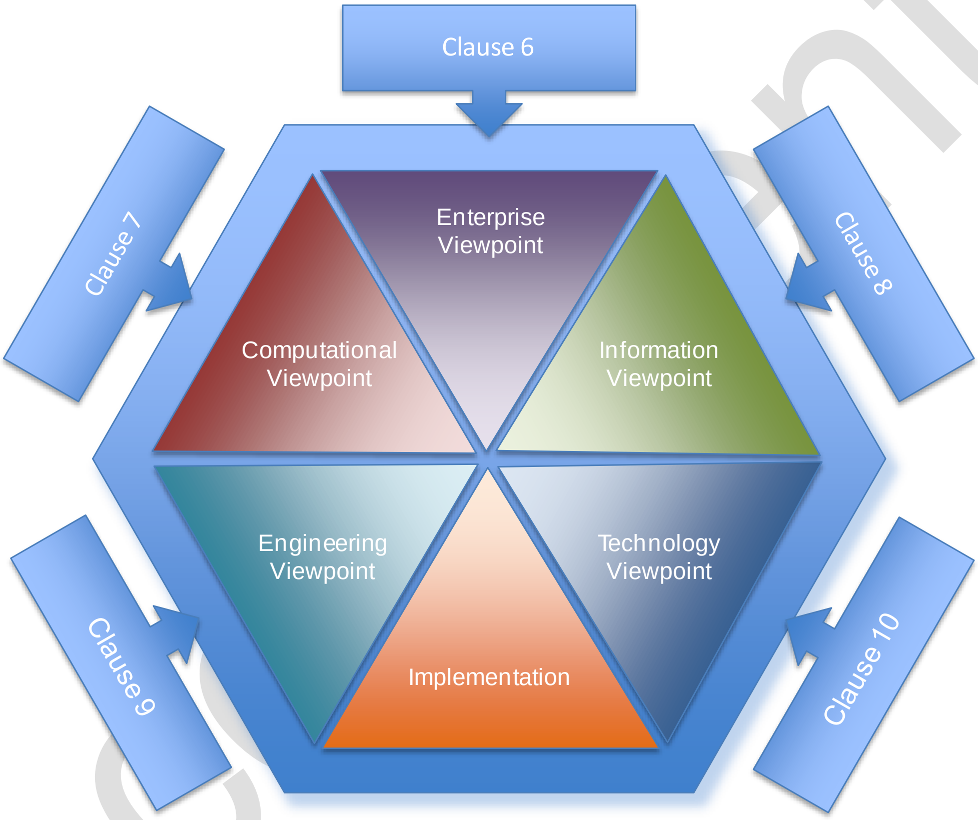


Figure 3 Clauses and corresponding described RM-ODP viewpoints

5.3 Service models, processes and service oriented architectures

The spatial data in an SDI are a model of the real world. On top of the data, and by making use of the metadata, services can be built to make the data accessible through the web and to use them in any information system by e.g. viewing, downloading, or processing them. A Service Oriented Architecture (SOA) enables new and existing enterprise systems to share services, information and data across technical platforms, departments and ultimately across organizational and regional boundaries. In current systems services are also used as an integration approach for multiple architectural styles, including both document styles and synchronous RPC (Remote procedure call) styles of services, and also include integration with event management and an event driven architectural styles as well as support for interaction with sensors through services. Process support as with process management and workflow systems, including service composition, orchestration and choreography, can be viewed as the sequencing of behaviour in the implementation of services.

5.4 The model-driven approach

A model driven approach has been established in both ISO/TC211 and OGC in particular for the data-centric view, but is now also emerging from services. Recent work in international standards such as SoaML (from OMG) and current activities such as USDL (from W3C), provides new facilities for the modeling and specification also of services in a platform neutral way.

The further description of approaches for service modeling is further described in the sections about SoaML and USDL under the computational viewpoint.

5.5 System-of-Systems Engineering

The notion of “System of Systems” (SoS) and “System of Systems Engineering” (SoSE) emerged in many fields of applications [], to address the common problem of integrating many independent, autonomous systems, frequently of large dimensions, in order to satisfy a global goal while keeping them autonomous.

In spite of a large scale integrated system, the SoS components can operate independently to produce products or services satisfying their customer objectives. The component systems may be connected by implementing one or more interoperability arrangements that do not require tight coupling or strong integrations. In keeping with the definition, SoS key concepts are:

- *large-scale systems*: in a SoS the subjection of subsystems to a central task often introduces new challenging problems that for small systems may reduce the advantage. Therefore for small systems other solutions may be more effective and efficient.
- *heterogeneous*: homogeneous systems may be merged in an integrated system without the need of SoS engineering tasks.
- *independently operable*: the realization of a SoS must not affect the normal and usual working of the composing systems. The SoS engineering implements agreements supplementing without supplanting the existing.
- *networked together*: the composing systems need to inter communicate to achieve the common goal.

These concepts help to understand when a SoS approach is a valuable solution. This allows a SoS to maintain its inherent operational character even as system components join or disengage from it. Since SoS is a construct of both legacy and new systems, an important feature is the attention to flexibility and holistic aspects.

SoS engineering deals with planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into an SoS capability greater than the sum of the capabilities of the constituent parts.

In a SoS Engineering process, interoperability is a fundamental requirement. In order to achieve it, such a SoS must effectively deal with various issues. The information crosses trust boundaries, where each system is controlled and managed independently, and involves social, political and business considerations. By an architectural point-of-view, the quantitative and qualitative differences in the data exchange across the disparate systems must be dealt with. For example, the architecture typically involves different technology stacks, design models and component life cycles. In a SoS, the systems under consideration are loosely coupled, i.e. minimal assumptions can be made about the interface between two interacting systems. Thus, when dealing with loosely coupled systems, a system’s interface can be described in terms of the data model and the role (producer or consumer) the system plays in the information exchange.

Standardisation as a means of achieving interoperability, is a key activity in a SoS engineering process. Due to the desire for adaptability in all areas of system of systems, these standards should, at a minimum, be developed as standards that are “open” to any entity participating or impacted by the SoS. Adaptability is necessary in a system of systems, since the membership or configuration is or can be dynamic, and the relationships among all of the systems in the SoS may not always be known.

6 Enterprise viewpoint

6.1 Overview

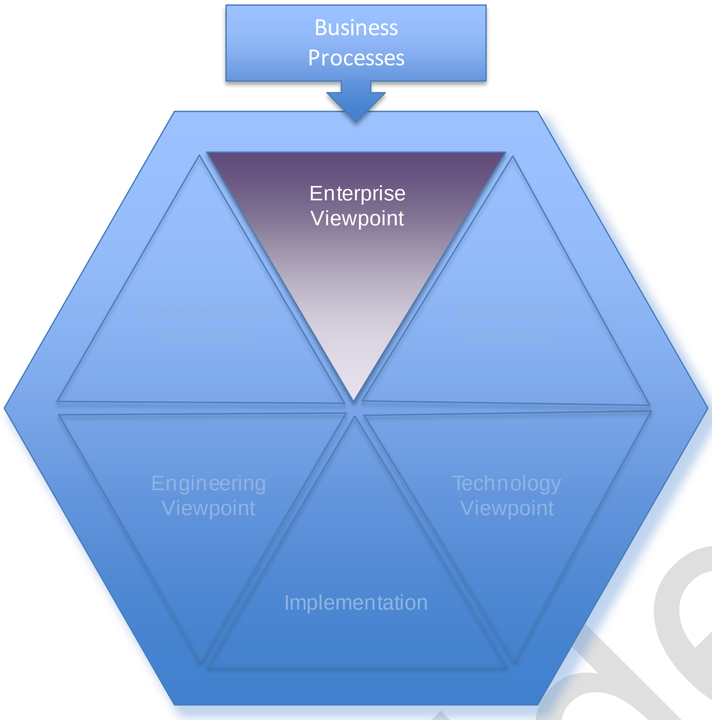


Figure 5. The Enterprise viewpoint

The Enterprise viewpoint (Figure 5) describes the context for a system and a set of services. It concentrates on the objectives, business rules and policies that need to be supported by systems and services. An enterprise specification of a service is a model of that service and the environment with which the service interacts. It covers the role of the service in the business and the human-user roles and business policies related to the service. *In the context of the service centric view there is a particular focus on the use cases and external functionally related to the particular services.*

Experiences with the development of SDIs has shown that it is very useful to have models for the enterprise viewpoint, focusing on generic descriptions of the usage, and typical process for usage, as this helps to shape the understanding of the functionality and constraints that are placed on systems and services. It also helps concrete project and development activities to place their needs in the context of both existing and available standards and services, as well as supporting the identification of new services.

Spatial Data Infrastructures (SDI) aim at supporting a multitude of users and organizations to support the variety of work processes they are involved in. In order to reach their goals of enhanced spatial data sharing, they should cover the business requirements of as many organizations and processes as possible.

A work process is defined as the way in which organizations create products, services or policies. It is a succession of structured and interconnected activities across time and space which, starting from an identifiable input, result in a defined output in the form of a product or service. In order to obtain the desired output the input should be transformed. Ideally, the transformation that occurs in the process should add value to the input and create an output that is more useful to the recipient either upstream or downstream. Traditionally work processes occurred within single organizations, but more and more they cross organizational and even country boundaries. Often a process is divided in several sub-processes due to complexity, which can in turn be sub-divided in a series of activities and tasks. Therefore, the simple input-throughput-output model will rather consist of several interconnected input-throughput-output chains whereby the output of one sub-process serves as the input for another sub-process.

Many processes create products based on other products, which is e.g. the case for manufacturers of cars. For work processes dealing with policy preparation, monitoring and evaluation, decision making, or service provision, the notion of data and information flows is crucial. Data and information are needed as input, in order to process them and to create new data and information that can be used to take decisions, to serve other organisations, policy makers or even individual citizens.

In the context of SDI-supported work processes, the focus is on the flow of spatial data. The explicit aim of an SDI is to enhance these flows, i.e. to make the exchange of spatial data and information between the different stakeholders more efficient and effective. In most decision making processes spatial data play an important role. An example of such a process, in which many public authorities, private companies as well as citizens are involved, is the water management & flood mapping process.

6.2 Relevant standards

BPMN – Business Process Model and Notation – this is a standard from OMG that is used for the purpose of both enterprise and business oriented modeling, as well as for technology mapping to process execution with technologies such as BPEL.

Use cases - UML, Use cases with use case templates[8], has, been the most used form for documentation of typical user needs for system and services functionality, within the SDI community. The UML standard provides only a graphical form for diagramming of use cases, while the SDI community typically has adopted various forms of use case templates for the technical documentation of use cases.

Agile requirements engineering with user stories – The software engineering community has recently evolved a set of approaches around agile methods (including SCRUM, KANBAN) that focuses on close interactions between potential system users and developers, and focuses on more light weight user stories as input to a system development backlog.

BMM – Business Motivation Metamodel – this is a standard from OMG that provides a foundation for modeling of vision, goals and objectives for an enterprise – with mappings to tactics for solutions including use of processes and services.

UML4ODP Enterprise specification profile (ISO/IEC 19793) - this ISO standard provides a UML profile for all of the main concepts defined in the RM-ODP enterprise viewpoints. It is a good reference and foundation for doing full RM-ODP enterprise viewpoint modeling, but in the SDI community it has so far been a preference to use a more light weight approach with BPMN and/or Use cases.

6.3 Example and tools

Annex A is showing an example of a use case based methodology.

[8] Cockburn, A. Writing Effective Use Cases. ISBN-13: 9780201702255. Addison-Wesley (2001)

7 Computational viewpoint

7.1 Overview

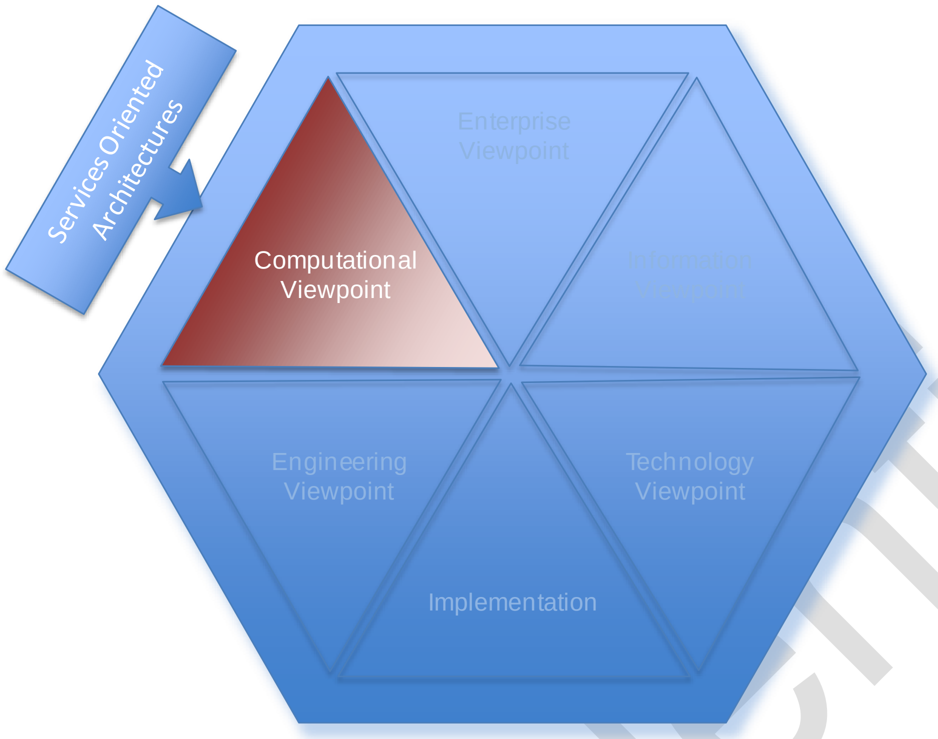


Figure 8. The computational viewpoint

The computational viewpoint, illustrated in Figure 8, is concerned with the interaction patterns between the components (services) of the system, described through their interfaces[9]. A computational specification of a service is a model of the service interface seen from a client, and the potential set of other services that this service requires to have available, with the interacting services described as sources and sinks of information. In the context of multi style SOA, the service specification might also include signals (events) that are generated or received, and support both synchronous and asynchronous interactions and both RPC oriented and document-oriented/RESTful styles of interaction. The computational viewpoint is the core viewpoint for the identification of interfaces and services.

7.2 Relevant standards

SoaML – Service oriented architecture Modeling Language (SoaML)[10]– is a new standard from OMG that provides a UML profile and a metamodel for the modeling of services. It was adopted by OMG in 2010 and the official final version of the standard, after the work of the Finalisation Task Force, was published in March 2012. SoaML has already since 2011 been supported by all major UML tool vendors.

[9] See Eye on Earth White Paper – “Recommendation 9: It is recommended that common services be established that implement the mentioned frameworks”.

[10] See http://www.omg.org/spec/SoaML/1.0/

USDL – Universal Service Description Language – The Unified Service Description Language (USDL)[11] is a platform-neutral language for describing services. The Unified Service Description Language is used for describing business, operational and technical parameters of services. Service descriptions then include information like pricing, legal, service provider, interaction methods, service level agreements and so on. This allows for more sophisticated use cases than service descriptions like WSDL allow today, e.g., comparison of services by price. W3C Incubator Group Report 27 October 2011[12] - where the recommendation is that the work of formally specifying USDL also under consideration of existing standards is continued at W3C

UML4ODP Computational specification profile (ISO/IEC 19793) - this ISO standard provides a UML profile for all of the main concepts defined in the RM-ODP computational viewpoint. It is a good reference and foundation for doing full RM-ODP computational viewpoint modeling, but in the SDI community it has so far been a preference to use a more light weight approach initially with UML class diagrams, and recently with investigations into the usage of SoaML and USDL.

7.3 Examples and tools

7.3.1 Service Modeling with SoaML

The Service oriented architecture Modeling Language (SoaML) specification defines a UML profile and a metamodel for the design of services within a service-oriented architecture. The goals of SoaML are to support the activities of service modelling and design and to fit into an overall model-driven development approach, supporting SOA from both a business and an IT perspective.

A further description of SoaML can be found in Annex C.

7.3.2 Lifecycle Service Components

Part 1 of this Technical Report identifies the core components of the SDI Reference model (Figure 10).

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 SDI domains. Likewise, GeoPortals are a specific type of geospatial applications.

[11] See www.internet-of-services.com

[12] See http://www.w3.org/2005/Incubator/usdl/XGR-usdl-20111027/#L16117

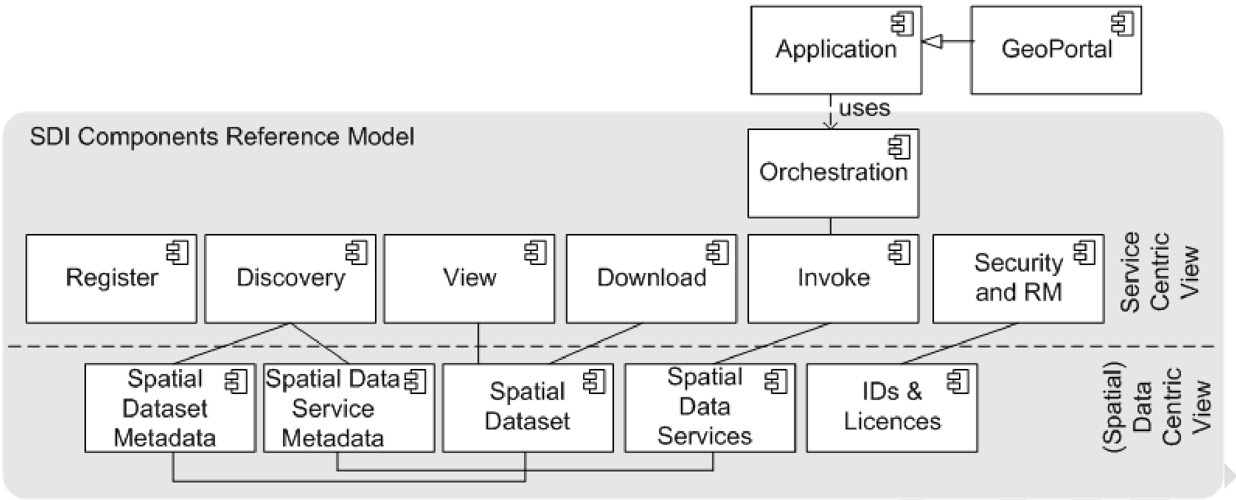


Figure 10. Core Components of the SDI Reference Model.

The primary organizing structure is determined by the following generic core lifecycle components (corresponding to the service centric view in Figure 10):

- **Register (Publish)**: 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.

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 SDI services. In doing so, we provide a bridge between practical SDI applications and the wider political framework. These findings could equally be applied to other geospatial and non-geospatial domains beyond the SDI domain.

7.3.3 Value Chain of SDI Knowledge Generation

When analyzing the requirements of SDI services for the terrestrial, atmospheric and marine sphere, six roles may be identified each of which contributes to the generation of SDI knowledge and are therefore part of the value chain.

- **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 SDI 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 SDI application focusing on a particular sphere, such as terrestrial biodiversity.
- **Decision Maker**, consuming an SDI 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 [13]. Note that workflow services may themselves get published in order to serve as building blocks for more complex SDI solutions.

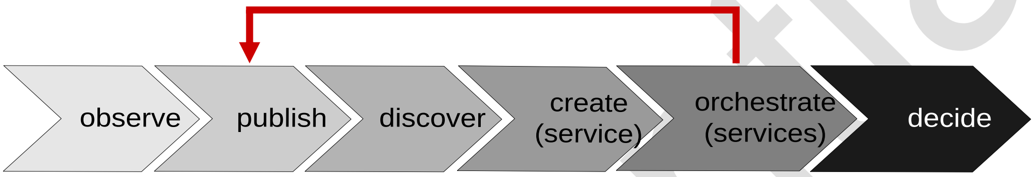


Figure 11. Added value chain of SDI knowledge generation

7.3.4 Overview of Stakeholders

The roles identified above are played by a variety of individuals and organizations:

- **Citizens** of a particular social, political, or national community;
- **SDI 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
SDI agencies	x	x		x		x
Public authorities		x		x		x
Industries			x	x	x	x
Platform providers				x		

[13] S. Schade, B. Fogarty, M. Kobernus, K. Schleidt, P. Gaughan, P. Mazzetti and A. Berre (2011). Environmental Information Systems on the Internet - A Need for Change. In Hrebicek, J., Schimak, G. and Denzer, R. (Eds.) Environmental Software Systems. Frameworks of eEnvironment, Proceedings of the 9th IFIP WG 5.11 International Symposium, ISESS 2011, Brno, Czech Republic, June 27-29, 2011, pp 144-153.

Infrastructure providers				x			
Sensor network owners	x	(x)		(x)			x

Table 1. Added-value chain of SDI knowledge generation

Table 1 provides an overview of the manifold mappings between these stakeholders and the different roles in the value chain of SDI knowledge generation. Notably, citizens can play all roles, they may even discover available information and provide new services (mash-ups).

7.3.5 Requirements for a next generation of SDI Services

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

- publication, discovery, access and visualization of SDI 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 SDI 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, GMES, SEIS, GEOSS,
  - relevant well-established communities, including research and e-government infrastructures , and
  - the mode relevant policies on international level, above all related to Public Sector Information (PSI) .

Dedicated components (SDI 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.

7.4 Architecture-based Enabler Classification

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

7.4.1 Architectural Classification according to ISO 19119

Figure 12 shows how the different types of enablers can be related in the context of a complete end-to-end ICT architecture.

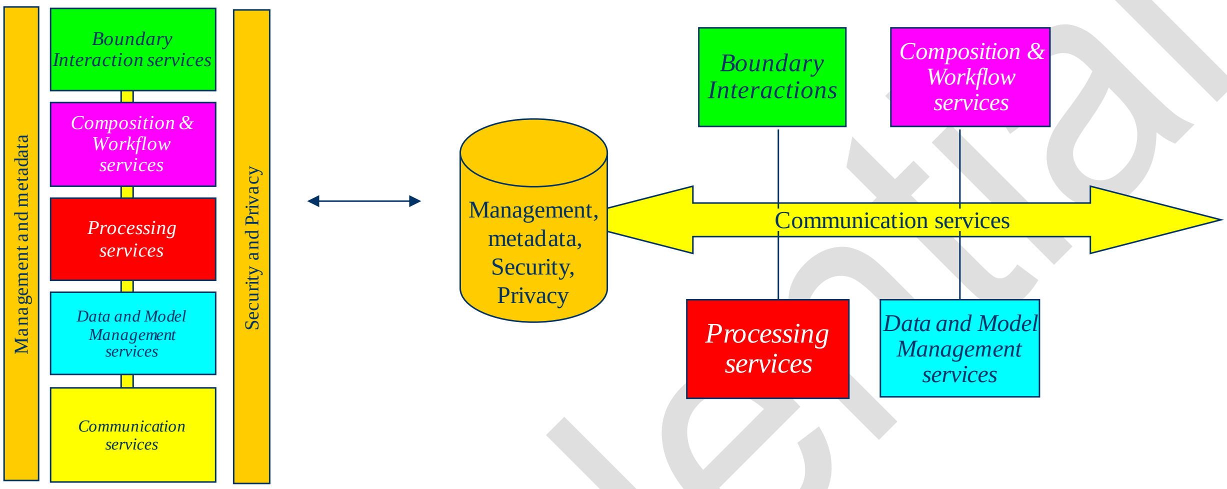


Figure 12. Relationships of enablers in both a layered and a bus architecture

This 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 clause 8.3. It follows the approach to define both generic domain-independent and specific enablers, such as geospatial and SDI specific enablers, in each of the following six groups which are colour-coded in Figure :

The names/types have been generalised from ISO 19119 to be able to support a slightly broader set of services.

- **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 EN 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 SDI information — involving processing of orders for buying and selling of geographic information and services. These services are described in more detail in EN 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. EN 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 EN ISO 19119. These services are based on the content of those standards in the EN ISO 19100 series that standardize the structure of geographic information and the procedures for its administration, including: EN ISO 19107, EN ISO 19108, EN ISO 19109, EN ISO 19110, EN ISO 19111, EN ISO 19112, EN ISO 19113, EN ISO 19114 and EN 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 EN 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. 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.

8 Information viewpoint

The information viewpoint as a separate view on SDI is illustrated in Figure 13.

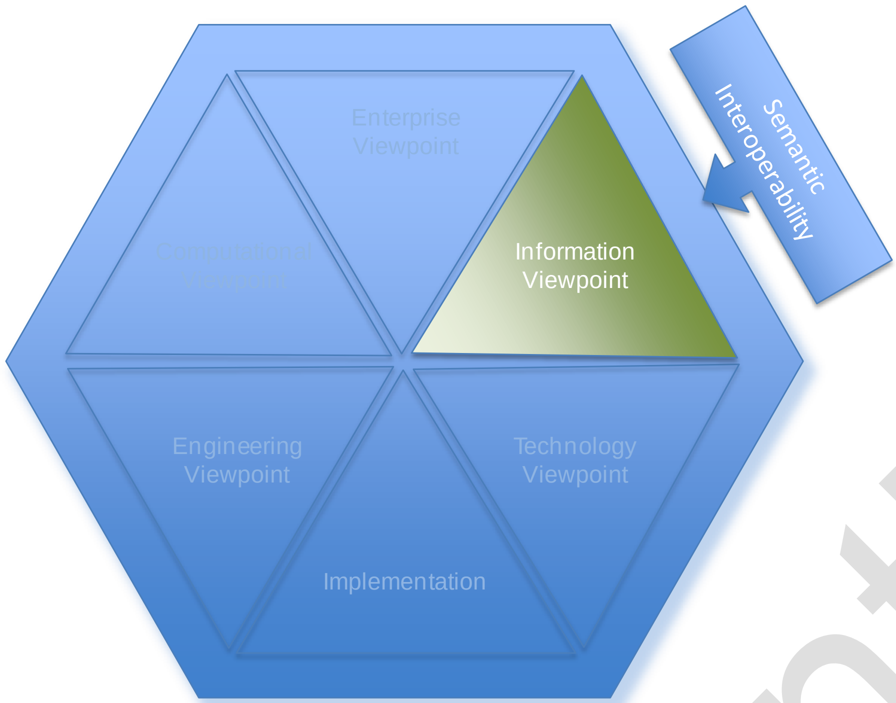


Figure 13. The Information viewpoint.

From the service centric view the information viewpoint is focusing on the information used by services. The information viewpoint is concerned with the semantics of information and information processing. An information specification of an ODP system is a model of the information that it holds and of the information processing that it carries out. *In the context of the service centric view there is a particular focus on the information being used and provided by the particular services.*

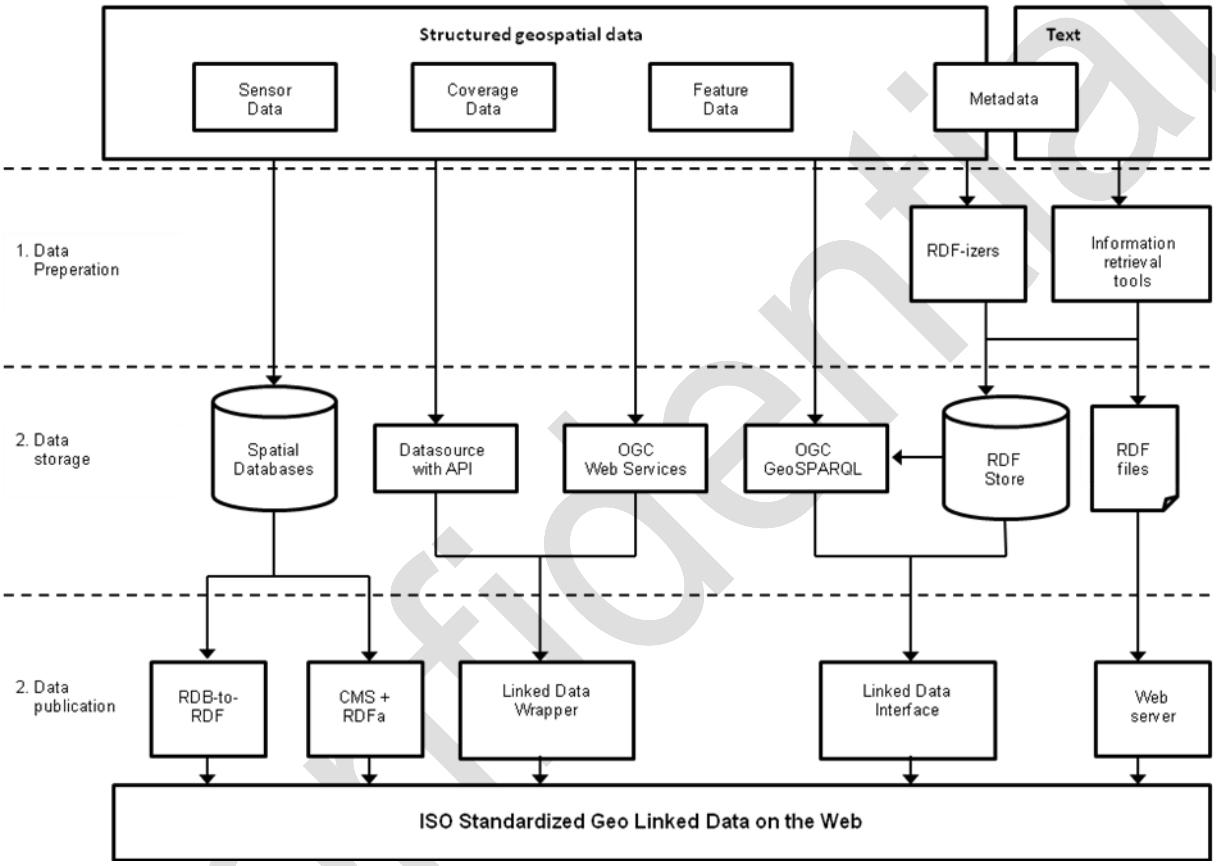
Part 3 of TR 15449 focuses on SDIs descriptions from a Data-centric view. This also provides the foundation for the information viewpoint for services. From the service-centric view the focus within the information viewpoint is on the identification of the information that is consumed and produced by services. Key aspect is the model-driven approach. Specific spatial domains define thematical information areas that are modeled using object oriented technology. The results are application schema that provide conceptual models. The conceptual models are published in data specifications including feature catalogues and these are implemented in data warehouses. Combined with standardized and SOA conforming encoding the result is clearly structured and standardized accessible spatial data.

Recent new developments in the field of disclosing spatial information on the web are in linked data[14]. Linked data is a concept of a Web of Data where data can be found on the web and contain links to other data distributed over the Web. As data are published conforming to the Web architecture, integration of data and multiple use become more prominent and less design dependent. This makes it slightly different to the traditional SOA approach. Traditionally the information structuring in SOA is based on identification of domains in which information and semantics are defined. Linked data bridges the gaps between the thematical domains by providing a mechanism to work without predefined information boundaries. In this regard linked data fits into the web 2.0 philosophy of participatory information sharing where information can be expanded by individuals or communities by interlinking of data.

[14] See Eye on Earth White Paper – "Recommendation 7: It is recommended that resources in environmental information infrastructures be made available following the Linked Data approach and technologies."

Linked data when related to the SOA must be seen as a complementary way to enlarge the possibilities for spatial data integration and spatial to non spatial data integration and broaden its use outside the geospatial community.

Linked data are published in the Resource Description (RDF) format. This is an alternative way of representing information in class diagrams based on expressions known as triples in the form of subject-predicate-object expressions. GML is closely related to RDF and can be transformed if considerations on HTTP URI's and stable links are taken care of. For evaluating the potential and effect of linked data for the geo-information standards community ISO TC 211 has established an adhoc group on linked data. The next figure shows the integration of linked data in the SDI concept. The structured geospatial data serve as data source for linked data interfaces.



Remark: picture from an article named: The Delft Report: Linked Data and the challenges for geographic information standardization. Francisco .J. Lopez-Pellicer, Luis M. Vilches-Blazquez, F.Javier Zarazaga-Soria, Pedro R. Muro-Medrano, O. Corcho. Revista Catalana de Geografía. 2012, vol. XVII, nº 44. ISSN 1988-2459.

The essential pillars of Linked Data (Bizer, 2009)[15] are traditional web technologies and use of light-weight techniques for data model representation. The former depends on the use of Uniform Resource Identifiers (URIs) as reference points. A URI may be used to uniquely identify both data and non-data resources

[15] Bizer, C. (2009). The Emerging Web of Linked Data. IEEE Intelligent Systems, 24(5). pp. 87-92

(Berners-Lee et al, 1998)[16]. Resolvers map a URI to the physical location of the resource, or in the case of non-data resources to a description.

Linked Data is usually implemented as common HTML for human interfaces, plus RDF for links with machine-processable semantics. RDF provides a structure for any form of description, and is the basis of the Semantic Web (Berners-Lee et al, 2001)[17]. RDF describes resources in the form of triples (subject-predicate-object) (Klyne and Carroll, 2004)[18]. A basic typing mechanism for subjects, predicates and objects is available as RDF-Schema (RDF-S). RDF-S allows for extensions in order to specify domain-dependent subtypes, and thus allow for a domain vocabulary in its own namespace. RDF comes with different encodings, one of which RDF/XML. The key elements of RDF/XML for linking are 'rdf:about' (identifiers or anchors) and 'rdf:resource' (pointers or links). Resources become a set in which elements are connected with links. By these means, users can navigate between data like browsing through web pages. Generally, each piece of data contains link(s) to other data. However, leaf nodes or endpoints of the graph may make use of any other format, which may not support linking. Content-negotiation in HTTP allows client applications (like browsers) to negotiate various data representations (Holtman and Mutz, 1998)[19].

Although RDF is recommended for implementing the Linked Data, as a single global model for all data sources, other structured formats can support semantic linking, e.g. GML as introduced by Schade and Cox, 2010)[20].

8.1 Relevant standards

UML, XML, GML – as general languages (graphical and text based) for information modeling.

EN ISO 19103 Geographic information: Conceptual Schema Language – for modeling of information and application schemas using UML

ISO/IEC 19793 – UML4ODP – Information specification profile_- this ISO standard provides a UML profile for all of the main concepts defined in the RM-ODP information viewpoint. It is a good reference and foundation for doing full RM-ODP information viewpoint modeling, but within the SDI community it has so far been a preference to use an approach with UML class diagrams, in combination with the use of XML, GML and potentially semantic technologies like OWL and Linked Open Data with RDF.

8.2 Examples and tools

The INSPIRE program provides an example of establishing a spatial information base in this case to support environmental policy on national and European level. Several technical reports, implementation rules and guidelines for data specifications resulted from this program. Regarding the information viewpoint and the actual production of data specifications a set of documents is available. The presentation of the documents in the order below in itself already is an example of how the information viewpoint in an can be SDI is addressed.

---

[16] Berners-Lee, T., Fielding, R. and Masinter, L. (1998). Uniform Resource Identifiers (URI): Generic Syntax. Internet Engineering Task Force (IETF) Memo – RFC 2396

[17] Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web. Scientific American Magazine, 2001.

[18] Klyne, G. and Carroll, J.J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004

[19] Holtman, K. and Mutz, A. (1998). Transparent Content Negotiation in HTTP. Internet Engineering Task Force (IETF) Memo – RFC 2295

[20] Schade, S. and Cox, S. (2010). Linked data in SDI or how GML is not about trees. In 13th AGILE International Conference on Geographic Information Science (AGILE 2010). http://agile2010.dsi.uminho.pt/pen/ShortPapers_PDF/73_DOC.pdf

- *Definition of Annex Themes and Scope*.
  Identification and description of 34 spatial data domains regarding their definition and scope. Examples of domains are hydrography, transport networks, administrative units, geology.
- *Methodology for the development of data specifications*.
  The process and proposed methodology of developing application schema and feature catalogues is explained. Use cases lead to identification of information requirements. These are transferred to initial spatial objecttypes and subsequent application schema's. An iterative process to test validate and restructure is described to arrive at specifications that fulfil described requirements.
- *INSPIRE Generic Conceptual Model.*
  Basic rules and principles are laid down to which all application schema have to comply. This document bridges the gap between the conceptual standards on geo-information modelling and its application in specific domain models. For instance reusable patterns are introduced on modelling unique identifiers, temporal models and a meta model is presented by defining dedicated stereotypes.
- *Guidelines for the Encoding of Spatial Data.*
  Application schema are implemented in GML application schema. Guidelines are presented to guide this implementation in a harmonized way and provide implementers with additional specifications on top of general GML standards.
- *INSPIRE Data Specification on .... – Guidelines.*
  The actual data specifications divided into 34 documents dealing each dealing with a separate information domain. Each including domain definition, use case description, application schema, feature catalogue and portrayal.

9 Engineering viewpoint

9.1 Overview

The engineering viewpoint is illustrated in Figure 14.

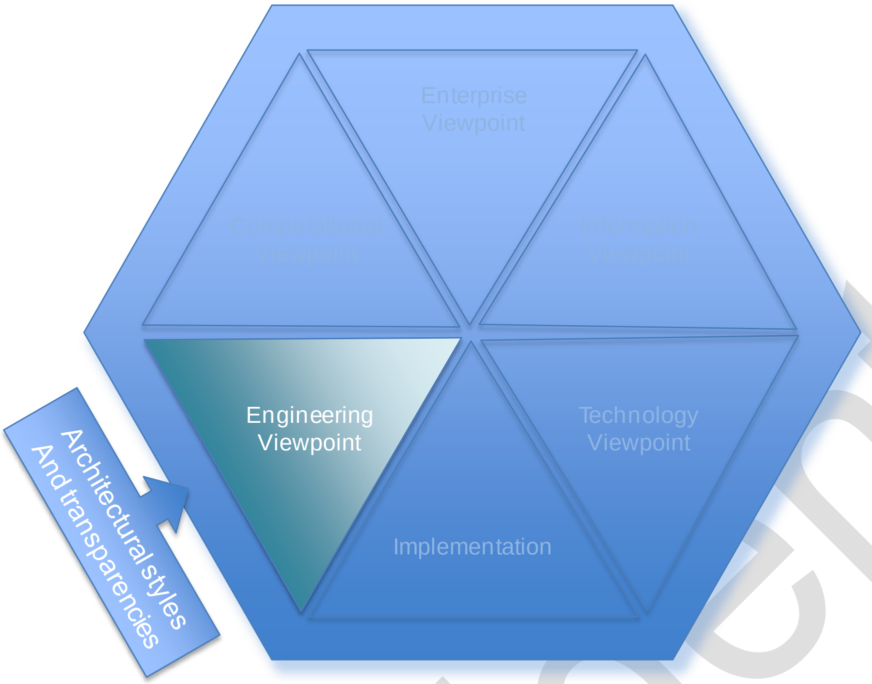


Figure 14. The engineering viewpoint

The engineering viewpoint is concerned with the design of distribution-oriented aspects, i.e. the infrastructure required to support distribution. An engineering specification of an ODP system defines a networked computing infrastructure that supports the system structure defined in the computational specification and provides the distribution transparencies that it defines. ODP defines the following distribution transparencies: access, failure, location, migration, relocation, replication, persistence and transaction. Security may also be a mechanism. In the context of this Part of TR 15449 there is a particular focus on the different mechanisms to support a multi style SOA in a unified logical model, including both RESTful and Synchronous services, as well as events and sensors.

9.2 Multi-style SOA

Geospatial, SDI, Earth and Space Science Informatics (ESSI) are recent disciplines aiming to provide scientists with advanced information and computational services in support of Earth and Space science, including SDI and related multi-disciplinary research[21]. In such context, geospatial services play an important role enabling geo-information sharing; this is essential to provide scientists with services for data discovery, publishing and access. The geospatial services are the building blocks for more complex disciplinary services, e.g. processing, simulation, etc.

---

[21] S. Schade, P. Mazzetti, Z. Sabeur, D. Havlik, T. Usländer, A. Berre, and J. Mon (2011). Towards a Multi-Style Service-Oriented Architecture for Earth Observations. EGU General Assembly 2011, Vienna, Austria.

Most initiatives for the specification and standardization of geo-information resources (e.g. services, models, and formats) adopted Web technologies as their protocols and encodings.

However, in the recent years the WWW has undergone important changes. The advent of new technologies - e.g. AJAX and JSON (http://www.json.org/) - new services – e.g. Web 2.0 services - and new architectural approaches - e.g. REST - although often based on newly interpreted existing solutions, have deeply changed the way the Web is built and experienced by the users.

Therefore in the current World Wide Web, SOA-based services, Web 2.0 applications, RESTful architectures and systems based on traditional Web (i.e. the so-called XML-over-HTTP) or mixed approaches, coexist. Since the adoption of a specific architectural style affects the overall characteristics of the resulting system, different styles fit better for different user and system requirements. While, in principle, it would be possible to reduce this heterogeneity (e.g. avoiding mixed architectures often characterized by poor performances), it is highly probable that the existence of different user and system requirements in a global system like the Web, even in a specific domain such as the environment, will result in a irreducible heterogeneity that must be addressed. Indeed different systems may provide relevant SDI resources (data and services), and the need for a more general architectural style, capable of accommodating and harmonizing all the different approaches, arise.

SOA was initially conceived to work at enterprise level, but turned out to be very useful in the SoS context. In particular, high level of interoperability can be achieved in SOA networks through the adoption of following some constraints:

- Common payload (the actual data that is being exchanged) and protocol: each service provides an interface that is invoked through a payload format and protocol that are understood by all the potential clients of a service.
- Published and discoverable interfaces: each service has a published and discoverable interface that allows systems to search for services that are best suited for their purposes.
- Loose coupling: services are connected to other services and clients using standard, dependency-reducing, decoupled message-based methods such as XML document exchanges.
- Multiple communication interfaces: services can implement separately defined communication interfaces.
- Composability: Because services are coarse-grained reusable components that expose their functionality through a well-defined interface, systems can be built as a composition of services and evolve through the addition of new services.

The concept of standardization as a means of SOA significantly achieving interoperability, is a key activity in a SoS engineering process. Due to the desire for adaptability in all areas of system of systems, these standards should, at a minimum, be developed over as standards that are “open” to any entity participating or impacted by the time in order to accommodate SoS. Adaptability is necessary in a system of systems, since the challenges encountered in membership or configuration is or can be dynamic, and the relationships among all of the systems in the SoS may not always be known.

SOA was mainly conceived to work at enterprise level; thus, for large (e.g. international) and heterogeneous (e.g. multi-disciplinary) infrastructures. Some of the new challenges that had to be must addressed include: semantic interoperability, heterogeneity of the technological standards heterogeneity, organizational interoperability, and above all the fact that the ICT development obsoletes the technologies used in SOA applications on a time scale much shorter than the life-span of the large SoS network

Service-oriented architecture (SOA) is widely accepted as the paradigm of choice to loosely couple software components in distributed (SDI) applications. However, no agreed conceptual foundation of a geospatial SOA, that is an agreed service meta-model that is also compliant with geospatial service standards of OGC currently exists.

In fact, a number of competing architectural paradigms evolve in-parallel, each with their own respective advantages in certain usage areas. The ORCHESTRA project introduced the idea of technology-independent

RM-OA specifications as a way to allow development and co-existence of technologies in SOA applications. The SANY Sensor Service Architecture already introduces the elements of event-driven and resource-oriented architecture in addition to request/reply architectural style.

Multi-style SOA for SDI Usage Area of the Future Internet should be rigorously based upon comprehensive SOA design patterns and leverage the Future Internet functions. The flexibility offered by multi-style SOA should allow the building of mash-up applications for multiple usage scenarios.

9.3 Relevant standards

9.3.1 Service-Oriented-Architectures (SOA)

SOA – Service Oriented Architecture can be considered both from an organizational and business perspective as well as from a technical perspective. SOA is a means of organizing solutions that promotes reuse, growth and interoperability. It is not itself a solution to domain problems but rather an organizing and delivery paradigm that enables one to get more value using capabilities which are locally “owned” and those under the others control. SOA reflects the reality that ownership boundaries are a motivating consideration in the architecture and systems design.

The central focus of SOA is the task or business function. The central concept of SOA is the service: a mechanism to enable access to a set of one or more capabilities. A service can enable users to perform arbitrarily complex tasks involving the resources which are handled by the service provider and not directly exposed to the user. Therefore, SOA defines a class of architectures which enable loosely-coupled access to generic capabilities provided by service providers. The generality of services in terms of information, structure, semantics, behavior, action and process models require the provision of functionalities supporting visibility and awareness –through service description and policy definition. This makes SOA really powerful but complex, especially if only simple tasks are required.

In the WWW the emergence of SOAP (formerly Simple Object Access Protocol) provided a common specification for service invocation between Web components. Lately SOAP, originally born for conveying remote methods invocation in XML, being fully suitable for generic messaging between objects, evolved in a more general standard for sending services calls targeted to endpoints exposed in the Web and addressed through a specific URL. Further specifications such as WSDL[22] , UDDI[23] , WS-I[24], WS-*, etc. from various standardization bodies, mainly W3C[25] and OASIS[26] make the SOAP suite, a complete set of standards for building SOA over the Web providing service description, cataloguing, security and so on. Indeed, this is currently the most spread solution for e-Business and e-Government systems.

SOAP Version 1.2 is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment. A SOAP Message is made of a Header and a Body. The optional SOAP Header element contains application specific information (like authentication, payment, etc) about the SOAP message. The SOAP Body provides a mechanism for transmitting information to an ultimate SOAP receiver, that is the service provider. An important characteristics is that a SOAP Message could be transmitted using any protocol as long as it allows to transfer the serialized Infoset to the destination. Several transport mechanisms (bindings) are defined for SOAP using application-level protocols such as HTTP and SMTP. This generality is obtained at the expenses of the loss of protocol-specific characteristics. For example the HTTP binding utilizes HTTP as a transport-level protocol: the semantics of the request line and of most of the HTTP headers is actually lost.

---

[22] See http://www.w3.org/TR/wsdl

[23] See http://www.oasis-open.org/committees/uddi-spec/

[24] See http://www.ws-i.org/

[25] See http://www.w3.org/

[26] See http://www.oasis-open.org/

9.3.2 Representational State Transfer (REST)

The W3C document "Web Services Architecture" makes SOAP the fundamental basis for a "new" Web, a Web of exposed services instead of shared documental resources. But the great success of SOAP in many application fields like e-Business and e-Government did not guarantee the same success in other fields and applications characterized by different requirements. SOAP fits well to SOA, where different organizations expose complex services (e.g. banking transactions, travel reservations, commercial orders) implemented in background facilities which can be composed in workflows for carrying out high-level business processes. These great capabilities have the drawback of a complex infrastructure for services discovery, description, etc. However, common Web applications, in particular the so-called Web 2.0 services, are light services dedicated to publish and access structured and semi-structured information (e.g. Web sites, Web interfaces to databases and repositories, Content and Document Management Systems, blogs, etc.), a context where SOA seems to be overloading. W3C have proposed a new vision of the Web architecture to make it conform to its original concept. Such architecture is based on an architectural style named REST (Representational State Transfer).

REST is a resource-oriented style for distributed systems defined to describe the original Web architecture and to guide its future evolution preserving its fundamental characteristics, scalability. Although REST is defined bearing in mind the Web, all the architectures satisfying the REST constraints are REST based architectures, not only the Web itself. The term "RESTful" was introduced to describe system architectures based on the REST style. RESTful architectures present two essential characteristics deriving from the Uniform Interface constraint:

a) all the significant resources are addressed and accessible through the same set of methods (common interface).

b) logical connections between resources are made explicit as hyperlinks.

Note that REST is not a technology. In particular, REST is neither simply XML+HTTP nor any HTTP API which seem to be common misunderstandings.

9.3.3 Web 2.0

It is still difficult to agree on what the Web 2.0 really is. One definition is *"a set of economic, social, and technology trends that collectively form the basis for the next generation of the Internet—a more mature, distinctive medium characterized by user participation, openness, and network effect"*. Comparing some of the most known applications some commonalities that can be considered the principles behind the Web 2.0:

1. The Web As Platform
2. Harnessing Collective Intelligence
3. Data is the Next Intel Inside
4. End of the Software Release Cycle
5. Lightweight Programming Models
6. Software Above the Level of a Single Device
7. Rich User Experiences

In order to facilitate the development of the so-called Web 2.0 applications, specific strategies, patterns and technologies have been developed or improved. They are often indicated with well-known (and often misunderstood) terms.

- *Mash-up (or mashup)*: A mashup is a technique for building applications that combine data from multiple sources to create an integrated experience. In the Web 2.0 this is often done directly in the browser (consumer mashup) using open Web APIs (e.g. Google Maps API, Wikipedia API, OpenLayers, etc.). The mashup approach helps to achieve the “Web As Platform” and “Lightweight Programming Models” principles

- *Lightweight technologies*: technologies that do not require a heavy upfront investment or operational requirements. These are simpler and less cumbersome to work with. The downside, of course, is that lightweight technologies can be less feature rich than their more "heavyweight" alternatives. Examples of technologies considered lightweight are JSON (Javascript Object Notation) for semi-structures data representation, and Javascript/ECMAScritpt as programming language. In the Web 2.0 they help to reach the rapid development required by the “End of the Software Release Cycle” and the “Lightweight Programming Models” principles.
- *AJAX*: The term AJAX (or Ajax) was coined to a new approach to web applications development characterized by:
  - standards-based presentation using XHTML and CSS;
  - dynamic display and interaction using the Document Object Model;
  - data interchange and manipulation using XML and XSLT;
  - asynchronous data retrieval using XMLHttpRequest;
  - and JavaScript binding everything together.

  The term was actually a shorthand for Asynchronous Javascript and XML. However, Ajax now encompasses the use of specific technologies like Javascript and XML. Other technologies can be adopted for scripting (e.g. Adobe Flash instead of Javascript) or data representation (e.g. JSON instead of XML). Even the XMLHttpRequest object can be replaced by other techniques for asynchronous retrieval of Web resources (e.g. by use of IFrames). What actually characterizes Ajax is its application model: an Ajax application eliminates the start-stop-start-stop nature of interaction on the Web by introducing an intermediary—an Ajax engine—between the user and the server. The Ajax engine allows the user’s interaction with the application to happen asynchronously—independent of communication with the server.

10 Technology viewpoint

10.1 Overview

The technology viewpoint is illustrated in Figure 15.

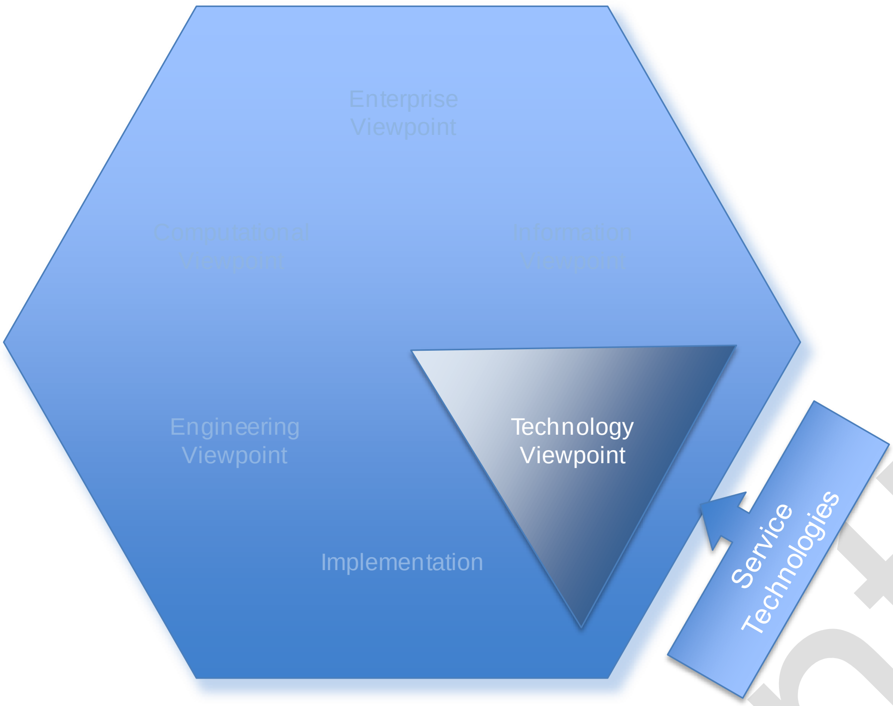


Figure 15. The technology viewpoint.

The technology viewpoint describes the implementation of the ODP system in terms of a configuration of technology objects representing the hardware and software components of the implementation. It is constrained by cost and availability of technology objects (hardware and software products) that would satisfy this specification. These may conform to platform-specific standards that are effectively templates for technology objects.

10.2 Architectural Classification according to Cloud Computing Service Categories

Cloud computing has a significant potential in the context of the management, preservation and sharing of data. It is a computing model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (NIST 2011). Thus, cloud computing focuses on the delivery of computing as a service, where shared resources, software and information are provided to computers and other devices as a utility, usually over the Internet.

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). The SaaS, PaaS and Iaas providers all have a range of both generic and specific enablers.

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. Further information on this subject is provided in the following Clause 11.

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.

11 Services and Security

11.1 Services - security and rights management

Security and Rights Management is one of the core components identified in the SDI Component Reference model. The primary function of the component is to manage access and usage rights to resources, including spatial data and spatial data services. The Security and Rights Management component within the Service-centric View integrates with the Identity and Licence component within the Data Centric View.

Effective data sharing and re-use of spaital data and spatial data services presents more than purely a technical challenge. A number of business interoperability aspects need to taken into account in order to streamline security and rights mangement and reduce the complexity accessing and using spatial data sets and spatial data services from the application and user perspective. A key purpose of the security and rights managment component is enable providers of spatial data services to evolve towards more harmonised ways of accessing and licensing the re-use of spatial data sets and sptial data services.

The key objectives of the Security and Rights Management component are to:

- Make it easier for users to understand the licensing terms in a simple and understanable way.
- Allow providers of spatial data sets and spatial data services to license their resources under well defined licensing models.
- Allow consumers to easily combine content from multiple providers and understand how that combined data might be re-used.
- Ensure that privacy is supported in compliance with data protection regulations
- Encourage the evolution of standard and harmonised approaches which facilitate interoperability at technical and business level.

System Management and Security Enablers, as identified in Clause 7.4.1, 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. Also, security and rights managements aspects are related to the System management/Security enablers on the SaaS and PaaS level.

11.1.1 Managing access and re-use rights

Many spatial data providers make use of different approaches to manage access and re-use of their spatial data sets and spatial data services. Security approaches typically are used to control authentication, authorisation and access control. Once the spatial data set or spatial data services has been accessed, then various legal measures many then be used to control the onward re-use of those data.

While it is recognised that many existing spatial data infrastructures this way, the Security and Rights Management component should be architected so providers are able to adopt standardised and harmonised licensing models and make use of technical protection measures as well.

11.1.2 Key drivers behind effective rights management

Geospatial content and services are becoming increasingly available in digital form and it is now easier to copy, modify and re-use that content. Organisations involved in the capture and sharing of geospatial content now find that they need to protect their intellectual property as it is shared through the digital network:

- **Content security**: When geospatial content contains private or sensitive information, providers need to control access to the data. Content security may be necessary to support a commercial business model, or to comply with agreements with other providers, or to respect privacy rights, or to restrict access to sensitive data.

- **Content licensing**: Licensing agreements offer an additional level of intellectual property protection between providers and consumers of content. A licence agreement grants certain rights to consumers that might include broader rights for use and re-use of content than would normally be permitted under a standard copyright statement. Very often content is licensed under a given provider's specific licensing terms and conditions, however some communities have developed harmonized licensing models which can simplify the access and re-uses of content from the consumer's perspective.

- **Charging for content**: Intellectual property rights and charging for content are two separate issues. Many providers' business models require that content is charged for based on some form of pricing model. Creating a sustainable network for geospatial content requires some form of recognition of the value generated by the consumer of the content.

11.1.3 Challenges of implementing rights management

Rights management cuts across political, legal, social and technical aspects and aligning these aspects to arrive at a suitable implementation poses a significant challenge. Our existing political, legal and social frameworks were built for a different world based on managing and trading physical property. Adapting and modifying these frameworks to support managing and sharing intellectual property represents a dramatic cultural change. Within the geospatial community we are facing the challenges of increasing pressures to make data 'freely' available whilst many content providers need to sustain their businesses by charging for data.

When developing a spatial data infrastructure, achieving consensus on an implementation approach can be elusive. Whilst there is often a general recognition that a problem exists, different perspectives can make it difficult to agree a solution. Also, it is a challenge to gain traction and ownership of the rights management implementation within a given geospatial community of interest. Often those responsible for the central functions of a geospatial infrastructure are reluctant to implement any rights management capacity but rather delegate responsibility to individual providers. Providers are often constrained by their own licensing terms and conditions, and may be resistant to solutions that involve the adoption of more harmonised licensing models. Individual providers might implement point solutions, but this is then unlikely to support any form of interoperability across providers. Consumers tend to welcome rights management solutions based on simplified and harmonised licensing models, but are resistant to solutions which they believe will impose additional constraints on how data is accessed and used.

11.1.4 Securing OGC Web Services

Within this sub-clause, the work done by the OGC for security and rights management is used as an illustrative example of how security might be implemented. OGC Web Services can be secured with information technology standards developed by other standards organisations. Exchanging and processing of geospatial Information in a federation requires interoperability on different levels:

- Data Level Interoperability ensures the ability to "consume" the information

- Service Level Interoperability ensures the ability to exchange / obtain the information to be "consumed"

- Security Level Interoperability ensures the ability to the above in a reliable and trustworthy fashion

Interoperability at all the above levels can be implemented by using standards from the OGC and other bodies

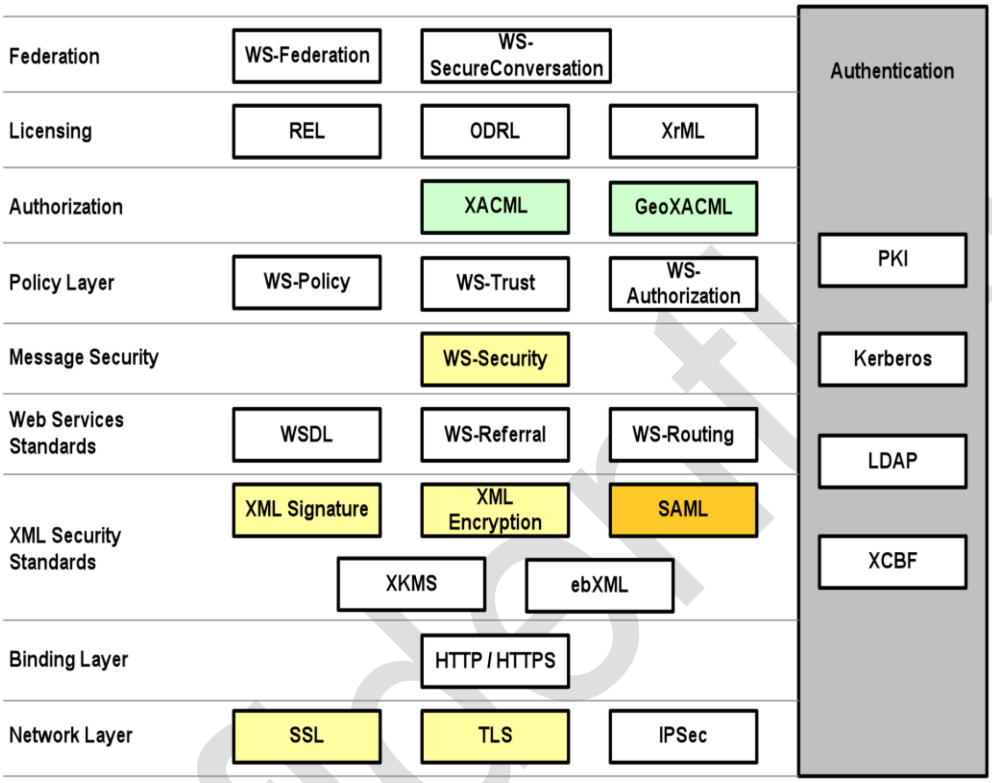


Figure 16. Security mechanisms used by OGC

11.1.5 Security Requirements

ISO/IEC10181(all parts) define a set of requirements in terms of a security framework for open systems. In order to protect the exchange of information between secured systems and the management of the stored data, the standard states that "... security services may apply to the communicating entities of systems as well as to data exchanged between systems, and to data managed by systems."[27]. In subsequent parts of the standard, the requirements and the following security frameworks are defined:

- **Authentication Framework**: ISO 10181-2 defines all basic concepts of authentication in Open Systems: It identifies different classes of authentication mechanisms, the services for their implementation and the requirements for supporting protocols. It further identifies requirements for the management of identity information.

[27] See OGC and Rights Management - An OGC White Paper - https://portal.opengeospatial.org/files/?artifact_id=46622&version=1

- **Access Control Framework**: ISO 10181-3 defines all basic concepts for access control in Open Systems and the relation to other frameworks such as the Authentication and Audit Frameworks.
- **Non-repudiation Framework**: ISO 10181-4 refines and extends the concepts of non-repudiation, given in ISO 7598-2. It further defines general non- repudiation services and the mechanisms to provide these services.
- **Confidentiality Framework**: ISO 10181-5 defines the basic concepts of confidentiality, identifies classes of confidentiality mechanisms and their maintenance. It further addresses the interactions of the confidentiality mechanisms with other services.
- **Integrity Framework**: ISO 10181-6 defines the basic concepts of integrity, identical to the Confidentiality Framework.
- **Security Audits and Alarms Framework**: ISO 10181-7 defines the basic concepts for security audit and alarms and the relationship to other security services.
- **Availability**: This is a requirement that is in particular important in a Service Oriented Architecture. It is defined in ISO 7498-2 as „The property of being accessible and useable upon demand by an authorized entity. " Adapting that to a Service means that the service shall be executable whenever there is a need.

Annex A
RM-ODP viewpoints and the Parts of TR 15449

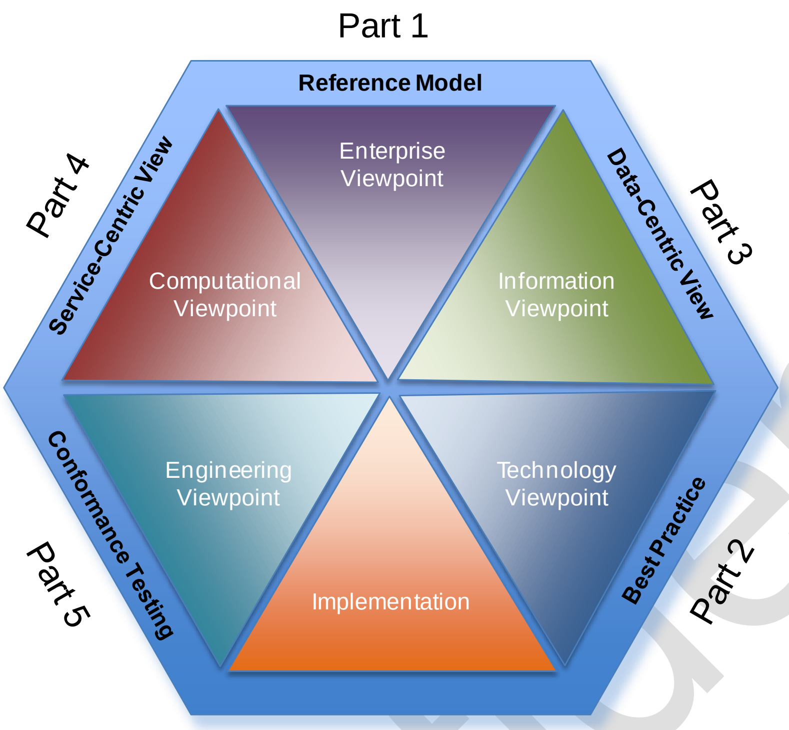


Figure 17. RM-ODP viewpoints related to TR 15449 parts and implementation

Figure 17 is an illustration on how the different parts of TR 15449 themselves can be viewed as descriptions and recommendations with a main emphasis on some of the different RM-ODP viewpoints, if we view the whole domain of SDI in general as a focus area. Part 1 provides an overview and context for the area of SDIs and gives an introduction to the other parts. Part 2 discusses experiences and best practices from past projects and activities that has resulted in use of technologies in the context of implementations. Part 3 focuses on a data-centric view on SDIs with a natural anchor in the Information viewpoint. Part4 (this part) focuses on a service-centric view on SDIs founded on the computational viewpoint (sometimes also referred to as the services viewpoint). Part 5 (proposed) will focus on conformance testing, and how actual implementations can be validated for conformance with models and specifications – including the realisation of mechanisms to support this from an engineering viewpoint.

Annex A
Example – Use case based methodology

The following illustrates an example of a use case based methodology for the identification of needed resources in an SDI context, including both data and service resources.

11.1.6 Example of the application of a methodology based on use cases

This section provides an overview about an SDI oriented Use Case Analysis Methodology. This methodology is derived from the SERVUS methodology[28] 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 SERVUS methodology relies upon a resource model as a common modelling language which is derived from the Representational State Transfer (REST) architectural style for distributed hypermedia systems as conceived by (Fielding, 2000). 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.

[28] See 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. http://digbib.ubka.uni-karlsruhe.de/volltexte/1000016721

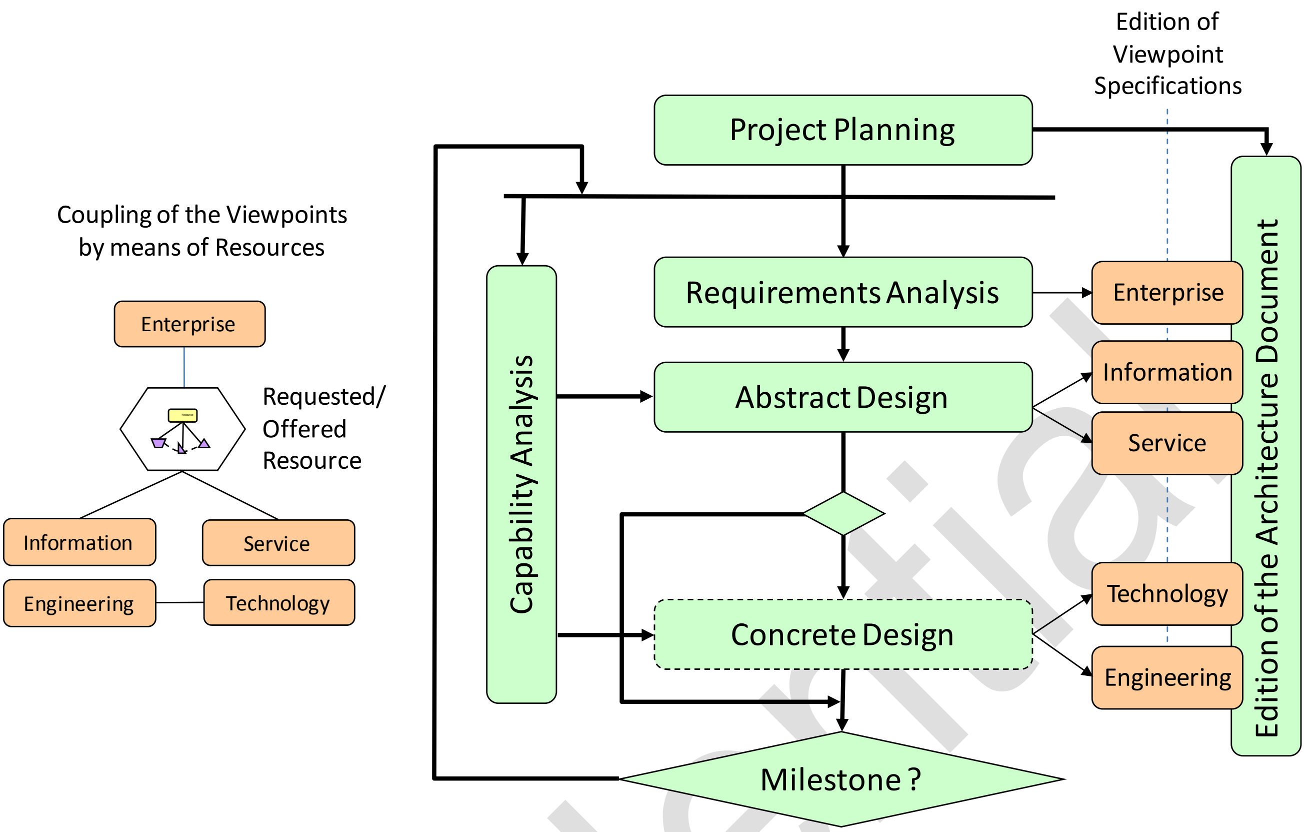


Figure 6 Relationship of RM-ODP viewpoints and analysis and design

Figure 6 shows how the various RM-ODP viewpoints typically are focused on during the different steps of service analysis, design and implementation. The main viewpoint during initial analysis is the enterprise viewpoint. This also serves as the foundation for describing the resources (in terms of data, services and/or sensor information) which is required. An initial step will then be to compare the requested resources with potentially offered resources through a discovery and search process, in order to identify if the request for resources can be met by resources that already are available.

11.1.7 Use Case and Requirements Modelling

Use case modelling has been shown 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. geospatial, environmental) 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 should 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 the geospatial context use cases are typically described in a semi-formal way, based on a structured textual description in tabular form derived from a template. Recent European research projects (such as SANY[29], ENVIROFI[30], EO2HEAVEN[31] and TRIDEC[32]) based the description of their use cases on a similar template.

Based upon this approach, [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. 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.

This approach provides a basis for use case development. 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 expected that each use case will be described 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 Annex B. 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. Use Case Analysis Process

Figure 7 illustrates the analysis phase as a prelude of the SERVUS Design Methodology[33]. As a first step of an analysis iteration loop is a set of preliminary use cases (UC) is identified, mostly by those thematic experts who drive the study.

---

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

[30] EU FP7 FI PPP project, ENVIROFI, http:///www.envirofi.eu

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

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

[33] See Usländer, T. and 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

The methodology proposes that use cases are initially described in structured natural language but already contain the list of requested resources. 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. 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.

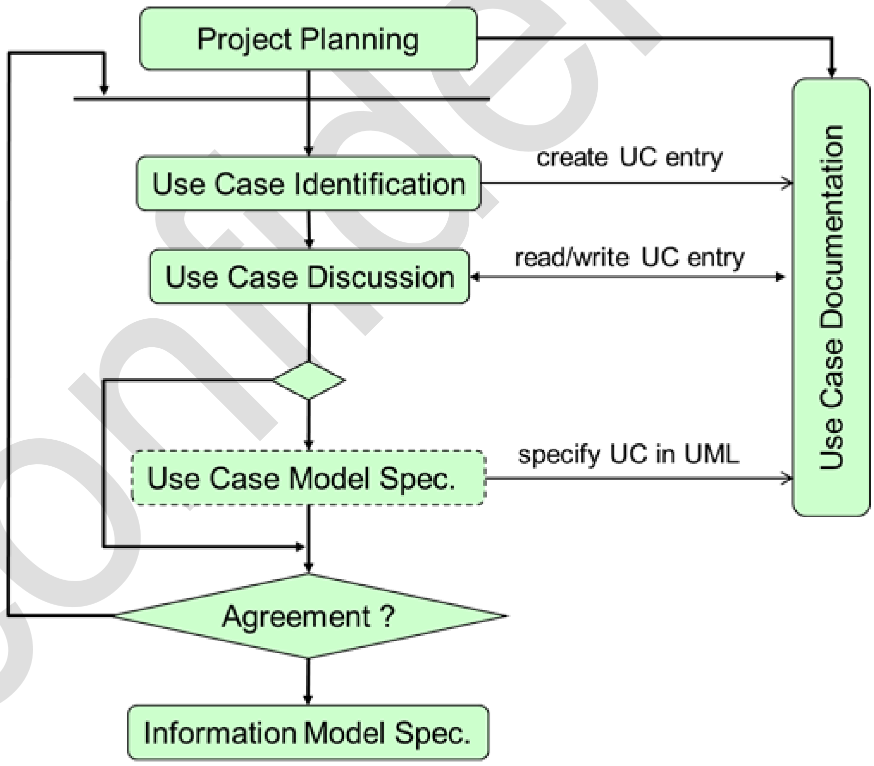


Figure 7. Procedure of Use Case Analysis

Annex B
Example – Use case template

This template is an extended version of the original template defined by Cockburn (2001)[34], in particular extended with a possibility to describe Requested Information Resources found suitable in an SDI setting.

Use Case Template	Description	Examples
Use Case Name	Name of the use case	Visualise proposed water height after the tsunami event
Use Case ID	Unique identifier of a use case	
Revision and Reference	Revision = version number of use case ID Reference = URL of the use case (you get the URL by right-clicking on the entry in the index column)	V02, http://SDI.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. The actual UML diagram figure may be added at the bottom of the template by uploading a bitmap generated from a UML editor.	
Status	Status of the use case development	One of the following: • Planned • 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: • Must have: The system must implement this goal/ assumption to be accepted. • Should have: The system should implement this goal/ assumption: some deviation from the goal/assumption as stated may be acceptable.

34 Cockburn, A. Writing Effective Use Cases. ISBN-13: 9780201702255. Addison-Wesley (2001)

		• 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 overall reference architecture.	*Context dependent*
Actor	List of users of the use case (actors)	Examples may be citizen, administrator or employee of a SDI 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 (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. 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) • user-specific effect (read, update) • alert (manage)
Preconditions	Description of the system/user status statement) that is required to start the execution of the use case. 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. 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. 2. He specifies one or more

		components (default should be all). 3. He sets a time-frame (last 24 hours, last week, last month) 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.	

Table 2. Description of the Use Case Template

Annex C
Service Modeling - SoaML

SoaML (Service oriented architecture Modeling Language) is a UML profile and metamodel for service modeling recently standardised by OMG. The SoaML specification defines three different approaches to specifying services; simple interfaces, service interfaces and service contracts.

The SoaML specification defines a UML profile and a metamodel that extends UML to support the range of modelling requirements for SOA, including the specification of systems of services, the specification of individual service interfaces, and the specification of service implementations. The SoaML metamodel extends the UML metamodel to support an explicit service modelling in distributed environments. This extension aims to support different service modelling scenarios such as single service description, service-oriented architecture modelling, or service contract definition. This is done in such a way as to support the automatic generation of derived artefacts following the approach of Model Driven Architecture (MDA)

UML is a general-purpose modelling language for visualising, specifying, constructing and documenting artefacts of software-intensive systems. A UML profile customizes UML for a specific domain or purpose by using extension mechanisms such as stereotypes and metaclasses. Figure 9 shows the main stereotypes defined in the UML profile for SoaML, e.g. the stereotype «ServiceInterface» extends the UML metaclass *Class*.

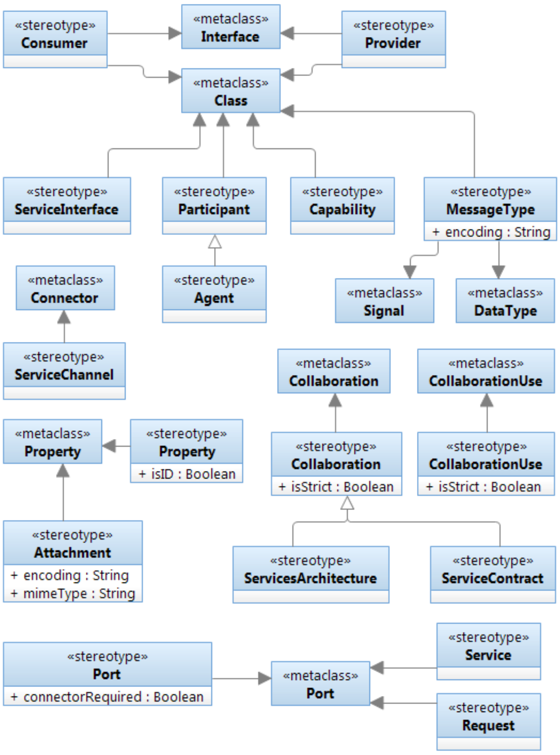


Figure 9. Main UML extensions defined as stereotypes in the UML Profile for SoaML

SoaML extends UML in six main areas: Participants, service interfaces, service contracts, services architectures, service data and capabilities.

- *Participants* are used to define the service providers and consumers in a system. A participant may play the role of service provider, consumer or both. When a participant acts as a provider it contains *service ports*, and when a participant acts as a consumer it contains *request ports*.

- *Service interfaces* are used to describe the operations provided and required to complete the functionality of a service. A service interface can be used as the protocol for a service port or a request port.

- *Service contracts* are used to describe interaction patterns between service entities. A service contract is used to model an agreement between two or more parties. Each service role in a service contract has an interface that usually represents a *provider* or a *consumer*.

- *Services architectures* are used to define how a set of participants works together for some purpose by providing and using services. The services are expressed as service contracts in a services architecture.

- *Service data* are used to describe service messages and message attachments. The *message type* is used to specify the information exchanged between service consumers and providers. An *attachment* is a part of a message that is attached to rather than contained in the message.

- *Capabilities* represent an abstraction of the ability to affect change. Capabilities identify or specify a cohesive set of functions or resources that a service provided by one or more participants might offer.

SoaML supports different approaches to SOA. This has resulted in the definition of different but overlapping language constructs in the UML profile. The specification distinguishes between three different approaches to specifying a service:

- The **simple interface** based approach uses a UML interface to specify a one-way service interaction.
- The **service contract** based approach extends a UML collaboration to specify a binary or n-ary service interaction.
- The **service interface** based approach extends a UML class to specify a binary or n-ary service interaction.

Both the service contract and service interface based approaches entail the specification of simple interfaces, typically one for each of the roles participating in the service interaction. Thus a service contract or a service interface can be seen as an extension of the simple interface based approach.

SoaML supports the specification of multiple SOA architectural styles – such as both synchronous (RPC) style and asynchronous document/RESTful services style, and also events/signals and sensors as services.

Typically the creation of a service model comprises the following aspects:

- Identify services, the requirements they are intended to fulfill, and the anticipated dependencies between them. (Derive this possibly through use case analysis with respect to an architectural references model, based on specifications from the enterprise viewpoint).

- Specify services including the functional capabilities they provide, what capabilities consumers are expected to provide, the protocols or rules for using them, and the service information exchanged between consumers and providers.

- Defining service consumers and providers, what services they consume and provide, how they are connected and how the service functional capabilities are used by consumers and exposed by providers in a manner consistent with both the service specification protocols and fulfilled requirements.

- Defining the policies for using and providing services and the quality of service provided.