Title:
A5.2-D3 [3.1] Specification Introduction and Overview
Editor:
Thorsten Reitz (FHG-IGD)
Authors:
Thorsten Reitz (FHG-IGD), Ana Belen Anton (ETRA), Marian de Vries (TUD), Daniel Fitzner (FHG-IGD)
References:
A5.2-D3 [3.0] A Lightweigt Introduction to the HUMBOLDT Framework A1.8-D4 User Involvement Document - Third Version A5.2-D2 [2.0] Specification Introduction and Overview A7.0-D1 Concept of Data Harmonisation Processes A5.3-D3 HUMBOLDT Commons Specification / Framework Common Data Model 3.0 A5.2-D3 [3.2] Mediator Service Component Specification A5.2-D3 [3.3] Conceptual Schema Specification and Mapping A5.2-D3 3.4 Context Service Component Specification A5.2-D3 [3.5] Workflow Design and Construction Service Component Specification A5.2-D3 [3.7] Information Grounding Service Component Specification A5.2-D3 [3.6] Processing Components General Model and Implementations

Status/Quality Assurance:	
☑ Document Outline	☑ Review WP05 Leader
☑ Content Draft	☑ Review WP Leaders (WP04, WP08)
☑ Request for Comments	☑ Review ExB (COORD, TM, COMM)
☑ Final	☐ Review Associated Projects (...)

Delivery Date:	**03.12.09**

Short Description:
This document focuses on providing an overview of the HUMBOLDT framework for service integration and the toolset for data harmonisation. It details the approach for the creation of this third milestone specification, gives a detailed example of HUMBOLDT software usage and provides an overview of the framework and its components. This overview is formalized in the form of the RM-ODP viewpoints, and starts with a high-level business requirements perspective. For a more lightweight and condensed introduction to the HUMBOLDT software, please refer to A5.2-D3 [3.0]. This document is based on Document [2.0] of A5.2-D2, but stands on its own.

Keywords:

Framework specification, methodology, approach, rationale, architecture, process.

History:

Version	Date	Author(s)	Status	Comment
001-337	27.03.09	Thorsten Reitz	NEW	Initial version of the Specification Introduction.
002-346	06.04.09	Thorsten Reitz	RFC	Complete except chapter 5.4
003-366	16.04.09	Thorsten Reitz	RFC	Almost content-complete version, provided for i/e review
003-444	28.04.09	Thorsten Reitz	FINAL V2	Integrated reviews by Daniel Fitzner, Eva Klien
004	30.11.09	Daniel Fitzner / Thorsten Reitz	FINAL V3	Updated for V3.

Table of contents

Figures

Tables

1 Introduction

1.1. Purpose and Structure of this document

Deliverable A5.2-D3, the Specification of the HUMBOLDT Framework M3.0, consists of a set of separate documents. This document (Document [3.1]) provides an introduction to the specification work by giving a rationale and an overview of the HUMBOLDT main software products. With this document, we provide the context for the rest of the A5.2-D3 documents (Documents 3.1-3.7) that deal with the detailed component specifications. For a more lightweight and condensed introduction to the HUMBOLDT software, please refer to *A5.2-D3 [3.0] A Lightweight Introduction to the HUMBOLDT framework.*

The following list gives an overview on the structure of A5.2-D3 documents:

- four framework component specifications (Documents [3.2], [3.4], [3.5] and [3.7]),
- the specification of the HUMMBOLDT Alignment Editor for specifying mappings between conceptual schemas (Document [3.3.1])
- the common framework for conceptual schema harmonisation (Document [3.3])
- high-level aspects of processing services and six processing service specifications (Document [3.6]).

A specification of the common data structures used in the Framework can be found in D5.3-D3 HUMBOLDT Commons Specification / Framework Common Data Model V3. Further, document [3.3] provides the conceptual frame for conceptual schema transformation in HUMBOLDT.

This document describes the approach for the creation of the 3.0 specification, the rationale for the selection of the individual components and thus gives an overview of the framework as it stands in M3.0. Furthermore, it contains a reflection on the lessons learned from the Framework M1.0 and M2.0 specifications and the implementation work carried out so far.

The document is structured into three main sectionsfor providing answers to the following questions:

- If you are mostly interested in the question “why was something developed”, please check chapter 2.
- If you are mostly interested in the aspect “how” something was developed, please go to chapter 3.
- If you are mostly interested to directly go to the “what has been developed”, please read chapters 4 to 5.
    - Chapter 4 gives an End-to-End example of different users interacting with HUMBOLDT software.
    - Chapter 5.1.1 explains the underlying, generic harmonisation and transformation processes that the HUMBOLDT software supports.
    - Chapters 5.1. to 5.6 present a summary of the actual specification, including an enterprise viewpoint, the logical composition of the framework, construction of services from these components and technical aspects.

The work programme of this phase focused on two main areas. First, the work with application contacts has been intensified, resulting in a number of scenario requirements that have been collected and integrated into the specification documents. Second, the component specification have been edited and reworked with the goal of reducing complexity in the explanations and making them easier understandable for non-technicians.

1.1.1 Responsibilities

1. **[3.2] Mediator Service Component Specification**
   - Responsible Editor: Thorsten Reitz (FHG-IGD)
   - Application Contact: Annette Breu
   - Implementation Team Leader: Bernd Schneiders (LOG)
   - Other Contributors: Marian de Vries (TUD), Ana Belén Antón (ETRA)
2. **[3.4] Context Service Component Specification**
   - Responsible Editor: Daniel Fitzner (FHG)
   - Application Contact: Moses Gone (FHG)
   - Implementation Team Leader: Not required for M3
   - Contributors: Ana Belén Antón (ETRA)
3. **[3.5] Workflow Design and Construction Service Component Specification**
   - Responsible Editor: Daniel Fitzner (FHG-IGD)
   - Application Contact: -.-
   - Implementation Team Leader: Moses Gone (FHG-IGD)
   - Contributors: Thorsten Reitz (FHG-IGD), Moses Gone (FHG-IGD)
4. **[3.7] Information Grounding Service Component Specification**
   - Responsible Editor: Ana Belén Antón (ETRA)
   - Application Contact: Moses Gone
   - Implementation Team Leader: Not required for M3
   - Contributors: -.-
5. **[3.3] Conceptual schema specification and mapping**
   - Responsible Editor: Thorsten Reitz (FHG-IGD)
   - Application Contact: Astrid Fichtinger (TUM)
   - Other Contributors: Marian de Vries (TUD)
6. **[3.3.1] HUMBOLDT Alignment Editor Specification**
   - Responsible Editor: Thorsten Reitz (FHG-IGD)
   - Application Contact: Sisi Zlatanova (TUD)
   - Implementation Team Leader: Thorsten Reitz (FHG-IGD)
   - Contributors: Marian De Vries (TUD), Jan Schulze-Althoff (ETHZ), Cristian Filippo Daffinà (TPZ)

For M3, the General Model for Processing Components and all individual processing component specification have been merged into a single document *A5.2-D3 [2.6] Processing Components General Model and Implementations*.

7. **[2.6] Processing Components General Model and Implementations**
    - Responsible Editor: Jan Jezek (HSRS)
    - Application Contact: Daniel Fitzner (FHG)
    - Implementation Team Leader: -.-
    - Contributors: Thorsten Reitz (FHG-IGD), Daniel Fitzner (FHG)
8. **Conceptual Schema Transformer Specification**
    - Responsible Editor: Thorsten Reitz (FHG)
    - Application Contact: Astrid Fichtinger (TUM)
    - Implementation Team Leader: Jan Jezek (HSRS)
    - Contributors: Jan Kolar (INGR-CZ), Jan Schulze-Althoff (ETHZ), Marian de Vries (TUD), Thorsten Reitz (FHG-IGD), Steve James (LOGCMG)
9. **Language Transformer Specification**
    - Responsible Editor: Jan Kolar (INGR-CZ)
    - Application Contact: Daniel Kristof (FÖMI)
    - Implementation Team Leader: Jan Kolar (INGR-CZ)
    - Contributors: Thorsten Reitz (FHG-IGD)
10. **Multiple Representation Merger Specification**
    - Responsible Editor: Jan Kolar (INGR-CZ), Thorsten Reitz (FHG-IGD)
    - Application Contact: Roderic Molina (GISIG)
    - Implementation Team Leader: Jan Kolar
    - Contributors: Marco Corsi (TPZ), Ana Belén Antón (ETRA)
11. **Coordinate Transformation Service Specification**
    - Responsible Editor: Thorsten Reitz (FHG-IGD)
    - Application Contact: Daniel Kristof (FÖMI)
    - Implementation Team Leader: Thorsten Reitz (FHG-IGD)
    - Contributors: -.-
12. **Edge Matching Service Specification**
    - Responsible Editor: Thorsten Reitz (FHG-IGD)
    - Application Contact:
    - Implementation Team Leader: Ana Belen Anton (ETRA)
    - Contributors: Roderic Molina (GISIG), Ana Belen Anton (ETRA)

The following components are also part of the HUMBOLDT Framework, but their specification has not been officially continued throughout: Quality Measurement in Transformation, HUMBOLDT Model Editor and HUMBOLDT Model Repository. Therefore, the M2.0 specification documents of these components sill represent the current state.

1.2. Who should read this document?

This document is intended to be read by everybody in the HUMBOLDT project who plans to use the HUMBOLDT framework, either directly or indirectly. This means mostly the scenario groups. Especially the Specification Introduction and the Enterprise Viewpoint (chapters 5.1 and 5.2.) give an introduction to the scope and philosophy of the framework. For developers, this document is meant to provide a high-level introduction to the individual component specification documents.

1.3. Definitions valid in this document

This section gives the core definitions valid for the context of this document.

Service Oriented Architecture (SOA)	Service Oriented Architecture is an architectural and organizational paradigm for organizing and utilizing distributed processing and storing capabilities that may be under the control of different ownership domains by defining loosely-coupled relationships between producers and consumers[1].
Service Component Architecture (SCA)	A specification dealing with composition and communication of service components, independent of both the communication and the component implementations. Please refer to deliverable A3.4-D1, p. 59
Service Component	A self-sustained logical Software Component which offers operations described through its public interface. Such a component can be deployed as a Web Service or as a local service. A Service Component is internally constructed from multiple Modules.
Spatial Data Infrastructure (SDI)	A SOA of spatial data, metadata, users and tools that are interactively connected in order to use spatial data in an efficient and flexible way. Tools and services connect via computer networks to the various sources.
Information Model (IM)	An Information Model is, within the field of data modelling, an abstract but formal representation of entities including their properties, relationships and the operations that can be performed on them. The usage of this term in this document does not distinguish between the logical or conceptual schema.
Module	A module is a software part that consists of multiple packages that does not have an interface to be exposed as a service, but is rather used internally in Service Components.
Data Harmonisation	(Creating) the possibility to combine data from heterogeneous sources into integrated, consistent and unambiguous information products, in a way that is of no concern to the end-user

Table 1: Definitions valid in this document

1.4. Requirement References in this document

Throughout this document, you will find identifiers formatted like this: HUMB01. These refer to high-level requirements listed in the Enterprise Viewpoint (chapter 5.2.1, Business Requirements).

1 Based on the OASIS and OMG definitions of SOA.

2 M3.0 Objectives of Specification

Specification work has to fulfil several objectives coming from multiple sources, which then also translate into high-level requirements in the Enterprise Viewpoint. First-order objectives are the functionalities mentioned in the Description of Work and confirmed in the Detailed Implementation Plan for the project months 24 to 42, which are listed in chapter 2.2.

2.1. M2.0 Objectives

The objectives for M2.0 of the HUMBODOLDT specification have been categorized into *objectives from the detailed implementation plan*, *objectives from internal project results* and *objectives from external and internal feedback*. All of these objectives have already been satisfied or at least discussed within the M2.0 specification. However, some of the M2.0 objectives, such as objectives from the HUMBOLDT Handbook of Standards or objectives derived from reviewer comments are relevant for M3.0 as well and are therefore included again.

2.2. Objectives from the Detailed Implementation Plan PM24-42

The following passages from the WP05 description of DIP24-42 refer to the objectives of the M3.0 specification:

> **WP05: Combined Abstract objective descriptions**
>
> *PM36: A5.2-D3 Framework specification Version 3.0*
> Functional Extension of the Framework towards new requirements not yet covered

For a discussion on how this requirement has been covered see section 2.4.3.

2.3. Objectives from internal project results

2.3.1 State of the Art, Handbook of Standards

One of the core elements of the HUMBOLDT philosophy is to re-use the existing, and the state of the art documents confirmed our original assumption that there would be quite a few suitable standards and tools around that could be incorporated into the HUMBOLDT framework.

The following table shows the specification-related standards recommended in the State of the Art deliverables A3.2 to A3.5 and collected in the Handbook of Standards (HoS 3.0) deliverables and their integration status in the specification and implementation work.

Source	***Title***	***Bindingness***	***Status***
HoS 2.0	ISO RM-ODP	Mandatory	Used in all specification documents
HoS 2.0	OMG UML 2.0	Mandatory	Used in all specification documents
HoS 2.0	OGC CSW 2.0.1	Mandatory	Used in implementation (Information Grounding Service)
HoS 2.0	OGC Web Feature Service 1.1	Mandatory	Used in specification (Mediator Service, Information Grounding Service)

HoS 2.0	OGC Filter Encoding 1.1	Mandatory	Used in implementation (Spatial and other Constraints in Mediator Service)
HoS 2.0	OGC Web Coverage Service 1.1	Mandatory	Used in specification (Mediator Service)
HoS 2.0	OGC Web Map Service 1.1.1	Mandatory	Used in implementation (Mediator, Information Grounding Service)
HoS 2.0	OGC Styled Layer Descriptor Profile of the Web Map Service 1.1	Optional	Used in implementation (Context Service, Mediator Service)
HoS 2.0	OGC Web Map Context Document 1.1	Optional	Used in implementation (Context Service, Mediator Service)
HoS 2.0	OGC Styled Layer Descriptor 1.0	Mandatory	Not used, due to usage of v1.1 and Symbology Encoding
HoS 2.0	OGC Symbology Encoding 1.1.0	Optional	Used in implementation (Context Service, Mediator Service)
HoS 2.0	ISO 19115/19139	Mandatory	Used in implementation (all components)
HoS 2.0	ISO 19119	Mandatory	Used in specification (Information Grounding Service, Workflow Design and Construction Service)
HoS 2.0	ISO 15836 Information and documentation - The Dublin Core metadata element set	Optional	Currently unused
HoS 2.0	INTERLIS 2	Recommended	Currently unused
HoS 2.0	ISO 19103 Geographic information - Conceptual schema language	Recommended	Indirectly used (using implementation/less abstract specifications in all components)
HoS 2.0	ISO 19109 Geographic information - Rules for application schema	Recommended	Used in implementation (most components)
HoS 2.0	OGC Geography Markup Language 3.2.1	Mandatory	Used in implementation (most components)
HoS 2.0	GeoTIFF	Mandatory	Currently unused to due lacking Coverage support
HoS 2.0	Hierarchical Data Format - Earth Observation Systems	Optional	Currently unused

HoS 2.0	DTED, NTIF, NetCDF	Optional	Currently unused
HoS 2.0	ISO 19125 Geographic information - Simple feature access - Part 1	Optional	Currently unused
HoS 2.0	W3C SOAP 1.1	Mandatory	Used in implementation (most components)
HoS 2.0	W3C Web Services Description Language 1.1	Mandatory	Used in specification and implementation (most components)
HoS 2.0	W3C eXtensible Markup Language 1.0	Mandatory	Used in specification and implementation (all components)
HoS 2.0	W3C XML Schema 1.0	Mandatory	Used in specification and implementation (all components)

The second table shows the more tool- and implementation oriented recommendations from the state of the art deliverables, specifically A3.2-D1 and A3.4-D1:

Source	***Title***	***Bindingness***	***Status***
HoS 2.0	OMG XMI	Optional	Currently unused
HoS 2.0	Java Standard Edition 1.5	Mandatory	Used in all component implementations
HoS 2.0	Java EE 1.5	Optional	Partially used (Servlet API 2.3)
HoS 2.0	Eclipse RCP	Recommended	Used in implementation (WP07 Data Modelling Tool)
HoS 2.0	Spring Framework	Optional	Used in implementation (Mediator Service, Context Service)
HoS 2.0	Hibernate 3.1	Recommended	Used in implementation (Context Service)
HoS 2.0	GeoAPI	Recommended	Used in implementation (all components)
HoS 2.0	GeoTools	Recommended	Used in implementation (Mediator Service)
HoS 2.0	GDAL/OGR	Recommended	Currently unused
HoS 2.0	W3C XSL Transformations 1.0	Mandatory	Used in testing (CST)
HoS 2.0	W3C Scalable Vector Graphics 1.1	Optional	Currently unused
HoS 2.0	INTERLIS tools	Recommended	Currently unused

HoS 2.0	UGAS (ShapeChange)	Recommended	Currently unused because of stopped development (last version is more than two year old)
HoS 2.0	Enterprise Architect 7.0	Recommended	Used in specification and implementation
HoS 2.0	Protégé	Recommended	Used in specification and implementation (Case studies for Harmonisation toolkit)
HoS 2.0	GeoNetwork	Recommended	Used in implementation (Information Grounding Service)
HoS 2.0	Micka Metadata and catalogue system, CatMDEdit, GeoMedia	Recommended	Currently unused
HoS 2.0	UMN Mapserver 4.6	Recommended	Used in implementation (Information Grounding Service)
HoS 2.0	GeoServer 1.6.0	Recommended	Used in implementation (Information Grounding Service)
HoS 2.0	Deegree 2	Recommended	Currently unused, but usage investigated for GML parsing and generation
HoS 2.0	THREDDS/OPeNDAP	Optional	Currently unused
HoS 2.0	Udig 1.1	NA	Currently unused
HoS 2.0	QGis	NA	Currently unused
HoS 2.0	Openlayers	NA	Used in Protected Areas Scenario
HoS 2.0	Mapbender	NA	Currently unused

As can be seen from the lists, almost all mandatory (16 out of 18) and many recommended and optional elements (16 out of 33) are already incorporated. In those three cases where a mandatory tool or standard is not used, it is either because the need was not yet there (GeoTIFF and WCS) or because a newer version of the standard/tool is in place, as with SLD/SE. In addition to the sources given in the Handbook of Standards, some tools and standards were directly picked up from one of the State of the Art documents, like the Service Component Architecture (SCA). These are noted in the individual component descriptions.

Please note that tools or standards that are “under observation” have been omitted from the list if there was not a specific reason for their inclusion. Also, “trivial” tools and specifications like HTML, GIF and JPEG were omitted for brevity's sake.

As a final note, the State of the Art reports provided important input in terms of the infrastructure to integrate with, not just the standards to use.

2.4. External and Internal Feedback

During the specification work done so far, there were several opportunities to gather feedback on the proposed components, both by project partners and by external colleagues.

2.4.1 Feedback from the EC Reviewers

In December 2008, the second annual review for the HUMBOLDT project took place. From that review, we received some important feedback:

- ECR01: External and internal interoperability and loose coupling have to be ensured to meet the requirements of a SOA. So far, these capabilities are not apparent in the documentation of the Framework Specification. Consequently, the means for this, such as the provision of Interface Control Documents (ICDs), have to be made explicit in the documentation.
- ECR02: In the same context, it has to be made clear that HUMBOLDT software is not a monolith, but indeed a modular system whose individual parts can be used as self-sustained services, following a SOA pattern. Towards this point, it is necessary to add a service viewpoint to the specification documents or to otherwise explain how HUMBOLDT components fit into existing SOA structures.
- ECR03: Furthermore, it is currently hard to understand the developments, therefore some effort has to be spent in better explaining them, especially to the scenarios and outside stakeholders.
- ECR04: HUMBOLDT should take a look at the final version of RM-OA and see how far it can be applied in HUMBOLDT.

This points have from our perspective been addressed with the M2.0 and M3.0 documents. For example, *Interface Control Documents* have been added to the specifications for the HUMBOLDT services. Further, the provision of a high-level, lightweight introduction to the framework (see A5.3-D3 [3.0]), explaining the software for non-technicians with basic knowledge of OGC-standards should make the framework better understandable to outside stakeholders.

2.4.2 Feedback from Associated Projects

After the release of the 1.0 specification, there was considerable interest in the Mediator Service Specification, which was provided to BOSS4GMES to give them the possibility for feedback and influencing further development according to their requirements.. Information on the Edge Matching Service was provided to NatureSDIplus, ESDIN and a currently on-going matching project between Saxony and the Czech Republic. The latter added concrete requirements to this processing service, such as the capabiltiy to maintain correct topology of point-based features in relation to surfaces after edge matching. Finally, there was initial feedback on the HUMBOLDT Alignment Editor from ESDIN. However, most of the feedback was provided on a general level.

After the release of the 2.0 specification and subsequent implementation, there was feedback from several new projects, among them NatureSDI, ESDIN and a Czech-Saxon cross-border data harmonisation project. Users of these projects tried out software, mostly HALE and the CST, and provided feature requests and reported some bugs, such as issues related to using the software on different platforms.

2.4.3 Internal Feedback and Requirements

The project internal feedback in M3.0 focused on requirements gathering from the scenarios. For finding out the requirements that the scenarios have with respect to data harmonisation, the following approach was implemented, based on the Process Specification as developed by WP04:

1. In-depth analysis of the use cases and data sets involved in the Protected Areas Fast Track Scenario and the Schema Translation Prototype for the ERiskA scenario, leading to an initial set of requirements which was used for writing the 2.0 specification;
2. With the start of the work on the 3.0 specification, analysis workshops with each of the other scenarios, including the gathering of data sets to be transformed, were conducted. These workshops had the goal of resolving the hen-egg problem ("tell me what your software can do, and I will tell you what we can use it for") by explaining the current scope of the software and checking what modifications would need to be done to make the approaches selected for EriskA and Protected Areas work in the other scenarios;
3. Collection of requirements in an on-line tool based on the VOLERE templates, which help in creating high-quality, verifiable requirements;
4. Organisation and execution of a requirements consolidation meeting with the full architecture team and representatives of seven scenarios, which led to a set of eight common requirements clusters and about 40 common, high-priority requirements;
5. After this process, the set of common requirements was passed on to all component owners. Each component owner then updated the specification to reflect additional requirements that came up in this process, and provided information on which requirement led to which decision in the specification;
6. New requirements that were identified after the consolidation meeting are handled using the Change Request process as described in "A4.3-D2 Report on Process specification evaluation and improvement (Final Version)". Most of these change requests were concrete, detailed feature requirements to a component already in development, and therefore these feature requests were directly given to WP08 for implementation.

2.5. Summary of Objectives

The objectives as described in this chapter can be summarized into two categories:

1. Align the specification work with HUMBOLDT scenario requirements on data harmonisation
2. Consider the requirements and objectives already present for M2.0 in the new specification work and the new framework features as derived from the scenario requirements.

3 Approach for Specification

The technical work in HUMBOLDT is divided into specification, development and technical validation. Three work packages were defined at the beginning together with major milestones that can be found in the Description of Work. However several changes have occurred meanwhile due to the experienced gained during the last two years of work, and the procedures and infrastructure have been improved to better use the resources and to obtain further results.

The first development phase was based on a technology driven approach – several technologies and methodologies were tested, We are now moving to a scenario driven approach, where the harmonisation requirements coming from scenarios will play a major role in the technical work.

For this purpose, we define short milestones (monthly based) both for specification and implementation, so we can efficiently incorporate the harmonisation requirements of the HUMBOLDT scenarios.

3.1. Application of the RM-ODP and RM-OA

The underlying methodology for software development in HUMBOLDT is RM-ODP. Being both a distributed project in terms of the types of actor groups involved and having the goal of developing software for a distributed infrastructure, namely the European SDI, adopting the RM-ODP seemed like a good idea.

The RM-ODP family of recommendations and international standards defines essential concepts necessary to specify open distributed processing systems in an incremental manner and is focused on five prescribed viewpoints. It provides a well-developed framework for the structuring of specifications for large-scale, distributed systems.

The architecture of HUMBOLDT is defined as a set of components, connections and topologies described through the series of views proposed by the RM-ODP. The HUMBOLDT framework will have multiple kinds of users and developers and each group will view the system from their own perspective. The picture below shows schematically the aforementioned viewpoints and their relation to typical artefacts of an IT system definition.

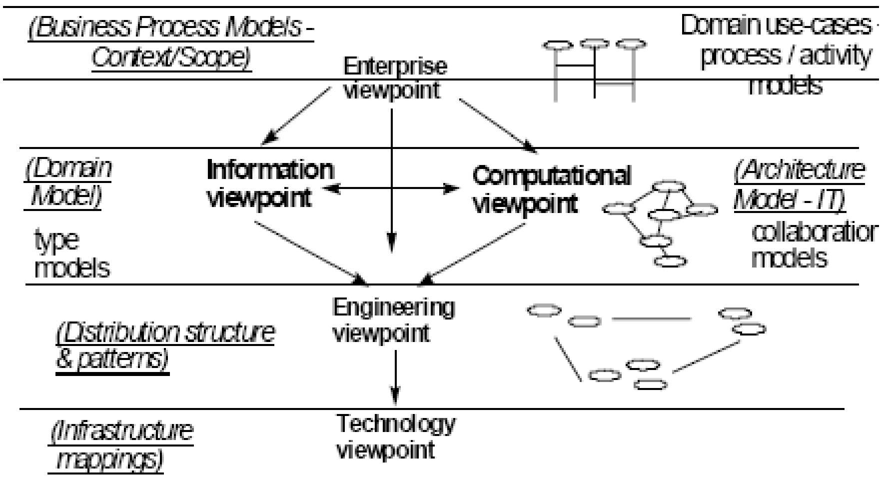


Figure 1: Viewpoints of the RM-ODP and their relationships

The RM-ODP standards have been widely adopted: they constitute the conceptual basis for the ISO 19100 series of geographic information standards (normative references in ISO/DIS 19119); they have also been employed in the OMG Object Management architecture and they are used as the basis of the OGC reference model.

Viewpoint Name	***Definition of RM-ODP Viewpoint (ISO/IEC 10746-1:1998)***	***How viewpoint is addressed in the HUMBOLDT framework***
Enterprise Viewpoint	A viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies for that system.	This viewpoint describes the functional purpose of the framework and what is tried to accomplish by it. It covers the information on typical business processes to be supported by the usage of the HUMBOLDT framework and scenarios as well as the use cases prepared for the HUMBOLDT development. The viewpoint is both intended to be read by the developers for gaining understanding of the business aims and also by the stakeholders of the system to be developed to verify that the system will fulfil their actual requirements.
Computational Viewpoint	A viewpoint on an ODP system and its environment that enables distribution through functional decomposition of the system into objects which interact and interfaces.	This viewpoint shows an overview of the composition of the components in the framework, including their internal modules, their collaboration with other components and the interfaces used for that purpose. It describes the logical architecture and the descriptions of individual components of the framework.
Information Viewpoint	A viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing.	This viewpoint covers the domain model of the framework. The domain model consists of various data structures which are being used for storage and exchange between the service components defined in the computational viewpoint. These may include the domain model of the component as well as specific message structures which it exchanges with other components via their public interfaces.
Engineering Viewpoint	A viewpoint on an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.	This viewpoint is focused on the overall technical architecture of HUMBOLDT, the deployment and integration aspects of the framework like computers, networks etc. It also deals with the distribution of the framework and deployment, security and persistence aspects.
Technology Viewpoint	A viewpoint on an ODP system and its environment that focuses on the choice of technology in that system.	This viewpoint presents the technology selection on HUMBOLDT. It describes the hardware and software components used; a selection of technologies such as libraries and protocols to use takes place.

Table 2: Usage of the RM-ODP viewpoints in HUMBOLDT specification work

Table 2 shows how the RM-ODP viewpoints are used in the description of HUMBOLDT architecture.

The use of RM-ODP in HUMBOLDT provides us with some core advantages:

- It provides a set of already established reasoning patterns to help us specify and design our system. Those patterns assist us in identifying the fundamental entities of the system and the relations among them. In this sense, RM-ODP encourages us to ask the right questions with the appropriate degrees of abstraction and precision for building useful specifications.
- It helps us thinking from different perspectives (or viewpoints), thus supporting the HUMBOLDT requirement collection and analysis phases of the development.
- It offers an infrastructure and a common reference model within which different requirements expressed in separate languages (those from the viewpoints) can be consistently integrated.
- It provides HUMBOLDT system designers and developers with a set of mechanisms to facilitate their jobs, together with a technological infrastructure that supports the model.
- It provides facilities for building robust, efficient and competitive applications, that will be interoperable with other systems that conform to the same standards, and it is sufficiently backed by industrial products.

RM-OA is the Reference Model for the ORCHESTRA Architecture – an adaptation to RM-ODP developed in the context of the European project ORCHESTRA. In the RM-OA, the viewpoints are refined and in the case of the computational viewpoint extensively modified to *"highlight the fact, that an ORCHESTRA deployment will have the nature of a loosely-coupled distributed system based on networked services rather than a distributed application based on computational objects"*. (RM-OA v2.1, p. 34)

In the HUMBOLDT project, we allow for both perspectives: On the one hand, there is the Framework itself, which is intended to be a toolbox of reuseable software components, like the Spring Framework or Geotools. On the other hand, there are application deployments built on these framework components. Most framework components can also be deployed as services by themselves.

The main sustainable project outcome is the HUMBOLDT service integration framework and the harmonisation toolkit applications built from its components. The Framework makes it very easy to construct different kinds of simple or complex (web) services, but the Framework is not in itself a bunch of services. Rather, it is a set of components with single-way dependencies that can be partially or completely used to quickly construct a harmonisation (web or otherwise) service.

With this intent, the RM-OA cannot be fully applied to the Framework specification part of our project. We have, however, come to the conclusion that many of the issues that occur when applying the RM-ODP on a SOA project are also valid for us and that some of the solutions given with the RM-OA are very valuable. Therefore, we have taken over the following elements from the RM-OA into our specification procedure:

- Usage of UML as a basis for the implementation of a Model Driven Architecture;
- Splitting of specifications into an abstract, contractual part and into a reference implementation/TCK part;
- Usage of Interface Control Documents or similar formalized service interface descriptions;
- Provision of a Service Viewpoint as part of the computational viewpoints, especially in all scenario specifications.

In addition, some of the high-level requirements identified in the RM-OA do also apply in HUMBOLDT (please refer to section 5.2.1.6). Other parts could not be used since the goals of the ORCHESTRA project, as reflected throughout the Enterprise Viewpoint of the RM-OA [RM-OA, section 6.2.1 pg. 43], are different from the goals of the HUMBOLDT project.

3.2. V1.0 Experiences

One major input to the specification work done in the last phase of the project was the experiences gained from specifying and implementing work in the v1.0 phase. The main achievements were the specification of a framework that consisted of several components and that had the capability to perform simple geoinformation harmonization mostly on a syntactical and geometric level as well as the implementation of several of the specified components, specifically the Information Grounding Service, the Context Service and the Mediator Service.

3.2.1 Process Experience gained

During this last period, a problem was a lack of communication between the technical team and the scenarios; due to this we have tried to establish a more close cooperation between the Architecture Team and the Scenario Team. Hence, a complete methodology – processes and tools – was developed in WP4 in order to define how to influence the developments.

Additionally clear contact points for both teams were identified, this has end in a better collaboration where direct means of communication have been established.

Regarding the specification processes, for M1.0 we switched to a one-document-per-component approach, plus a central document, with one responsible editor and a team of contributors per document. This has been kept since it has proved to be a good approach. Nevertheless we saw that the specification process should be more visible to participants during all the process – and not only at the end –, the solution was to use a wiki to compile all contributions so anyone at anytime could check what was going on.

We also noted that in previous stages the specification was not well understood by the scenarios. To overcome this issue we have used concrete examples based on scenarios specifications, written in such a way that it is understandable to all stakeholders, and making clear what the added value is in each one of the components.

A clarification in the scope of the work of WP8 and WP5 was also carried out, WP5 was giving too many details about the implementation in the past, now some of these concrete details have been transferred to WP8, such as description of algorithms, and the technology and engineering viewpoint of the RM-ODP document.

3.2.2 Architecture Experience gained through V1.0 specification

The requirements received during the 1.0 specification phase could be satisfied within the defined architecture, which consequently underwent only minor evolution. The main point learned in this phase was that the specification work package has to provide better definitions of how components behave as services, i.e. describe their interfaces in a formal way, such as by providing a WSDL and/or XSD consistently.

3.2.3 Technical Experience gained through V1.0 implementation

The V1.0 implementation has demonstrated the strengths of using the spring framework. Currently both the mediator and context service components have been completely developed using this JEE framework. This framework has provided the following core benefits:

- Improved speed of development as less class integration code is required to be developed;
- Simplified testing at the unit and integration level;

- Springs flexible configuration allows classes and components to be replaced simply without code changes. This will be of great benefit in version 2.0 as we replace some of the components developed in version 1.0 to WSDL;
- The contract first approach mandated by Spring WS for the WSDL definition ensured greater language interoperability.

Going forward for version 2.0 two frameworks which may be evaluated to improve the configuration and integration of the framework are:

- *CFX*, enables the development of services. These services can use a variety of protocols such as SOAP, XML/HTTP, RESTful HTTP, or CORBA and work over a variety of transports such as HTTP, JMS or JBI. This will allow the Humboldt framework to be more loosely coupled and therefore provide greater usage and integration possibilities to its future users. This framework is also spring compatible;
- *Metro*, Metro is a high-performance, extensible, easy-to-use web service stack. This framework can be integrated into Spring and is well known for its strengths in cross language interoperability.

3.3. V2.0 Experiences

The specification work for version 2.0 has shown a substantial improvement in the work with application contacts from the HUMBOLDT scenarios. Most of the decisions taken in specification work have been driven or confirmed by concrete scenario requirements. The work on desktop applications such as the HUMBOLDT Alignment Editor has proven to be a good way to get more concrete feedback from HUMBOLDT user groups and to be able to communicate the project results more easily.

However, there were still some issues identified that needed improvement and which have been addressed in the work for M3. For example, it has been realised that the specification was still too complex to be easily understandable. In M3, this has been addressed by providing more explanations in the individual specification documents, aligning the examples to the *end-to-end example* given in chapter 4 of this document, as well as providing a lighweight introduction to the framework (see: A5.2-D3 [3.0]). Further, the specification documents have been updated according to input from the implementers in WP8.

4 An End-to-End Example of using the HUMBOLDT Framework

This chapter outlines the capabilities of the HUMBOLDT Framework and it shows the benefits that applications built on it can provide to different user groups (data custodian, data integrator, end user of geodata). . For this, a use case from the Protected Areas Scenario is introduced and for each user group this common scenario is unfolded iteratively.

4.1. Introduction to the Protected Areas Scenario

In the Protected Areas scenario, there are three functional goals that the organizations and people working in it want to achieve and which are in their characteristics typical for most HUMBOLDT scenarios:

1. Improved tourism valorisation of the protected areas;
2. Improved protected areas management;
3. INSPIRE-IR conforming data provision and reporting.

4.1.1 Goal 1: Improved tourism valorisation

To reach the first goal, several different, sometimes conflicting views have to be merged: On one hand, allowing the touristic exploitation of an area can have big advantages, such as making people understand why a certain area needs to be protected, but also to ensure a sustainable income for people living in the region of the protected area.

One the other hand, some areas are highly sensitive and should not be disturbed at all. Balancing the interests of nature conservation and touristic exploitation, which has to include making an interesting offer, therefore has been a topic since the first national parks and other protected areas have been introduced. To make good decisions and to offer interesting experiences to tourists, a wide range of different data need to be taken into account. These include information on protected areas and special protection zones, on flora and fauna habitats, on topography and hydrography and of course on touristic infrastructure and points of interest. All these different data sets have been accumulated over a manual process in the past, and have essentially been thrown together, with different problem occurring as a result of that. When the data sets are available, interesting and sustainable hiking paths and other touristic infrastructure can be planned and selected.

4.1.2 Goal 2: Improved protected areas management

The second goal is based on the creation of a development plan of the regional park and to guaranteeing a seamless operational functionality between the park and its communes. Compared to the first goal, which is concerned with planning of a specific activity within the park, this goal therefore also includes collaboration with surrounding areas. It has much in common with the first one with respect to the required data sets, but has a broader focus – the big picture of protected areas management is required, which involves working with lots of different information sources, setting up reports and monitoring on-going developments and projects. In this framework, the assigned objectives –- as far as management of geo-information is concerned – are:

1. to create a web access where all geo-information of the park (to be previously assessed and systematised) is shareable, also with the various external administration and decision levels to ensure that everybody bases decisions on the same data, which is most up-to-date
2. to make the municipal plans technically compliant with the provisions of the regional government
3. to create conditions for a good communication towards citizens
4. to train people working in the park and in the communes to use properly the created tools

4.1.3 Goal 3: INSPIRE-IR conforming data provision and reporting

The third goal, the INSPIRE-IR-conforming data provision, has a different focus. Here, the protected area agency is not providing data for its own use, but rather for the use of others partaking in a common European Spatial Data Infrastructure. This has many benefits such as making the data collected here better accessible to the public, to researchers and to decision makers. To achieve this goal, the agency needs to provide the data that originates from its activities, such as information on species distribution and on habitats, via standardized services and using harmonised application schemas. For this purpose, a set of application schemas on protected sites, species distribution and other aspects have been or will be drafted by INSPIRE Data Specification Drafting teams. The agency now needs to ensure that data it collects and that falls within one of the themes defined by the INSPIRE directive is made available for View and Download services, again according to the regulations of the directive.

4.2. The domain business process and the Actors in it

All three goals can be reached through improved geodata integration and harmonisation and the provision of a scenario-specific information system. These points can all be supported by HUMBOLDT software. How this happens and what different people can do to use HUMBOLDT software is explained by providing an End-to-End set of user stories.

This set of user stories describes the task of creating a web portal that can be used by hikers for identifying suitable hiking routes. An important goal of the portal project is to keep the portal independent from the data sources of one region in order to enable the reuse in different areas, such as in between Spain and Portugal. Another technical goal is to enable users to employ OGC-conforming clients for retrieving data from the portal. Hence, the portal server must offer standardized OGC-interfaces for data access, such as WFS or WMS.

The aim of this example is to show how the task of setting up this portal and using it is achieved using the HUMBOLDT components. The two other goals are also referenced where necessary, explaining the common methodology which can be used to satisfy all three goals.

Within the scenario description, the data integrator is embodied by Luigi, a programmer and IT expert. Luigi is responsible for maintaining the IT-infrastructure of the protected areas management agency. The data custodian is represented by Carla, an expert in geospatial data models and – together with Luigi - responsible for the data sets involved. Finally, the end user of geodata is embodied by Mario Rossi Buhl, a regional officer at the Territorial Planning Department of an Italian region. Mario is the person responsible and with the initial need for the web portal on hiking paths. He is supported by Luigi and Carla in achieving his task.

Before the portal can built, the data sets involved in the calculation need to be identified. In order to be able to do this, Mario makes the methodology for the calculation of the hiking routes explicit, which is described as follows:

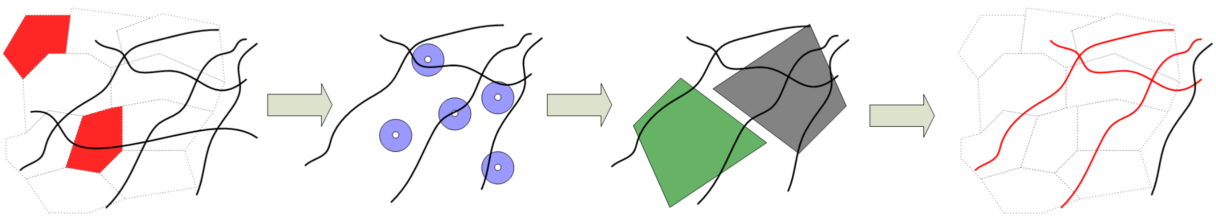
- The first and most important condition is that the hiking routes shall lead through the protected areas managed by the protected areas management agency, Mario’s employer. The goal is to make hikers familiar with the preservation of nature and to improve the income of the people living in that area.

- Further, the hiking routes shall lead close to stopping places, e.g. places for overnight staying. The potential hikers using the portal should have the possibility to define/quantify the term “closeness” individually when using the portal.
- Additionally, there are some special protected areas that should not be entered by humans and therefore be avoided by the hiking paths.
- Finally, the delivered data on the hiking paths should include some information on the area that is crossed, such as forest or wood.

Based on these requirements, the following minimally required data sets involved can be identified.

- Footpaths and Hiking Trails: This data set delivers information on potential hiking paths.
- Protected Areas: This data set delivers both, data on protected areas that should be crossed by the hiking paths, as well as data on special protected areas that must be avoided by the paths.
- Stopping Places: This data set delivers data on stopping places such as places to stay overnight or panoramic places.
- Vegetation: This data set delivers data on the vegetation coverage, such as forest, wood or rocks.

Based on the methodology and the data sets involved, an abstract chain of processing steps can be identified for calculating the paths. This chain is shown in the Figure below.



(a) (b) (c) (d)

Figure 2: Methodology for calculating the hiking paths

First, those areas are selected that should not be crossed by the paths (a) and the paths not crossing such areas are identified. Then, a buffer is calculated around stopping places and only those hiking paths are selected, that are close to (at least one) of them (b). Further, the information on vegetation is attached to the hiking paths in (c) and finally, only those hiking paths are selected that cross protected areas (d).

Note that the identification and setting up of an application processing chain is of less importance for the second and third goal. For these goals, the identification of the required data sets can be done by directly checking the harmonised data model. The INSPIRE-IR conforming data provision usually does not require application specific processing, even though the generation of reports might benefit from these capabilities.

4.3. Setting up the Software

The aim of this section is to show how the HUMBOLDT framework can be used for building the web portal described in the previous section. Before most of the HUMBOLDT components can be applied and the web portal can be deployed, there are several preconditions. Some are concerned with setting up the infrastructure (web services, HUMBOLDT components) and some with preparing the data sets involved for further processing.

4.3.1 Setting up the infrastructure

Involved HUMBOLDT User Groups: Data Integrator (Luigi)

Involved HUMBOLDT components: Information Grounding Service (IGS), Workflow Design and Construction Service (WDCS)

Luigi, the data integrator, is responsible for maintaining the IT infrastructure of the protected areas management agency. For doing this, Luigi has to accomplish several tasks. First, he downloads and deploys all HUMBOLDT components necessary for building the web portal on hiking paths.

Further, all of the involved data sets need to be published via OGC compliant interfaces, e.g. WFS in the case of vector-data, WCS for raster etc. and they must be registered to some geospatial catalogue. These two tasks are outside the HUMBOLDT framework itself and do not involve HUMBOLDT components.

For being able to apply the framework and to profit from its capabilities, Luigi registers the individual catalogues that hold the metadata of available data sources to the Information Grounding Service (IGS). The IGS is the HUMBOLDT component responsible for the discovery of geospatial data services. It does not replace existing catalogues but serves as a catalogue of catalogues and is therefore capable of discovering a huge number of different data sets, e.g. those covering different administrative areas and registered to different catalogues. Furthermore, it can make use of thesauri and defined mappings between different application schemas (i.e. a mapping linking types from CORINE to another land cover/use classification scheme) to perform search broadening.

Furthermore, the processing components required for calculating the hiking paths (such as the Intersection operation) need to be registered to the HUMBOLDT Workflow Design and Construction Service (WDCS). Since most of the processing components involved in the calculation of the hiking paths are well-known and widely used such as Intersection and Buffer, they can be reused from already existing applications and do not need to be implemented again. HUMBOLDT enables this reuse by allowing users to register geoprocessing functionality implemented within OGC Web Processing Services to the WDCS.

4.3.2 Schema Definition and Schema Mapping

Involved HUMBOLDT User Groups: Data Custodian (Carla)

Involved HUMBOLDT components: Model Editor, Alignment Editor, Model Repository

Since the processing chain identified in the previous section includes some processing that can only be applied to data that adheres to a data schema known at the time of building the processing chain (e.g. selection based on non-spatial attributes), the data schemas of all data sets need to be mapped to a (previously defined) integrated schema.

For example, the existing data services that deliver data on Protected Areas use different and heterogeneous data schemas. In order to be able to process all the different heterogeneous data sets, an integrated Protected Areas schema (an extract of which is shown in Figure 3) needs to be defined and all individual source schemas need to be mapped to that integrated schema. This schema mapping enables –- at runtime –- the automated translation of data instances (e.g. GML instance files) from the source to the target schema and the subsequent data processing with respect to the integrated target schema. For defining the integrated schemas, Carla employs the HUMBOLDT Model Editor, a UML-based editor enriched with geo-specific constructs.

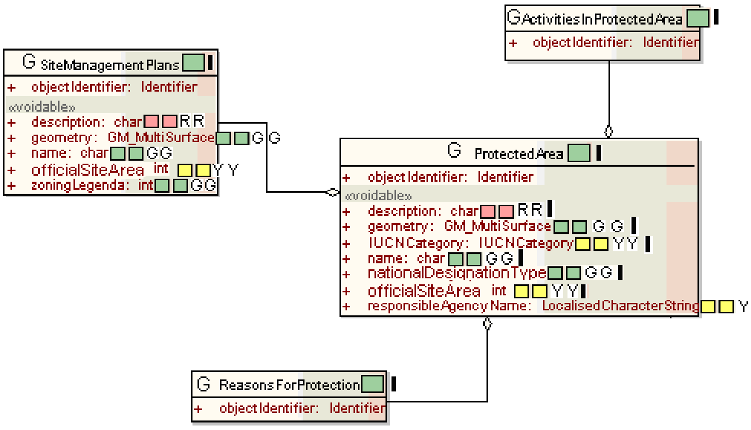


Figure 3: Extract of the Protected Areas Integrated Schema

The component which can be used by Carla for the definition of mappings between a source and a target schema is the HUMBOLDT Alignment Editor (HALE), shown in Figure 4. Carla employs HALE for mapping the individual data schemas to the integrated ones. Finally, she stores the newly defined schema and the mapping definitions in the HUMBOLDT Model Repository, where they can be accessed by anyone who needs them and is allowed to access them.

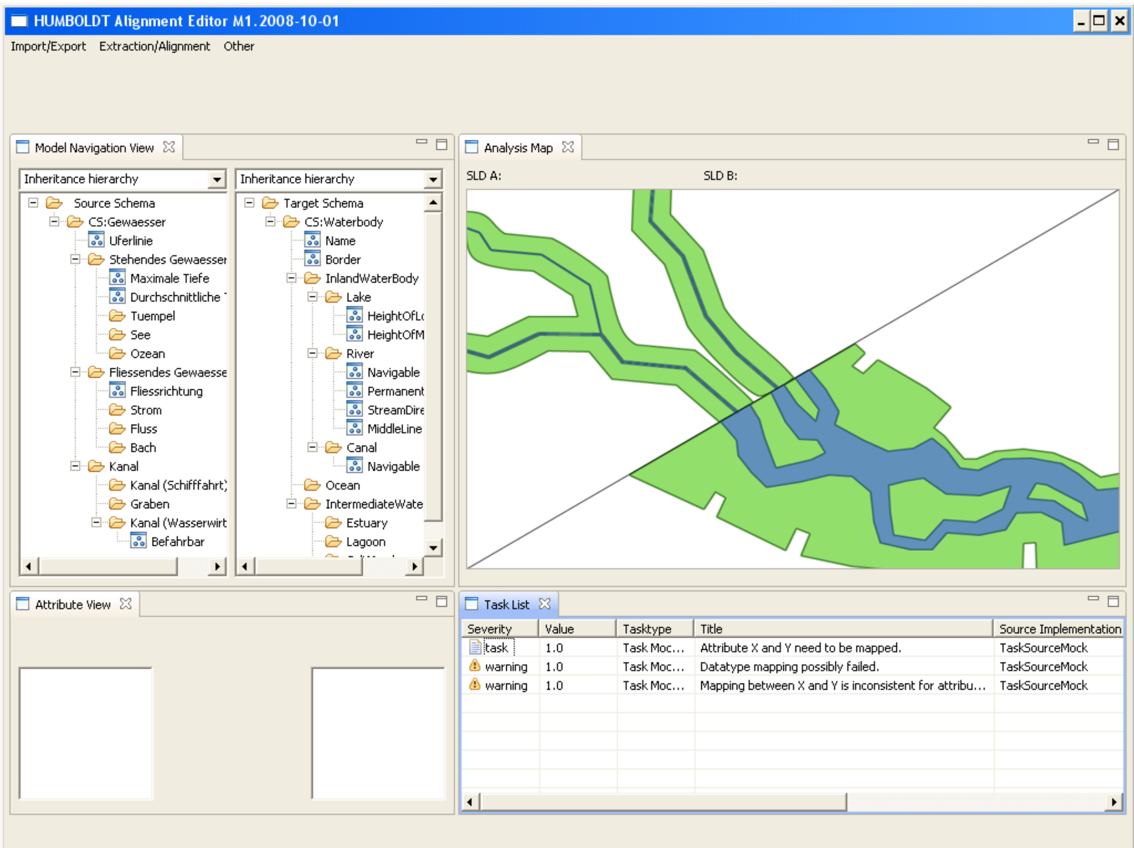


Figure 4: The HUMBOLDT Alignment Editor

Later, the mappings are loaded by the conceptual schema transformer, a Web Processing Service that uses the information stored in them to transform geodata instances from a source schema to the integrated target schema.

4.3.3 Definition of the application specific processing chain

Involved HUMBOLDT User Groups: End User (Mario), supported by Data Integrator (Luigi)

Involved HUMBOLDT Components: Workflow Design and Construction Service (WDCS), WDCS GUI, Geospatial Processing Components (e.g. WPS) registered to HUMBOLDT

The abstract chain of processing steps shown in Figure 1 can be transferred to a chain of concrete geoprocessing functionality. This is done by Mario, using the graphical user interface of the Workflow Design and Construction Service (WDCS), called the Workflow Frontend (WF). The WF enables him to connect the processing components registered to the system (e.g. external WPS) into a chain whose execution calculates the hiking paths according to the functional methodology for hiking routes calculation explained in the introduction to the scenario. The resulting chain of geoprocessing functionality is called a Basic Workflow in HUMBOLDT terminology. Figure 4 shows the input/output signature of the Basic Workflow "Sustainable Hiking Paths", abstracting from the concrete chain of processing.

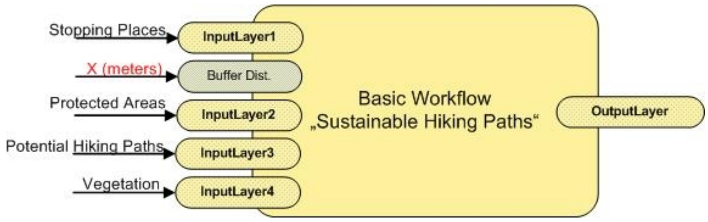


Figure 5: Abstract view on the Basic Workflow

The Basic Workflow Sustainable Hiking Paths is abstract in the sense that it is data independent. This means it does not yet hold information on concrete data services that deliver input. Instead, it contains constraints on the potential input data sets, such as the schema constraints. This means the same methodology can be used both in Italy or in Portugal and Spain, and needs only to be defined once. This becomes especially valuable when thinking of applications of a European scale, with possibly hundreds of different data sources.

During execution, concrete data download services (also called grounding services) that satisfy the input constraints are automatically discovered and attached to the workflow, resulting in an Executable Workflow.

The names of the layers that serve as input to the workflow such as the name Protected Area are the names of FeatureTypes from the integrated schemas to which all individual data schemas have been mapped previously using HALE. This mapping is necessary since –- although most of the processing components within the workflow are schema-independent and operate on (GML) geometries –- there can be schema specific processing steps encapsulated within the workflow. For example, there is a selection of features based on a non-spatial attribute of the Protected Area schema. This selection is only possible if the data on protected areas that is used as input to the workflow adheres to the integrated schema for protected areas.

Finally, Mario registers the Basic Workflow he created to the HUMBOLDT system as a new extension to the Protected Areas integrated application schema as the Feature Type "Sustainable Hiking Paths". The Basic Workflow is stored within the WDCS workflow repository and can now be requested as any other Feature Type, using a WFS compliant client such as OpenLayers.

As stated before, the provision of INSPIRE-IR compliant data usually does not require the definition of an application specific processing chain but is more concerned with schema mapping and other aspects of harmonisation than the application-specific usage of the resulting integrated data.

4.4. The application in action: Retrieving "Sustainable Hiking Paths"

Involved HUMBOLDT User Groups: End User of Geodata / Geoinformation (Mario), Data Custodian (Carla)

Involved HUMBOLDT components: Workflow Design and Construction Service (WDCS), Mediator Service (MS), Context Service (CS), Information Grounding Service (IGS)

Features of the Feature Type Sustainable Hiking Paths can now be retrieved by end users of geoinformation. Every time such request is issued, the workflow is executed. Since the Basic Workflow is data independent, it can be reused by different user groups with different specific needs on the data to be returned, such as specific areas or spatial reference systems. The set of concrete end user constraints on the data to be returned is called a Product Definition.

4.4.1 Product Definition

Involved HUMBOLDT User Groups: End User of Geodata / Geoinformation (Mario), Data Custodian (Carla)

Involved HUMBOLDT components: Context Service (CS), Context Service Client (CSC)

For setting requirements, Mario accesses the Context Service by using the Context Service Client. Since the web portal delivers the data via standardized OGC interfaces, only those constraints on the data to be returned that are allowed within standard OGC requests can be directly passed to the system. This does not suffice in many cases, where not only simple portrayal products need to be returned, but rather complex transformation products. The aim and raison d'être of the Context Service is therefore to allow Mario to set additional constraints on the data to be returned that cannot be set within all standard OGC requests. A context example is shown in Table 3 and explained in the following.

Constraint Type	***Constraint Value***
Feature Type:	Integrated Protected Areas->Sustainable Hiking Paths
Spatial Reference System:	WGS84
Bounding Box:	{ …,… / …,… }
Language:	Italian
Quality:	Data must have been collected within the last five years.
…	…

Table 3: Example for a user’s context definition

The Feature Type requested is Sustainable Hiking Paths, which represents the Basic Workflow created by Mario beforehand. The data to be returned should adhere to the reference system WGS84 and additionally cover a certain area represented by the bounding box constraint. Further, the requested language is Italian and the data sets involved must have been collected within the last five years. While the Feature Type, the SRS and the BBox can be set within standard OGC requests, the language and quality constraint can not.

The same approach for defining a transformation product is also suitable in the case of the second and third goals. In INSPIRE, precise target descriptions are already being created, including many parameters that are also represented in the context, such as the schema description, quality parameters, areas of interest and others. Similar to the Basic Workflow definition, a product definition can be re-used simply by changing e.g. the spatial constraint. This means that a single product definition can serve as a product template for the entire ESDI.

4.4.2 Workflow Construction and Execution

Involved HUMBOLDT User Groups: End User of Geodata / Geoinformation (Mario), Data Custodian (Carla)

Involved HUMBOLDT Components: Mediator Service, Context Service, Workflow Design and Construction Service, Information Grounding Service

After defining the context, Mario requests – using an OGC-conformant client – the data. The component responsible for handling the request is the HUMBOLDT Mediator Service.

Within the Mediator Service, the request is merged with the context, retrieved from the Context Service. The set of constraints of request and context can contain common elements, e.g. the Feature Type requested. In this case, the request-constraints override the constraints set within the context. For example, the bounding box constraint Mario sends within the direct request overrides the bounding box constraint previously set within the context.

Based on the user request and the context, the Basic Workflow Sustainable Hiking Paths is retrieved and enriched with request specific parameters, e.g. the specific bounding box and spatial reference system requested by Mario.

1. Discovery and Automated Harmonisation

The input descriptions of the Basic Workflow are now passed as a discovery query to the Information Grounding Service. The IGS returns –- for each single input to the Basic Workflow such as "Protected Areas" –- a pointer to a number of data services (e.g. WFS) that can deliver data.

But since Mario requested data for an area that crosses the boundary between two countries and that is therefore not served by one single data service, only a spatial combination of the two data sources covers the area requested by the Mario. Figure 6 shows the two data services discovered for Protected Areas.

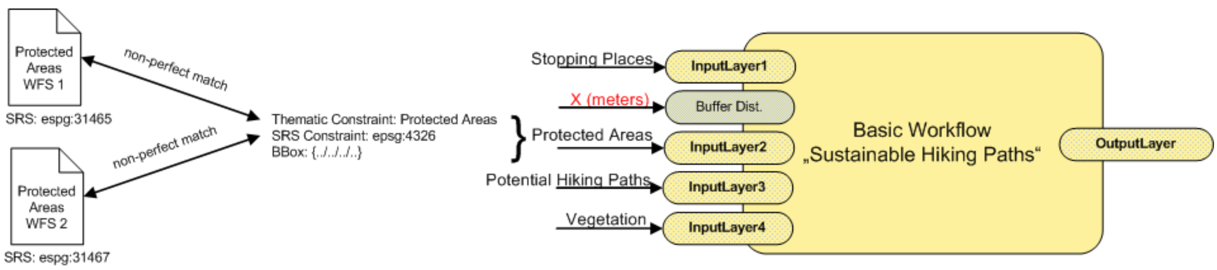


Figure 6: Discovery of Workflow input

The harmonisation requirements and the solution that is applied are shown for the Protected Areas layer in the following. The two data sources discovered (Protected Areas WFS1 and WFS2) can –- if spatially combined –- deliver input data to the Basic Workflow such that the output covers the area requested by Mario. However, both WFS deliver data within different reference systems since they are maintained by different data providers from different countries. Moreover, the method of data acquisition is different and therefore, the two data sets differ in precision, which is directly visible on the boundary of the two regions, as shown in Figure 7.

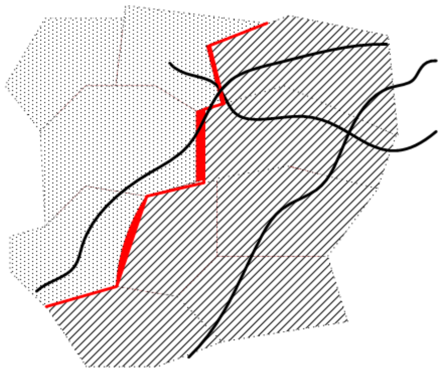


Figure 7: Harmonisation requirements

Hence, before both WFS can serve as input to the Basic Workflow, they require harmonisation of spatial reference systems as well as the alignment of their common boundaries, a process known as Edge Matching. Based on the metadata of the discovered services WFS1 and WFS2 and the constraints on the input to the Basic Workflow representing “Protected Areas”, the required harmonisation transformations are automatically identified and attached to the Basic Workflow. Additionally, the data delivered by both WFS is transformed to the integrated target schema for protected areas, based on the predefined schema mapping. The workflow resulting from the automated insertion of harmonisation transformers is shown in Figure 8.

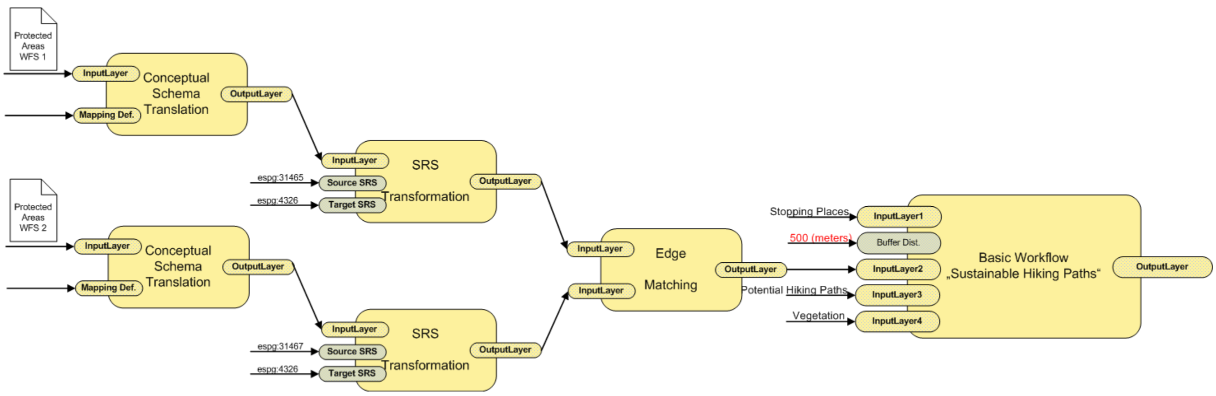


Figure 8: The final executable workflow

The automated process of adding harmonisation transformations to an application specific processing chain is called workflow construction and is performed for every single input (except the Buffer distance, which is delivered by the user) to the Basic Workflow if required.

This automated harmonisation takes place within the WDCS. When finished, an executable workflow description is returned to the Mediator Service. After executing the workflow, the Mediator Service returns the result to Mario.

Finally, Mario uses the print function of his client application and prints the hiking routes for his next field trips to verify the selection. He enjoys that part of the work most and is happy that the improved geodata integration facilitated by HUMBOLDT software allows him to start earlier with the hiking than planned.

The client can of course also be another downstream service, as it will often be the case with other scenarios and also for the third goal, the INSPIRE-compliant data access provision. However, the same approach of (partially) automated analysis and fully automated data harmonisation processing can be used for these goals, once the initial set-up steps are completed.

5 Specification Overview

This section contains the "what" of specification, focusing on two aspects: The business perspective and an overview of the whole framework and applications and services built on it.

This means that besides the computational viewpoint dealing with the logical architecture of the framework, it also contains an overview to the viewpoints, such as the enterprise viewpoint, which states the value proposition of the approach and shows usages in relation to the scenarios, and the Information Viewpoint, which gives a high-level overview of the most important domain models used.

5.1. Introduction

As data harmonisation is a complex activity that is required in many different application domains, HUMBOLDT software has a multitude of things to accomplish. Amongst those are the handling of many different types of data for transformation and presentation in very different environments and processes. These environments and processes are concretized more and more as the scenario requirements are identified.

In addition to Version 1.0 of the specification, where the main aim has been the creation of the basic infrastructure needed for the "technical harmonization process", this initial scaffolding has now been filled with concrete transformation services.

5.1.1 Introduction to Data Harmonisation

In this section, a view of the different problems that present themselves in geodata harmonisation is given to provide a perspective for the approach described later. It is based on the WP07 deliverables A7.0-D1/D3 and A7.1-D1, where a lot of analysis work was conducted that helped understanding the issues and possible solutions for them.

One starting point that we have been using multiple times was the classification of harmonisation requirements as used by INSPIRE. In this classification, there are several quite different and not always completely independent aspects:

ID	Topic	Subtopic	Explanatory notes
B	Terminology		General terms and definitions in all INSPIRE data specifications shall be drawn from the INSPIRE Glossary. Terms that are important in the context of a theme, but which are not part of the INSPIRE Feature Concept Dictionary (see 9.3) shall be defined in the INSPIRE Glossary.
D	Application schemas	ISO 19109 General Feature Model	Application Schemas are based on the General Feature Model in ISO 19109 and also contain information on the life-cycle of spatial objects (clause 9.7). Cross-theme schemas and schemas that are reusable in more than one theme are also described. As a conceptual Schema Language, UML 2.1 is to be used.
		ISO 19126/19135 INSPIRE Feature Concept Register	Application schemas and feature catalogues are complemented by a so-called feature concept dictionary register that is used to manage names, definitions and descriptions of all spatial object types that are used in INSPIRE application schemas / feature catalogues. The feature concept dictionary is one of the instruments for the cross-theme harmonization of concepts in INSPIRE and is maintained in at least in the official European languages, too.
		ISO 19110 Feature Catalogue	Feature catalogues contain a large subset of the application schema information. The feature catalogue plays three important roles: 1. It supports the styling of the application schema information into a human readable, textual presentation; 2. Feature catalogues are translated at least into all official languages of the European Union (the application schema is managed in English only); 3. Published via a registry service, it allows queries on and access to the individual elements in the application schema – both by human users via a portal as well as by software. For example, it allows direct access to the name and definition of an entry in an enumerated value in all supported languages.
E	Spatial and temporal aspects	ISO 19109 8.7 spatial object type with a value that is a spatial geometry or a topology; ISO 19109 8.9 spatial object type with a geographic identifier in a gazetteer.	The value domain of spatial properties should be restricted to the Simple Feature spatial schema as defined by OGC document 06-103r3 (Implementation Specification for Geographic Information - Simple feature access - Part 1: Common Architecture v1.2.0) whenever feasible. The specification restricts the spatial schema to 0-, 1-, 2-, and 2.5-dimensional geometries where all curve interpolations are linear.

ID	Topic	Subtopic	Explanatory notes
		spatial object type with a value that is a coverage function	An application schema package that uses coverage functions shall follow the rules of ISO 19109 8.2.5 for referencing standardized schemas, i.e. import the coverage schema specified by ISO 19123.
		spatial object type with a value that is a temporal geometry or a temporal topology; properties of the spatial object type with a value that is one of the basic types Date, DateTime and Time	Time is a dimension analogous to any of the spatial dimensions. Like space, time has geometry and topology. A point in time occupies a position that can be identified in relation to a temporal reference system. Unlike space, however, time has a single dimension. See ISO 19108 5.2
F	Multilingual text		For all geographical names and exonyms the support for multilingual text in the INSPIRE application schemas shall be considered.
G	Coordinate referencing – units of measurements	Coordinate reference systems	Systems for uniquely referencing spatial information in space as a set of coordinates (x,y,z) and/or latitude and longitude and height, based on a geodetic horizontal and vertical datum.
		Temporal reference systems	Temporal reference systems shall be described using the model specified in ISO 19108 5.3
		Units of measurement	Units of measurements shall be described using the model contained in ISO 19136 D.3.13. This model follows SI and ISO 1000 in supporting both base units and derived units.
		Geographical grid systems	Harmonised multi-resolution grid with a common point of origin and standardised location and size of grid cells
H	Object referencing modelling		If an application schema intends to allow that the spatial or temporal location of a spatial object may be provided directly, use a reference to a base spatial object or by provide a geographic identifier. Depending on the specific semantics and operational context of a spatial object type, this pattern may be adapted.
J	Portrayal		OGC Symbology Encoding

ID	Topic	Subtopic	Explanatory notes
K	Identifier management		Every spatial object type of Annexes I and II of the INSPIRE Directive shall receive a property of type “Identifier”, unless it is known that no requirement exists to identify or reference spatial objects of that type. The identifier has to remain unchanged during the life-time of a spatial object. The identifier has to ensure traceability and has to map/incorporate existing national identifier systems.
M	Metadata		Metadata for discovery and for first level evaluation of a spatial data set or spatial data series as required by the Directive, including issues of quality, validity, and conformity are mandated by the Implementing Rule on Metadata (based on ISO 19115). Where applicable, additional theme-specific metadata requirements and/or recommendations shall be specified in INSPIRE data specifications in conformance with ISO 19131.
O	Data & Information Quality (Model)		An INSPIRE data specification shall specify all data quality elements and sub-elements that have to be provided with the data set metadata in accordance with ISO 19113 and the implementing rule on Metadata. This shall include a statement on applicable data quality measures as defined in ISO/TS 19138
P	Data transfer	Delivery Medium	The delivery medium shall be specified in conformance with the implementing rules on network services.
		Delivery Format	The delivery format shall be specified in conformance with the guidelines on data encoding.
R	Multiple representations	Multiple Levels of Detail	In cases where multiple levels of detail are specified for a theme (see Recommendation 27), the representations should in general be required to be consistent one to another. Multiple representations can be used to link the representation of different levels one to another.
		Multiple Representations in different Application Schemas	See Object referencing modelling.

Table 4: Different harmonisation aspects taken into account for harmonisation processing in the HUMBOLDT project

This schema was used as a base for finding out which requirements the scenarios have. As can be seen, data harmonisation is a complex activity. Is is required in many different application domains, and therefore the HUMBOLDT framework has a multitude of things to accomplish. Amongst those are the handling of many different types of data for transformation and presentation in very different environments and processes. These environments and processes are concretized more and more as the scenario requirements are identified.

5.2. Enterprise Viewpoint

The Enterprise viewpoint of this document focuses on collecting the high-level requirements towards HUMBOLDT from several sides, such as the major initiatives HUMBOLDT is embedded in and the major project-internal requirements. Furthermore, it outlines the business processes covered and the added value provided by the specified components. A business process is defined as a set of linked activities that take an input and transform it to create an output, adding value to the input and create an output that is more useful and effective to the recipient.

5.2.1 Business Requirements

1. GMES Requirements

The original work assignment for HUMBOLDT focused on integration between INSPIRE and GMES initiatives. GMES has expressed this requirement in various documents, among them the GMES work programme for 2004 to 2008 (GMES KOM(2004) 65 final version, 3.2.2004), under which HUMBOLDT was also installed. From this document, the following requirements are taken:

Requirement GMES01:

> Integration of INSPIRE and GMES services *shall* be facilitated.

Requirement GMES02:

> Because of a lack of harmonised and interoperable data in GMES activities, a GMES data integration and information management toolset *shall* be developed.

Requirement GMES03:

> The applicability of the developed integration and information management toolset *shall* be demonstrated on a set of scenarios, including Land Cover and Vegetation, Water Resources, Marine Environment and traffic, Atmosphere, Risk Management and Security.

Requirement GMES04:

> For data harmonisation and the development of interoperable services, the current operational situation for end users as well as the integration of end users in all development phases *shall* be essential.

2. INSPIRE Requirements

INSPIRE aims to provide a European-wide unified view on certain spatial data themes as defined within the directive. This unified view is defined in several implementing rules (IRs), which describe both the service-based infrastructure and the characteristics of the data to be exchanged via this infrastructure. These characteristics are described in Data Specifications which are being developed by Thematic Working Groups for each of the spatial data themes included in the INSPIRE Directive Annex, starting with the themes in Annex I. These Data Specifications contain a set of requirements on data content and structure (laid down in application schemas), spatial and temporal reference system, data quality, encoding, portrayal, etc.

In the INSPIRE Network Services Model, the technical infrastructure required to create an INSPIRE-based European Spatial Data Infrastructure is described. This includes Discovery, View, Download, Transformation" and Invoke Spatial Data Services. Requirements for those services are laid down in separate IRs and technical guidelines for each type of service. Since one of the main goals of this project is to support the transition towards this SDI, HUMBOLDT software has to fit in with this technical architecture.

Mapping to the INSPIRE Network Services Model, there are several scenarios where HUMBOLDT components play a role:

1. As a simple or complex transformation service which can handle one or multiple aspects of data harmonisation
2. As a mediator, effectively a multi-purpose harmonising transformation service;
3. As a part of Invoke Spatial Data Services, with a special focus on invoking harmonising transformation services.

Especially complex transformations related to different conceptual schemas and subtle geometric differences are areas where support from HUMBOLDT is expected. These complex transformations also include handling of multiple representations, edge matching and combination of coverage and vector data.

In INSPIRE, the chaining of data services and transformation services into workflows has also been proposed under the title "Invoke Spatial Data Services" services. HUMBOLDT supports this topic specifically for harmonisation workflows, and supports both dynamic workflow composition for data harmonisation and static workflow execution. Both use OGC data provision and processing (transformation) web services. To summarize, the following high-level requirements have been expressed by INSPIRE which HUMBOLDT wants to address:

Requirement INSPIRE01:

> Transformation Services *shall* facilitate the transformation of existing geodata into harmonised and interoperable geodata. The exact structure of this interoperable geodata is defined on a logical and physical level by the Data Specification Drafting Teams of INSPIRE. (Article 11/1, point d)

Requirement INSPIRE02:

> It *shall* be possible to combine Transformation Services by means of "Invoke Spatial Data Services" services.

Requirement INSPIRE03:

> There *shall* be Transformation Services that can translate geodata from one conceptual schema, such as a pre-exisiting unharmonised schema, to another conceptual schema, such as an INSPIRE application schema for a given data theme. (Chapter 1.3 of Draft D3.10 Implementing Rules for Transformation Services, Version 2.0)

Requirement INSPIRE04:

> Transformation Services *shall* follow the Implementation Rules for Transformation Services (IRs) defined by INSPIRE. In cases where other services types are developed within HUMBOLDT, those services shall follow the respective INSPIRE implementing rules.

3. Project High-Level Requirements

Based on these initiatives and their needs, the HUMBOLDT project has a set of challenges that it wants to address. These have been refined in strategy sessions and lead to the following project goals (Ref. Strategy Session 03/2009):

- Supporting SDI enablement by providing functionality for covering the data harmonisation process as a whole;
- Offering the possibility to make use of single functionalities as part of existing infrastructure, such as in the scenarios;

- To summarize, the project aims to ease the harmonisation and interoperability within and between SDIs.

Based on these goals, the following requirements have been extracted:

Requirement HUMB01:

For providing high-quality inputs into the data harmonisation process, the project *shall* provide tools for capturing domain knowledge / application-specific knowledge for supporting the specification of transformations and the definition of the information product. These tools shall be interoperable with existing tools for e.g. data modelling.

Requirement HUMB02:

To ease harmonisation processes and to make them accessible over multiple SDIs, the project *shall* provide means for defining the information product, including parameters such as the target schema, SRS, spatial extent, to which data needs to be transformed;

Requirement HUMB03:

The project *shall* provide technical means for the identification of transformation needs as part of the overall processing of a request for harmonised information;

Requirement HUMB04:

The project *shall* provide a full technical specification and implementation of tools for enhancing the formalization of the transformation between two data schemas;

Requirement HUMB05:

To facilitate easy integration on a data and process level, the project *shall* view the handling of transformation needs as part of the overall processing of an information request;

Requirement HUMB06:

The project *shall* investigate means of enhancing the automation of the data harmonisation processes, depending on the possibility of capturing the required knowledge and the availability of this knowledge in the system;

These requirements are used for finding the focus points of the project, such as the handling of conceptual schema translations.

4. Scenario Business Requirements

The HUMBOLDT scenarios are a set of currently nine application cases which cover a broad range of thematic areas and therefore of required data sets. We therefore expect that harmonisation requirements coming from these nine scenarios to be quite comprehensive in covering typical harmonisation requirements for the 34 INSPIRE data themes. Furthermore, with each scenario also addressing a GMES topic, lessons learned should also be transferable to other GMES applications.

At the time of this writing, a detailed analysis has been conducted for HS-ProtectedAreas and for HS-ERiskA. This detailed analysis represents actual transformations of required data sets for the Use Cases defined in the scenarios. For all other scenarios, preliminary analyses were conducted which included surveying users (please refer to Deliverables A3.3-D1/2 for details) and surveying the existing infrastructure (please refer to document 594, 595, 596, 597, 598, 599 and 600). There are some high-level requirements the scenarios have in common:

Requirement SCEN01:

The project *shall* provide methodological and technical means for increasing the efficiency of currently established processes which have to deal with data integration and harmonisation;

Requirement SCEN02:

The project *shall* provide solutions for the integration of heterogeneous and distributed data sources into one target application or infrastructure, such as a GMES service or INSPIRE ESDI;

Requirement SCEN03:

The project *shall* develop its solutions in such a way that an integration into existing infrastructure is possible;

Further, in a requirements consolidation meeting with the full architecture team and representatives of seven scenarios, a set of eight common requirements clusters and about 40 common, high-priority requirements have been identified.

5. Related Project Requirements

HUMBOLDT has a formal relationship with several on-going research projects. In general, this means there is the requirement to cooperate with these projects and to also take into account their needs when developing harmonisation solutions. In the M2.0 development phase, there was one requirement voiced by the ESDIN project:

Requirement ESDIN01:

The project *shall* develop a tool that allows to define transformations on existing geographical data sets so that these data sets can be transformed automatically to conform to different target infrastructures, such as the INSPIRE-IR based ESDI.

6. Architectural Requirements

There are several general requirements towards a Service-Oriented Architecture. These general architectural requirements are based on the RM-OA list and adjusted where required.

Requirement ARCH01:

The system to be developed *shall* make rigorous use of Concepts and Standards.

Note: This requirement is addressed via the usage of the Handbook of Standards, which defines the de jure and de facto standards to use for various application parts. The final version of the Handbook of Standards will also include the INSPIRE IRs as far as they are available at that point in time.

Requirement ARCH02:

The system to be developed *shall* be composed of loosely coupled services and components.

Note: There are several overall guidelines to ensure this, which are detailed in section 5.3.1.1 (Services and Interoperability)

- *Components as Services: Use Interface Control Document or similar formalisms;*
- *Loose coupling was originally planned by using the OASIS standard for a Service Component Architecture (SCA). This concept allows so-called auto-wiring, where components can agree on a protocol via which to communicate with each other autonomously. We found the currently available implementation to have problems, therefore normal message/service bus structures are used;*
- *Service deployments use standards as defined in the Handbook of Standards;*
- *No cyclic dependencies between framework components are allowed, so that each part can be used on its own.*

Requirement ARCH03:

The developed architecture *should* be largely independent of technology.

Note: New technologies may bring new architectures with them, but basic principles services that are self-sustaining have been stable for at least 25 years.

Requirement ARCH04:

The system to be developed *shall* be designed and implemented following an evolutionary approach.

Note: Please refer to Deliverable A4.3-D1, which details processes for influencing the specification and the specification process itself.

Requirement ARCH05:

The system's architecture *shall* be designed in such a way that individual components can have independent internal architectures.

Note: The HUMBOLDT Architecture consists of two parts: The framework components, which complement each other and therefore have a certain dependence on each other, as with other frameworks and libraries as well (think GeoTools, Spring, …). The second part is the existing overall architecture in which Applications based on HUMBOLDT Framework components are deployed: Here independence has to be maintained and is thus a requirement.

Requirement ARCH06:

The architecture to be developed *shall* be generic with respect to its application domain.

Note: The HUMBOLDT Framework components and the Toolkit Applications shall be independent of the application domain. This means that the HUMBOLDT components should be designed in such a flexible and adaptable way that the HUMBOLDT components can be used across different thematic domains and in different organisational contexts.

Requirement ARCH07:

The MDA (Model Driven Architecture) shall be the central methodological approach for the harmonization process, since MDA is also the approach for INSPIRE interface specification, and other relevant processes.

5.2.2 Actors

For all component specifications, four main actors, each one representing one of the main user groups of the HUMBOLDT framework[2], have been defined. These actors are used in all component descriptions when describing a component's use cases. It is allowable to refine these actor descriptions or to introduce new actors, but at least one of these four main actors has to be driving a Use Case in the component.

1. DATA_CUSTODIAN

The DATA_CUSTODIAN is responsible for managing the provision of his organisation's data sets according to INSPIRE rules. This involves both working in data specification to define common conceptual schemas for his domains and to describe and supervise data harmonization processes by creating mappings between the conceptual schemas defined beforehand.

2 Please refer to document https://intranet.esdi-humboldt.eu/documents/details/555/7.

The DATA_CUSTODIAN is assumed to be a domain expert from a field that uses geoinformation, and has experience with data modelling and classification. It is not assumed that he has specific computer science knowledge.

2. DATA_INTEGRATOR

The DATA_INTEGRATOR is considered to be computer science expert or at least to have very good knowledge of information systems and infrastructures. She is responsible for maintaining and managing her organization's share of the spatial data infrastructure. This means setting up service components, configuring the service and assisting both END_USER(s)_OF_GEODATA and DATA_CUSTODIAN(s) with their work with the system.

3. END_USER_OF_GEODATA

The END_USER_OF_GEODATA is usually a GI expert who is accustomed to working with GI datasets of different types. He knows the concepts of the domain very well and also has a working knowledge of standards-based software. His main goal is working with harmonized data sets so that he can easily use existing data sets from the SDI for his processing, reporting and portrayal product generation responsibilities.

4. END_USER_OF_GEOINFORMATION

The END_USER_OF_GEOINFORMATION is interested in added value applications and services built on the spatial data infrastructure that process geodata towards a situation-dependent information. This can involve hugely different applications, from routing to decision support. As a rule, geoinformation will usually be in the form of portrayed products or reports, only seldom actual data.

For the purpose of defining Use Cases, the END_USER_OF_GEOINFORMATION is assumed to be neither an expert in computer science nor in geographic information science, but rather a decision maker or expert from a different domain.

5.2.3 Business Processes

A business process is defined as a set of linked activities that take an input and transform it to create an output, adding value to the input and create an output that is more useful and effective to the recipient. In HUMBOLDT, this definition is fulfilled by the two main processes reflected through use cases in all scenarios:

- *Provide Harmonized Geodata:* In this process, an organization is required to make an existing data set accessible and usable within the European Spatial Data Infrastructure, e.g. due to INSPIRE requirements or due to market requirements. This process therefore represents the data provider's view on the ESDI. Using the terminology from deliverable A7.0-D1, this business process represents the target definition process, but also the source definition and the mapping of the relationships between those.
  The added value provided by the business process is that data can now be accessed and used in a greater variety of applications, with lower integration hurdles for customers.
- *Use Harmonized Geodata:* In this process, an organization wants to use harmonized geodata and integrate it with their local resources for their specific business processes. This process therefore represents the data user's view on the ESDI. Again, in the terminology of the deliverable A7.0-D1, this represents the “technical process”.
  As with the harmonized data provision, the main added value of this process is the capability of using a great variety of geodata in other business processes without major integration issues, allowing richer services.

These two processes can be understood as the two directions of addressing an integration problem: In the first process, a local resource is integrated with the public infrastructure, whereas in the second process, a public resource is integrated with a local infrastructure. Both processes also involve not just geodata, but also geoprocessing services and capabilities as harmonization and integration targets.

These two business processes have a lot in common and share the same high-level activities, which are a Target Definition, a Transformation Definition and the actual Transformation. This is reflected in Figure 9, below, .

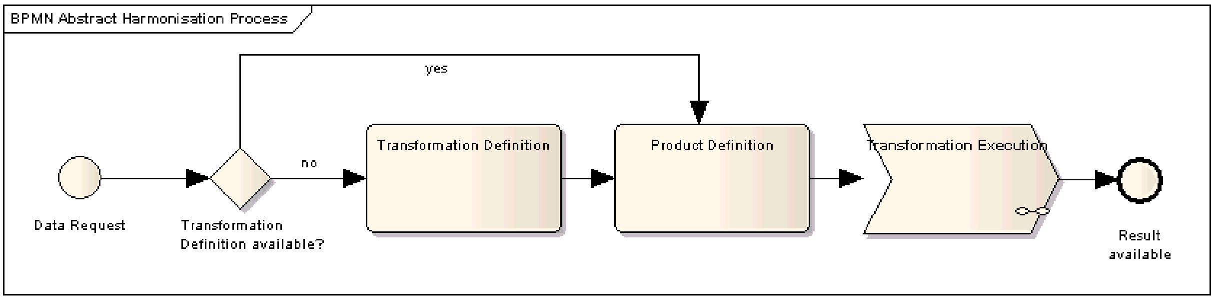


Figure 9: UML BPMN diagram of the common processes and activities of the two main business processes.

In this abstract process, there are three core steps:

1. *Transformation Definition:* Satisfaction of all transformation definition requirements, such as providing capabilities to do certain transformations;
2. *Product Definition:* Definition of the transformation target by providing a product definition;
3. *Transformation Execution:* Execution of the transformation on a given data set, resulting in a data set satisfying the product specification.

Each of these core steps can be mapped to different user groups and provides a business value on its own. As an example, the Transformation Definition phase provides the business value of reducing subsequent integration efforts and of allowing wider usage of the existing data, whereas the product definition phase enables the exchange of formalized product specifications, which can be seen as anything from a map to a complex service mashup.

Since the actual transformation execution is necessary for both business processes, it is described separately. Also, this conceptual and logical separation allows to apply different modes of transformation, such as on-line on-demand or off-line one-time transformations. The other two steps are reflected in the concrete business processes in different ways.

Each of the activities shown is reflected by one (or multiple) of the components in the HUMBOLDT framework, and often also by one of the applications form the toolset, as explained in the End-to-End example and in the more abstract following sections.

1. Provide Harmonized Geodata

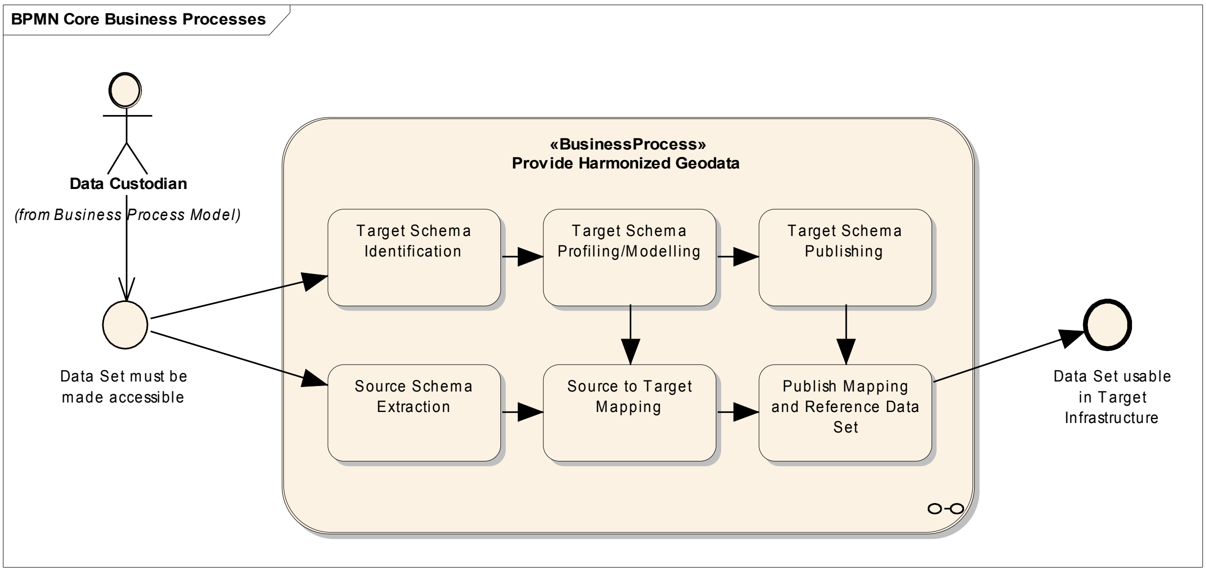


Figure 10: UML BPMN diagram of the "Provide Harmonized Geodata" process

1. *Target Schema Selection:* Discovery of an existing schema that matches the own domain best and can be directly used or adapted. This activity is usually carried out by the Data Custodian, but can also be valuable to Data integrators and End Users of Geodata.
   Components: Model Repository and HUMBOLDT Model Editor;
2. *Target Schema Modelling/Profiling:* Either creation of a new schema or profiling of an existing one, if none of the identified schemas matches well enough to be used directly. This activity is carried out by the Data Custodian.
   Components: HUMBOLDT Model Editor;
3. *Target Schema Publishing:* Publishing of the Target Schema either to the local or to the public infrastructure. This activity is carried out by the Data Custodian.
   Components: Model Repository, HUMBOLDT Model Editor;
4. *Source Schema Extraction/Modelling:* If the source Schema is not available in a sufficiently formalized and rich way, it needs to be extracted, either automatically or manually (or a combination of both). This activity is carried out by the Data Custodian.
   Components: HUMBOLDT Model Editor (this functionality is part scheduled for M3.0);
5. *Source to Target Mapping:* Creation of a set of transformation rules to transform a data set available in the source schema to a data set in the target schema, and possibly also from target to source. This activity is usually carried out by the Data Custodian.
   Components: HUMBOLDT Alignment Editor;
6. *Publish Mapping and Data Set:* Publish the source schema, the source-target mapping and the data set itself to the local/public infrastructure. This activity is carried out by the Data Custodian.
   Components: Model Repository, standard data access service (e.g. WFS), standard catalogue (e.g. CSW);

2. Use Harmonized Geodata

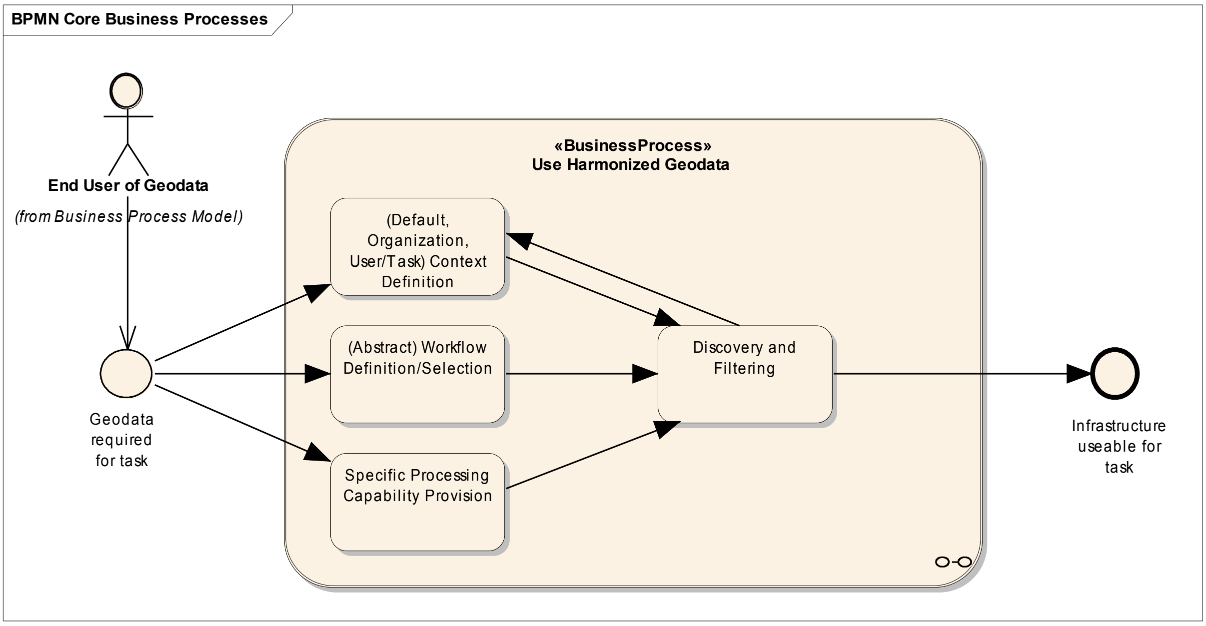


Figure 11: UML BPMN diagram of the "Use Harmonized Geodata" process

1. *Task Context Definition:* This is a preparatory activity that can be carried out by any of the user groups except the End User of Spatial Information. Defines the requirements of an infrastructure, an organisation, a user or a certain task towards the data and services that can be used in the context of a certain task. This can involve setting quality constraints, thematic constraints, including a target conceptual schema, and also metadata constraints, such as accepting only data sources from a national mapping and cadastral agency.
   Components: Context Service, Context Service Web Frontend;
2. *Abstract Workflow Definition/Selection:* This activity involves defining or selecting a pre-existing work flow consisting of the basic processing steps necessary for the accomplishment of a certain task, as far as they are represented by processing capabilities available in the SDI. This is usually done by the Data Integrator, and can involve the provision of specific processing capabilities.
   Components: Workflow Design and Construction Service, Workflow Frontend (scheduled for V2.0);
3. *Specific processing capabilities provision:* This activity involves implementing and deploying local or public processing services that are required for the specific harmonization and transformation steps needed for an End User's tasks. The activity is usually carried out by the Data Integrator.
   Components: Conceptual Schema Transformer, Coordinate Transformation Service, Edge Matching Service and others;
4. *Discovery and Filtering:* This activity, which is usually carried out by the End User of Geodata, involves browsing the available, already context-filtered resources to see which ones match best for the task at hand. The activity ends with a selection of the data sets required.
   Components: Model Repository, Information Grounding Service, Mediator Service;
5. *Retrieval of transformed/harmonized Geodata:* This activity, which can be carried out by all user groups, represents the actual retrieval of geodata (or geoinformation gained from the processing of geodata) from the ESDI. This involves all necessary harmonization and processing steps.
   Components: Model Repository, Information Grounding Service, Mediator Service, Workflow Design and Construction Service;

3. Transformation Process

The transformation process itself runs fully automatic based on the definitions for products and transformations created before. It is in large parts dependent on the concrete heterogeneities that need to be resolved. There are some processes that are common to most concrete transformation processes, though. These are outlined in the activity diagram in Figure 12.

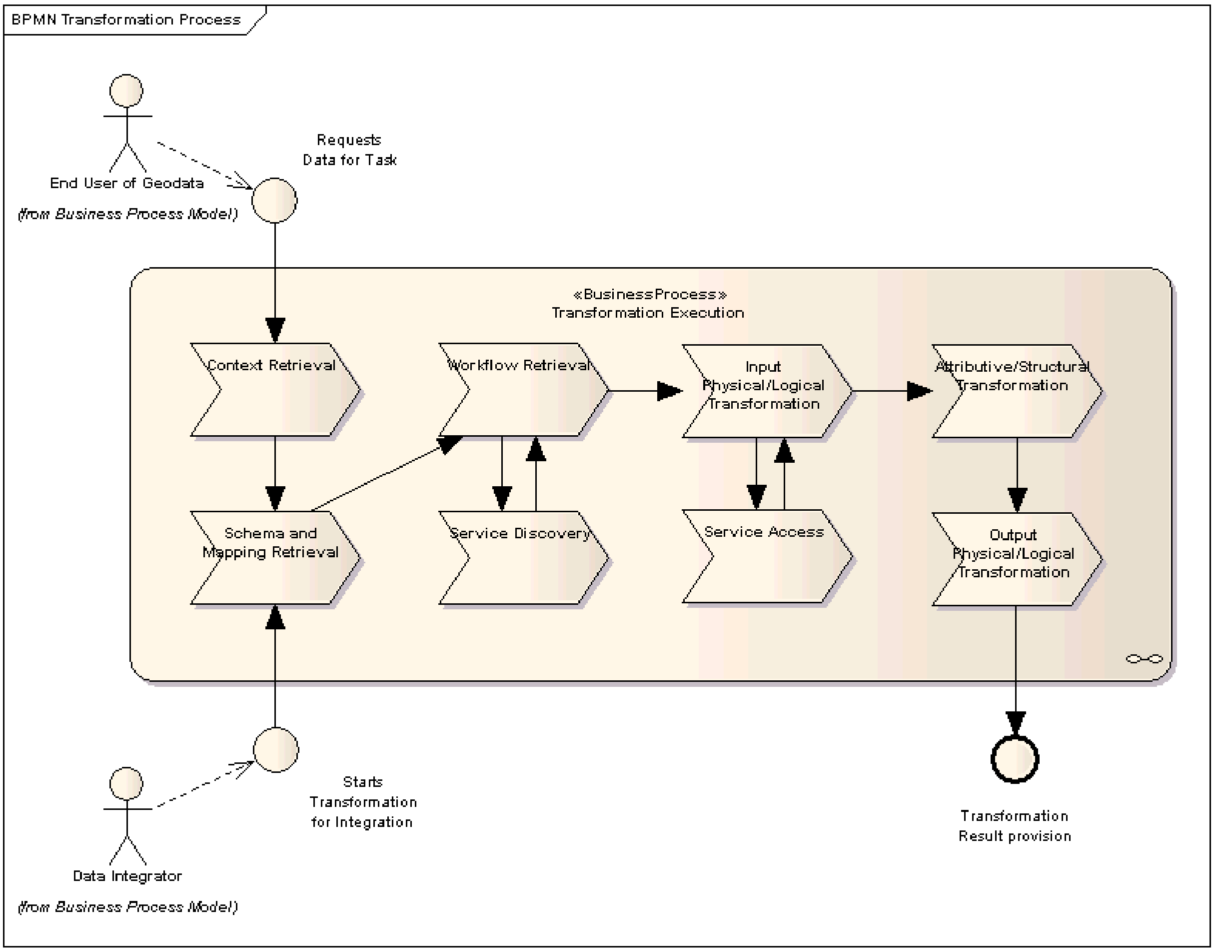


Figure 12: The process steps of the Transformation Execution for data usage and data integration.

This view of the business process already suggests three groups of software artifacts to be designed and implemented:

1. The *HUMBOLDT Harmonisation Toolkit*, which allows different user groups (Data Custodians and Data Integrators) to set up everything needed for a harmonisation transformation, such as describing source and target schemas, creating an alignment between them and defining a scenario-specific processing workflow;
2. The *HUMBOLDT Framework for Service Integration*, a set of service components which determines data source fitness, applies processing to data and delivers the harmonised data to clients;
3. The *HUMBOLDT Scenario Applications*, in which specific processing required for the scenarios can be executed.

An overview of these three groups is provided in the next section.

5.3. Computational Viewpoint

5.3.1 Approach

The main aim of specification has been the fulfillment of the high-level requirements outlined in the Enterprise Viewpoint, while at the same time taking into account the technical requirements stated by the scenarios and supporting the described business processes.

To reach these goals, we started with the creation of a generic infrastructure (ARCH06) needed for implementing the *"technical harmonization process"* (Transformation Execution, please refer to A7.0-D3), while taking into account the different data structures used in the scenarios (SCEN02) and also the need to support the *"target definition process"* (Transformation and Product Definition, please also refer to A7.1-D1).

This is accomplished via an architecture that contains both horizontal, cross-cutting service components and vertical, task- and domain-oriented applications. Both categories are represented in the following figure, where integration of existing services and applications in the scenarios and in the shared infrastructure with HUMBOLDT services and applications is shown.

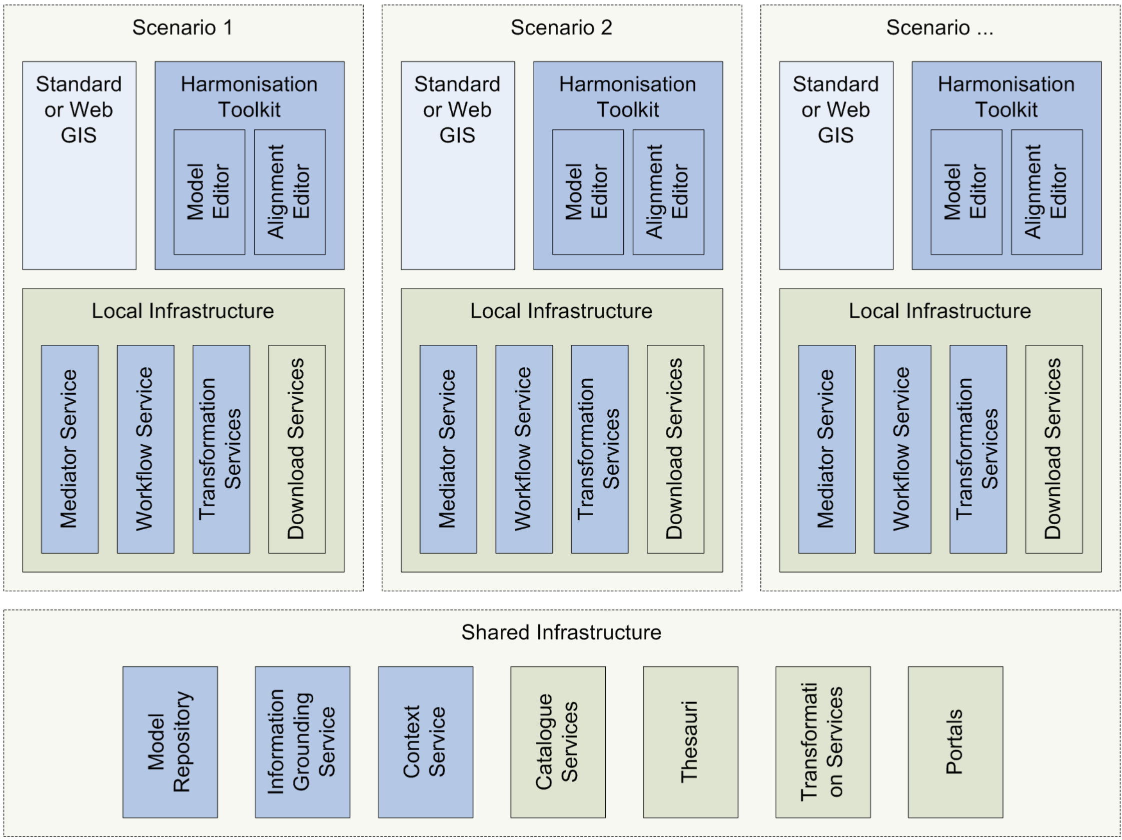


Figure 13: A HUMBOLDT-enriched SDI connecting different scenarios (HUMBOLDT components in blue, existing components grey, combined components bright blue)

The overall architecture foresees three major tiers that are integrated with the existing infrastructure of data portrayal, data provision, data processing and data cataloging services via standardized interfaces:

1. *A distributed, shared infrastructure tier* that gains added value from collaborative usage across organizations, countries and domains;

2. A *localized, specific infrastructure* of services that gains it's value from being specifically adopted to the needs of a single domain and/or organization. This tier includes tailored processing services, portrayal services and workflow management;
3. Finally, a *clients tier* that includes specific clients required for the discovery and editing of information provided through this distributed infrastructure as well as clients for the interaction with components of the specialized infrastructure tier;

In principle, every element available on the distributed infrastructure tier can also be deployed into the localized infrastructure tier in cases where information cannot be made public. However, especially with the goals of the INSPIRE initiative (INSPIRE03), it should be clear that quite a substantial portion of schema descriptions, mappings between those and to available services and also product descriptions schemata (Context documents) should be made publicly available.

The second major aspect of the concept was to be least invasive into current infrastructure (GMES04). HUMBOLDT software cannot be seen as something that will replace existing systems, but rather support and amend them in several specific places. To be able to do so, functionality is well-isolated from the interfaces by which it is accessed (ARCH05), resulting in components that can be adopted easily for different deployments and process synchronization styles, as shown in Figure 14 below.

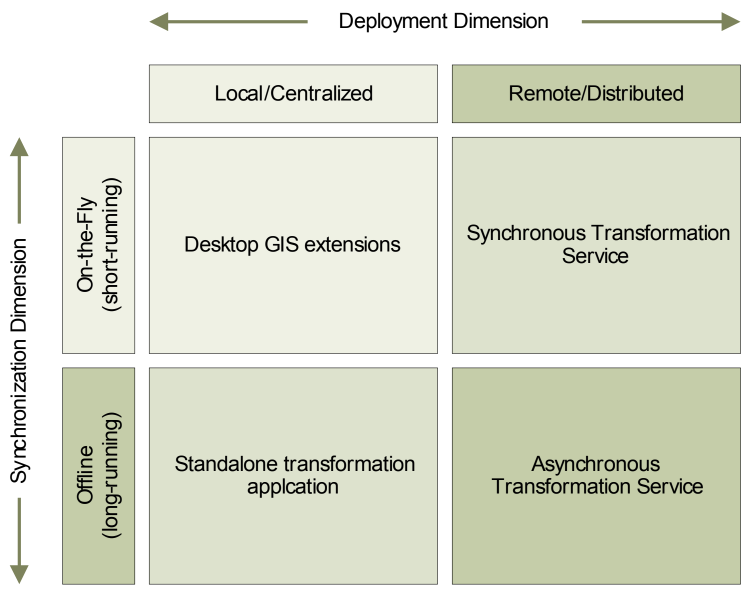


Figure 14: Deployment and synchronization dimensions taken into account for HUMBOLDT Service Components

Further to the goal of being minimally invasive, established interfaces are used and the infrastructure as a whole is designed to appear as a transparent proxy to users working on harmonized geodata sets. How this is accomplished in depth is explained in the next sections, and in greater detail in the individual Component Specifications.

The third major aspect concerns the notions of a *framework for service integration* and of a *harmonisation toolset*. The framework is considered to be a collection of service components that work based on predefined configuration and definitions to provide clients via standard interfaces with harmonised geodata (GMES01, GMES02). They can be used in different deployment contexts and are designed in such a way that they can be re-used functionally as often as possible. However, the components are not meant to be used directly by end users of geodata or geoinformation; they are rather something a developer would use to provide those main user groups with the tools they need, such as the portal described in chapter 4.

The toolset, compared to this, is actually comprised of applications ready to be used – it might have to be adopted to the application domain, configured and installed, but generally, these are applications directly addressing a user group, such as the data custodians or the data integrators. Furthermore, the toolset applications all have fixed deployment types, such as as a GIS plugin or as a web application. What can been seen from that is that a toolset application is built from multiple service components from the framework, and that a scenario application is built from toolset applications and framework service components.

1. Services and Interoperability

Internal and external interoperability are core requirements for this architecture, and this section outlines how it can be achieved.

External interfaces, i.e. interfaces between HUMBOLDT components and existing infrastructure or clients use, whenever possible, existing, established and standardized interfaces, such as OGC's specifications on the Web Feature Service, Web Map Service and Web Coverage Service (ARCH01). In cases where this is not possible, WSDL descriptions are to be used (ARCH05).

Internal interfaces, i.e. interfaces between HUMBOLDT components, are described by using WSDL and communicate using SOAP, and there may not be any cyclic dependencies between HUMBOLDT components to ensure loose coupling (ARCH02).

Transformation services (INSPIRE01, INSPIRE03), being the most important extension point in the HUMBOLDT Framework, make use of the OGC's Web Processing Specification 1.0.0 to ensure interoperability. In M2.0 and M3.0, the HTTP POST XML encoding is regarded as mandatory, whereas the SOAP encoding is regarded as optional (INSPIRE04). Implementations of transformation services in HUMBOLDT have to fulfill several rules explained in document [2.6] to ensure their interoperability and the possibility to chain them into harmonisation workflows (INSPIRE02).

2. Component-Based Design

As in the previous phases, the approach in specifying the HUMBOLDT Framework follows the concepts of *Component-Based Design* (CBD)[3], which can be combined very well with SOA principles such as loose coupling[4]. As it is the goal of the HUMBOLDT software to be adaptable to many deployment situations (ARCH02, ARCH03,), this concept with its focus on re-usable components fits very well.

In CBD, components and the contracts between these play a central role. A contract in this context has two different characteristics: one is as a usage contract, describing the commitment a component interface makes towards a client, the other as a realization contract between the specification of a contract and the actual implementation of the contract. This realization contract is formalized to ensure interoperability and loose coupling of the components, so that a component implementation could be replaced with a different implementation of the same realization contract.

Following these concepts, each component has provided interfaces and required interfaces. These specify the communication between components and define functional capabilities which a certain component can and has to offer. Finally, components can be combined into service components of lower granularity by using ports.

This basic concept is applied throughout several levels of abstraction that are used throughout all tiers that a system being designed has:

3 Szyperski, C. (1998): Component-based Software, Addison Wesley

4 This can be seen in many examples, such as the programming model called the Service Component Architecture (SCA), or in cases where OSGI is used to provide web services.

- *Component System Framework:* The sum of all components contained in the HUMBOLDT Framework, which can be recombined to form different services. The Component System Framework consists of 1..* Component Frameworks. It manages these and defines communication between the individual component frameworks. This management includes installation, instantiation and activation.
- *Component Framework:* A Component Framework is an architecture dedicated to a certain functionality an a certain amount of mechanisms implemented on Component level. The Component Framework manages a set of 1..* Components.
- *Component:* A Component is an aggregation of 1..* Modules which has a public interface that is being accessed by the Component Framework.
- *Module:* A Module is a component-internal structure that does not offer public services, but is used by one or multiple Components.

These terms will be used to group parts of the specification in the following sections.

5.3.2 Component Frameworks

HUMBOLDT software is grouped logically into three Component Frameworks:

- *Harmonisation Toolkit Component Framework:* A collection of components representing the interaction with Clients, i.e. human users and remote systems accessing HUMBOLDT functionality. In terms of the harmonization process, the components contained in this component framework support the target definition process.
- *Service Integration Component Framework:* A collection of components providing the core transformation and harmonization capabilities. This includes the management of information models, their relationships in between, their relationships to Grounding Services, the management of user context including Web Map Context, and quite a few more. In terms of the harmonization process, the components belonging to this component framework support the technical process.
- *Transformation Service Component Framework:* This is a collection of transformation services, which can be used to handle individual harmonisation aspects, such as conceptual schema transformation, edge matching or different spatial reference systems. Application-specific processing capabilities such as oil spill simulation also belong to this group.

These three Component Frameworks also represent three tiers of a HUMBOLDT framework-based system. The following component diagram shows how these three Component Frameworks relate to each other and which service components reside within each tier.

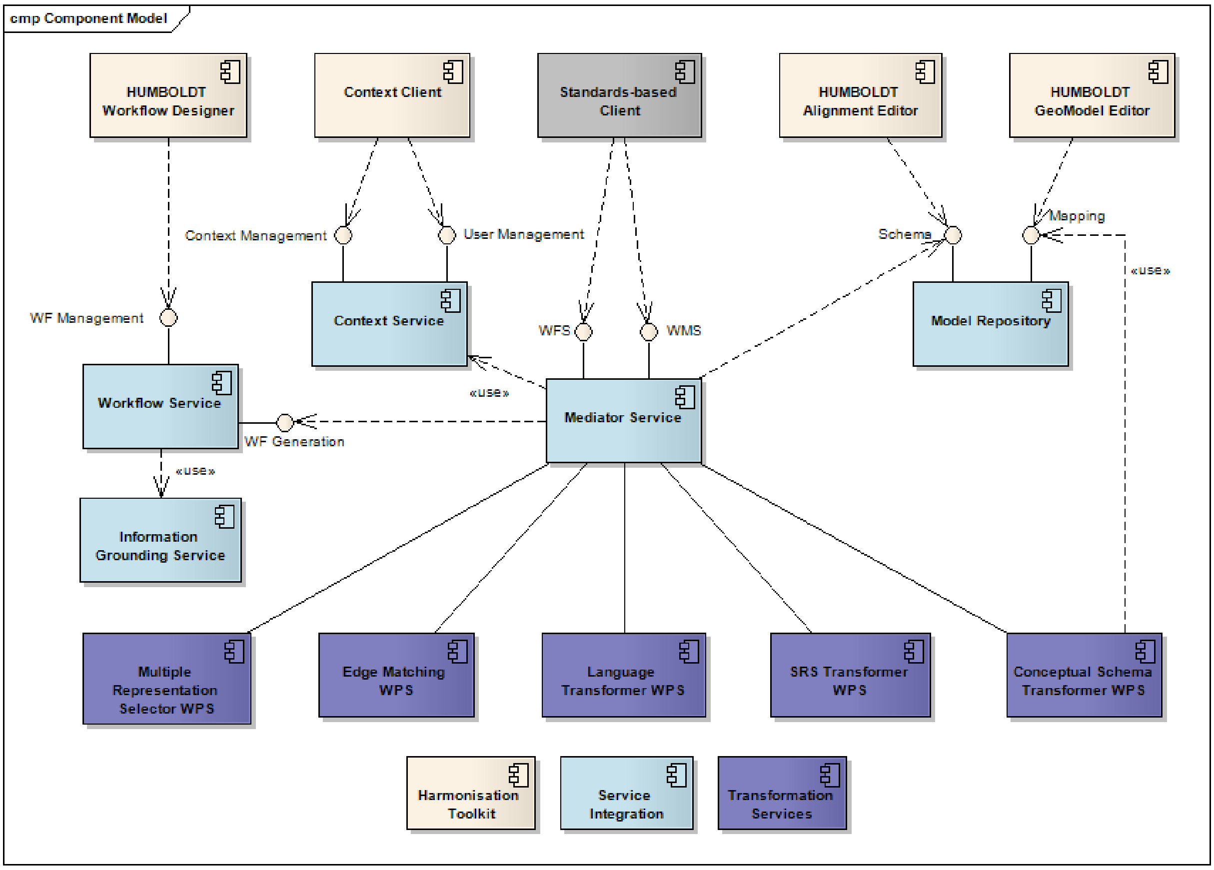


Figure 15: UML component diagram of the service components and the component frameworks in the HUMBOLDT framework

In addition to the services and applications of the HUMBOLDT framework, the diagram also includes several "grounding services". These grounding services represent the existing infrastructure of data provision and portrayal services as well as processing and cataloguing services. The "*StandardClient*" represents any already existing client software that can make use of OGC standard interfaces like WFS or WMS.

As an example of how these service components collaborate to handle one of the activities of the business process described in the Enterprise Viewpoint, the following descriptions and UML sequence diagram shall assist. Please note that the other activities are not described here in detail, so please also refer to the Enterprise Viewpoint of this document section and to the individual component specification documents.

Activity "Retrieve Harmonized/Transformed Geodata":

1. The `StandardClient` (an off-the-shelf desktop GIS with support for retrieving geodata and geoinformation via standard interfaces) sends a request, such as a Web Map Service `GetMap` request, to the Mediator Service, including a Context Identifier registered beforehand with the `ContextService`. This is done by using an URL whose host part contains a context identifier to retain standard compliance and at the same time be able to identify a user's harmonization requirements.
2. The `MediatorService`, specifically it's `InterfaceTranslationFaçade` module, accepts the request, decodes it and constructs an enriched request object from the incoming URL K/V pairs. It also uses the identification information in the URL to retrieve the user's harmonization preferences from the `ContextService`.

3. The enriched request is then delegated to the `MediatorService`'s module `RequestBroker`, effectively a system controller. Before continuing the execution of the request, the `RequestBroker` first checks whether the Feature Types and the conceptual schema specified in the thematic query constraints, which were expressed as Layer names in the case of a WMS GetMap query, are known..
4. In a next step, the `RequestBroker` uses the enriched request to request an executable workflow from the `WorkflowService` (WS). The WS accomplishes the assembly of this executable workflow via identifying an abstract workflow (the so-called basic workflow) using the FeatureType (i.e. layer name) that is requested.
   a) In case, there is no Basic Workflow for that particular FeatureType, the Workflow Design and Construction Service directly passes the user request as a data service discovery-query to the IGS. If there is a data service perfectly matching the user request, the WS returns a pointer to this service to the Mediator Service. In case, there are no perfect matches for the end user request, the WS analyses the (partial) matching data services, picks one of them and automatically constructs a harminisation workflow (e.g. by including schema translation, SRS harmonisation) that harmonises the data according to the user request. This harmonisation workflow is returned to the Mediator Service.
   b) In case there is a Basic Workflow defined for the FeatureType requested, the WS reads out the preconditions of the BW (i.e. the constraints on the data sources that deliver input) and the list of constraints passed with the enriched request and makes use of the Information Grounding Service for discovery of suitable (grounding-) data services that can deliver input to the BW.
      - In a case where no direct grounding is available (no data service perfectly matching the individual BW preconditions), the `WorkflowService` can either expand the query so that Grounding Services whose output will need some processing are taken into account or it can broaden the query by e.g. considering previously defined schema-mappings. After discovering such non-perfect matching services, the WS automatically constructs a harmonisation workflow such that all preconditions of the BW and the user request are satisfied. Finally, the executable workflow is returned to the Mediator Service. .
5. After retrieving the executable workflow from the `WorkflowService`, the `MediatorService` analyses whether any additional translation is necessary for the formulation of the query towards the identified grounding service. When this is not the case or when it is completed, the Grounding Service information together with the specific request parameters is passed to the `MediatorService`'s module `AccessQueueManager`, which handles all communication with upstream services. In parallel, the `RequestBroker` dispatches the workflow itself to another module that is part of the `MediatorService`, the `TransformationQueueManager`. This module is the actual workflow execution engine and executes the steps of the workflow as soon as the the required prerequisites, such as in the form of responses from grounding services, are available.
   a) When the upstream service accessed cannot deliver a GML-encoded answer, an additional physical schema translation can be necessary. This is handled by a so-called `DataAccessService`, together with any particularities of the protocol used to talk to the upstream service or data source.
6. The `AccessQueueManager` then dispatches the requests as necessary to the individual services outside the HUMBOLDT framework. When a request to a Grounding Service was successfully completed, the AQM returns it's answer to the `RequestBroker`, who dispatches it

to the `TransformationQueueManager`, thus already handling physical schema heterogeneity.

7. After completing execution of the workflow in the TQM, the result is first returned to the `RequestBroker`, then to the `InterfaceTranslationFaçade` which handles the encoding to the physical schema requested by the client and finally to the standard client who initiated the request.

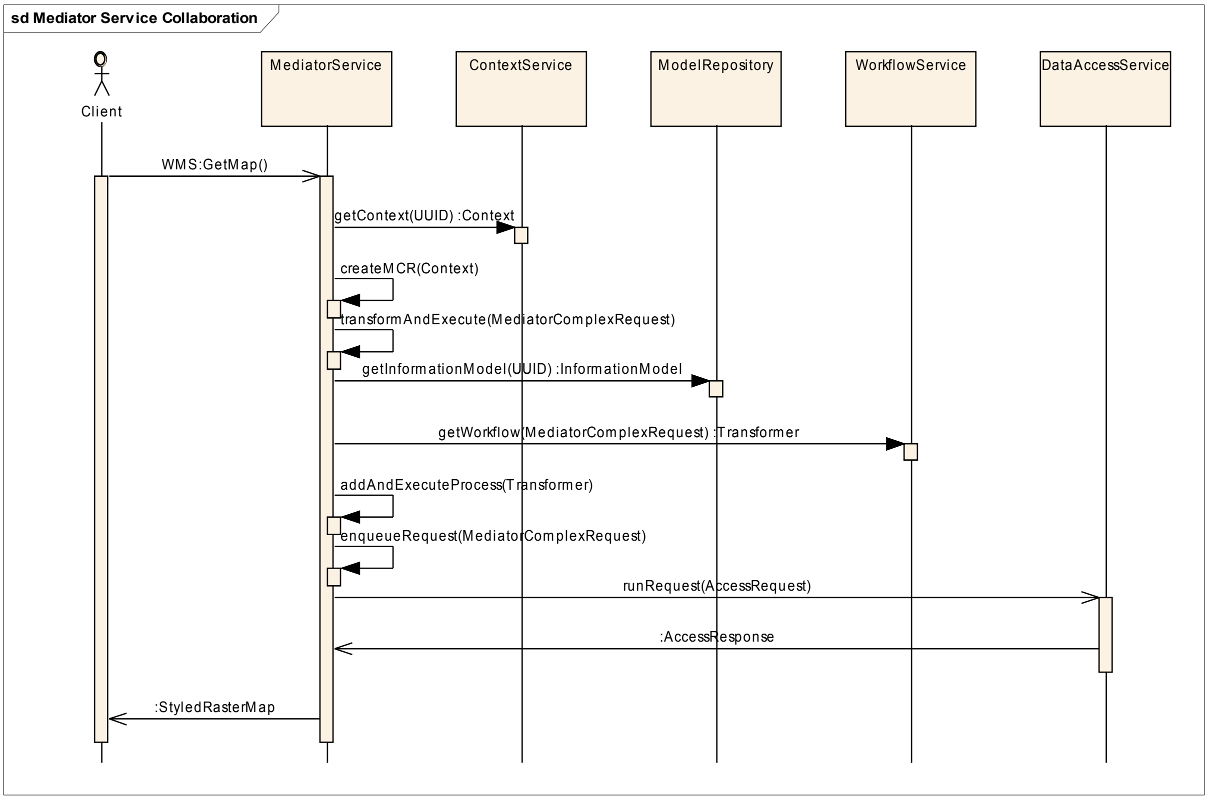


Figure 16: Service-level sequence diagram outlining the interactions of the Mediator Service Component with other service components in the HUMBOLDT framework, focused on the Mediator Service

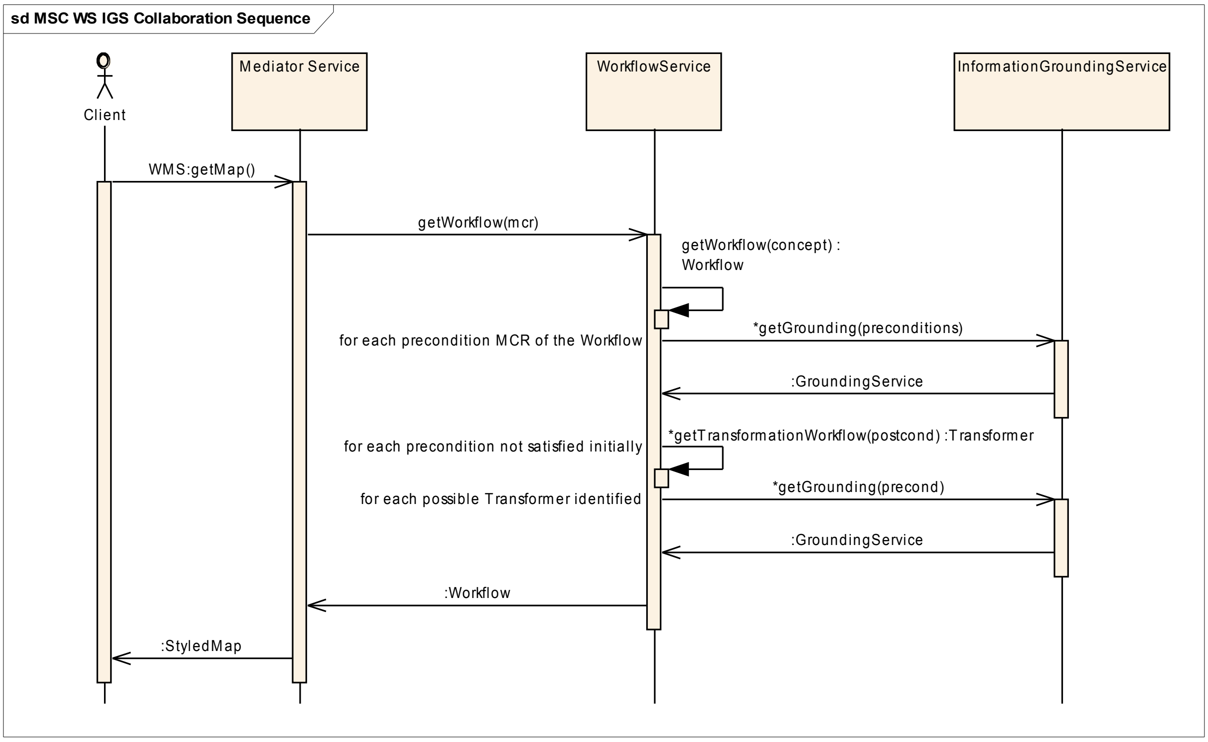


Figure 17: Service-level sequence diagram focusing on the interaction between the Workflow Design and Construction Service and the Information Grounding Service

For more information on the modules of the Mediator Service that are mentioned, please refer to the Mediator Service Component description in the next section.

5.3.3 Service Components

The service component descriptions given here focus on a short description of the purpose of the service component, as well as a shorthand Component Responsibilities Collaboration card-style information. For the available additional information, please refer to the six sub-documents belong to this document. The components that have been part of specification work are numbered like this [3.x] while for those componenets that have not been part of M3, we refer to the previous version (mostly M2), indicated by [2.x].

1. [3.2] Mediator Service

The Mediator Service is mostly a component used in the technical process part of data harmonization. From the perspective of end users of geodata and geoinformation it is a very important part of the HUMBOLDT framework, and at the same time one that aims to be as invisible as possible. The component is designed as a proxy service, which interferes with normal usage of geoservices and data only in a minimal way, to provide users with integrated services and harmonized geoinformation attuned to their needs, and also to make discovery of data and services a user-centered experience.

Responsibilities

- The MS is the main controller of this component framework.
- The MS coordinates all access to remote and local data sources and processing capabilities.
- The MS handles all physical, logical and conceptual schema transformations necessary by coordinating other components and services.
- The MS is also responsible for maintaining transparency in terms of providing a processing lineage of all processed data sets.

Collaboration

- *Model Repository:* The Model Repository is accessed to verify the thematic constraints imposed by the user or his context.
- *Context Service:* The CS is accessed to retrieve the context of any query directed to the Mediator Service.
- *Workflow Design and Construction Service:* The WS is accessed to retrieve an executable workflow based on the constraints defined through the user's context and query.
- *Data Access Services:* These are accessed when physical and logical schema heterogeneities as well as protocol heterogeneities have to be solved when accessing upstream services.
- *Data Transformation Services:* These are accessed when indicated so by the Workflow to be executed and can be wide-ranging, from coordinate transformation services to different web processing services and to a conceptual schema translation service.

Additional Information

The following component diagram gives an overview of the internal composition of the Mediator Service.

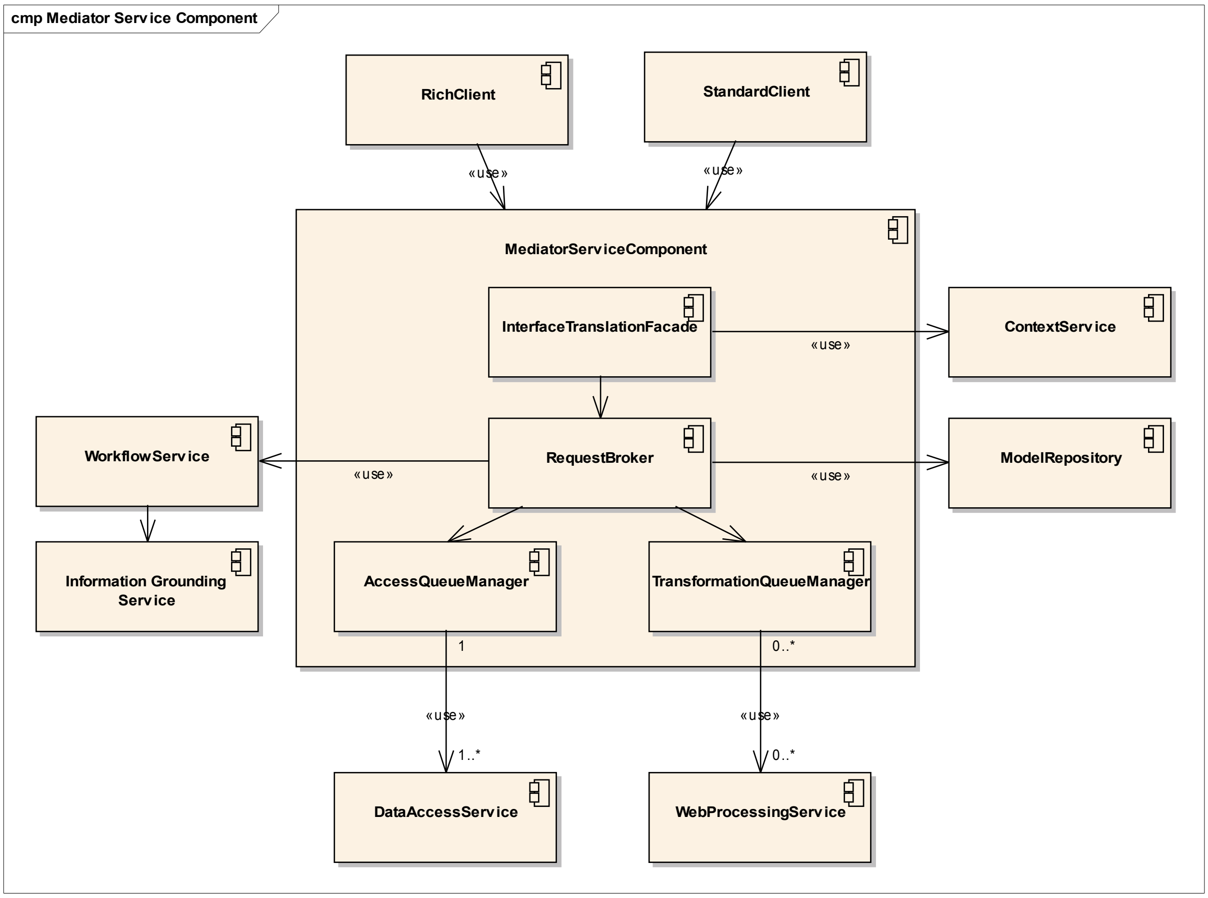


Figure 18: Component Diagram showing the main modules of the Mediator Service as well as usages of other services and service components

2. [2.3.3] Model Repository

The Model Repository is a structure that is part of many core business processes in HUMBOLDT, and is also able to provide added value to many tasks involving geoinformation on it's own. It is used both within the target definition process and the technical harmonisation process.

First of all, the Model Repository represents a distributed catalogue of Feature Catalogues that is enriched with information on how these Feature Catalogues and the Feature Types contained within relate to each other. This can be considered as a basic infrastructure aspect of a spatial data infrastructure, to which it adds value by providing the following capabilities:

- Discovery and Analysis of (classification) schemas used throughout the SDI;
- Publishing of new schemas and connection of the new schema to existing ones;
- Possibility of automated schema translation of a data set based on the schemas and mappings.

Consequently, the component helps in understanding schemas from different spatial and thematic domains, and it helps with bridging the differences between them. At the same time, it means only a minimal overhead to each organization using it, and does not require one central registry with high maintenance efforts.

Responsibilities

- The Model Repository offers both services for storing and retrieving information models (such as conceptual schemas) and also provides services for storing and retrieving mapping descriptions between information models. It bundles these two services since information models and mappings also directly relate to each other.
- It offers various operations to identify and retrieve an Information Model and to enrich it with further information.
- It also offers operations to provide various types of mappings, i.e. structured data describing a transformation of a certain data set to another physical, semantic or other schema.
- The Model Repository is not in itself responsible for executing the transformations from one Information Model to another, it just provides guidance on how to do so.
- The Model Repository is constructed in such a way that it can be distributed, so that there is one virtual Repository acting like a catalogue of all Model Repositories.

Collaboration

- The component is used by the Mediator Service, which retrieves Conceptual Schema information from it.
- The component is used by the Context Service to define thematic constraints.
- The component is used as the back-end for the Data Harmonisation Toolkit.
- The component is used by the Conceptual Schema Transformer [2.6.1] for retrieval of executable mapping documents.

Additional Information

- The MR addresses several of the project's high-level requirements (HUMB01, HUMB04, HUMB06).

3. [3.4] Context Service

The Context Service is the HUMBOLDT framework's means to enrich existing interfaces with semantic and other harmonization requirements without having to change any of the existing interfaces. It provides handling of all configuration aspects as well.

Responsibilities

- This component handles the configuration for all users that want to use the data harmonisation capabilities that HUMBOLDT offers.
- The component can manage users, i.e. allow creation, modification and removal of them.
- The component can manage contexts, i.e. allow creation, modification and removal of a context using its unique identifier. This includes Web Map Context 1.1 documents, which can be created for every managed context.
- The component allows administrators to specify harmonisation and business constraints for users.

Collaboration

- This component is being used by the Context Service Web Frontend for user and context management by data integrators.
- This component is being used by the Mediator Service to retrieve the information on contexts (i.e. context constraints, default, organization or user part of the contexts).
- The component can be used by the Standard GIS Client to load Web Map Context 1.1.1-compliant documents.

Additional Information

- The CS is capable of mapping the more complex context used in the HUMBOLDT framework to a simple Web Map Context document for better integration with WMC tools.
- The CS component also includes a Web Front-End for context and user management. This web frontend has the following responsibilities:
  - This component handles the configuration for all users that want to use the data harmonisation capabilities that HUMBOLDT offers.
  - The component can manage users, i.e. allow creation, modification and removal of them.
  - The component can manage contexts, i.e. allow creation, modification and removal of a context using its unique identifier. This includes Web Map Context 1.1 documents, which can be created for every managed context.
  - The component allows administrators to specify harmonisation and business constraints for users.
- This component addresses one of the project's high-level requirements (HUMB02).

4. [3.7] Information Grounding Service

This component is responsible for the linkage between items in an information model and concrete OGC or other services which are available in the Grounding Tier.

Responsibilities

- The IGS has to periodically harvest existing catalogue services in the Grounding Tier using the Harvest Module.
- The Request Handler module is responsible for providing the Workflow Design and Construction Service with the information of the Grounding Services which fulfil the imposed constraints when the Mediator demands it.
- The Grounding Service repository is a persistence facility for the metadata of the Grounding Services in which the Workflow Design and Construction Service is interested.

Collaboration

- This component is used by the Workflow Design and Construction Service to discover services matching the constraints imposed by a client's request.

Additional Information

- The IGS makes heavy use of CSW implementations, such as GeoNetwork.

- This component addresses one of the project's high-level requirements (HUMB06).

5. [3.5] Workflow Design and Construction Service

The Workflow Design and Construction Service (WDCS) component plays a vital role in two processes: First, it enables users to specify application specific processing chains (i.e. Basic Workflows) such as the one described in Chapter 4 of this document. Second, it provides the capability of analysing data sets / services and automatically constructing harmonisation workflows based on the metadata of available data services and end user requirements.

Responsibilities

- Workflow Management: The WDCS performs managerial functions by maintaining a repository of workflows and transformers.
- Workflow Construction and Provision: The WDCS constructs and provides the Mediator with Executable Workflows.

Collaboration

- The WDCS invokes the IGS to get information on available, matching grounding services.
- The WDCS is invoked by the Mediator Service for retrieval of an Executable Workflow.
- The WDCS makes use of available – HUMBOLDT internal and external – data transformation services. It does not execute them but composes them into workflows.
- The WDCS accesses the Model Repository for identifying available schema-mappings

Additional Information

- This component addresses several of the project's high-level requirements (INSPIRE02, HUMB03, HUMB06).

6. [2.9.x] Data Access Services

The main objective of the DAS is to handle physical and logical schema heterogeneities before processing data in the HUMBOLDT framework. There is one service per schema to access; to make this more concrete, there will be a DAS for every data structure that is not part of GML, such as shapefiles or COLLADA or NetCDF. Each DAS can be seen as a simple Format Conversion Service.

Collaboration

- These components are being used by the Mediator Service when accessing upstream services and data sources.

Additional Information

- In M3.0, these services are seen as modules within the Mediator Service, but can still be deployed as Format Conversion Services if necessary. In this case, they will be packaged as a Web Processing Service 1.0.0. The modules can also be used in other services, especially the transformation services, so that they can accept additional input and output formats in their process descriptions.

5.3.4 Data Harmonisation Toolkit

The DHT provides applications supporting data custodians and data integrators in the following activities:

- source conceptual schema extraction/modelling based on a concrete data set;
- automatic annotation of the concrete data set according to the extraction of it's conceptual schema;
- source to target schema mapping based on a known target schema and an extracted schema;
- publishing of source and target models as well as mappings;

All of these activities are partly automated and are executed in a task-oriented user interface that provides live feedback on the implications of model extraction and mapping decisions.

Collaboration

- The component makes use of the Model Repository for retrieval and persistent storage of schemas and mappings.
- The component makes use of the Mediator Service to enable live feedback in schema mapping processes based on reference data sets and the concrete data set to be harmonized.

Additional Information

- The DHT is being implemented on the basis of the Eclipse RCP.

1. [2.3.1] HUMBOLDT Model Editor

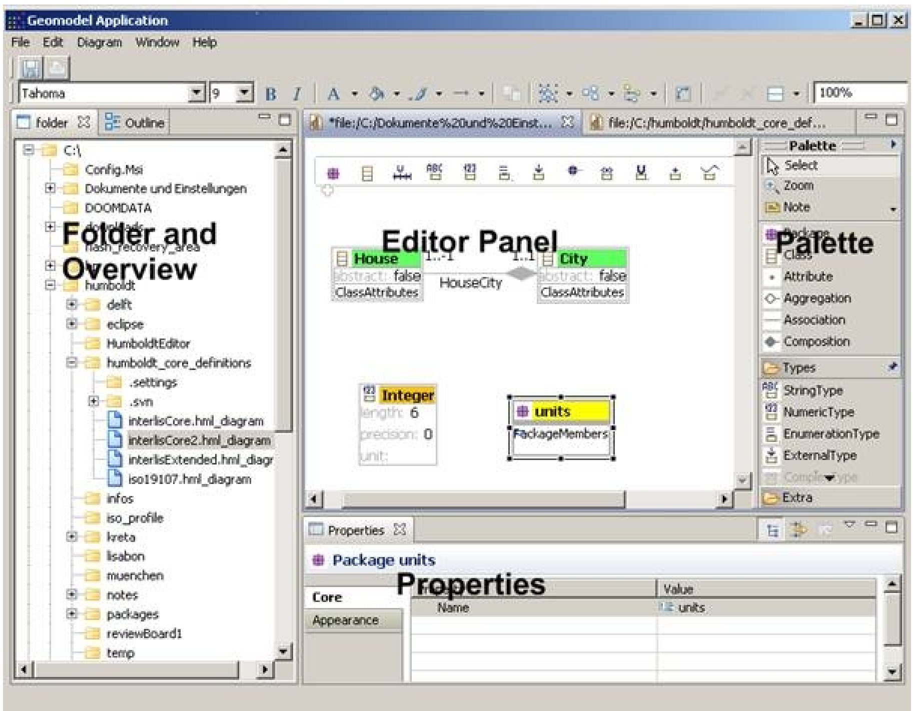


Figure 19: Layout of the HUMBOLDT Model Editor

The HUMBOLDT Model Editor is a tool to support users, specifically data custodians, in producing suitable data schemas and data specifications for geospatial demands and to support the analysis of existing data sources.

Furthermore, GML schemas can be generated in reproducible manner. The data models and the encoding of these is a valuable input to the harmonisation process. In that sense, the HUMBOLDT Model Editor is not designed as integrated component in a technical framework, but primary a standalone element to help the user in that task.

Collaboration

- This component is part of the Harmonisation Toolkit.
- This component stores models using the Model Repository.
- The HUMBOLDT Alignment Editor makes use of the schemas defined in this application..

Additional Information

- This component addresses several of the high-level requirements of this project (HUMB01, HUMB02).

2. [3.3.1] HUMBOLDT Alignment Editor

The HUMBOLDT Alignment Editor, short HALE, is a rich graphical user interface for defining mappings between concepts in conceptual schemas (application schemas created with the HUMBOLDT Model Editor), as well as for defining transformations between attributes of these schemas. These mappings are expressed in a high-level language and can later be used by the Conceptual Schema Transformer processing component to generate an executable form of transformation, such as an XSLT for XML input/output.

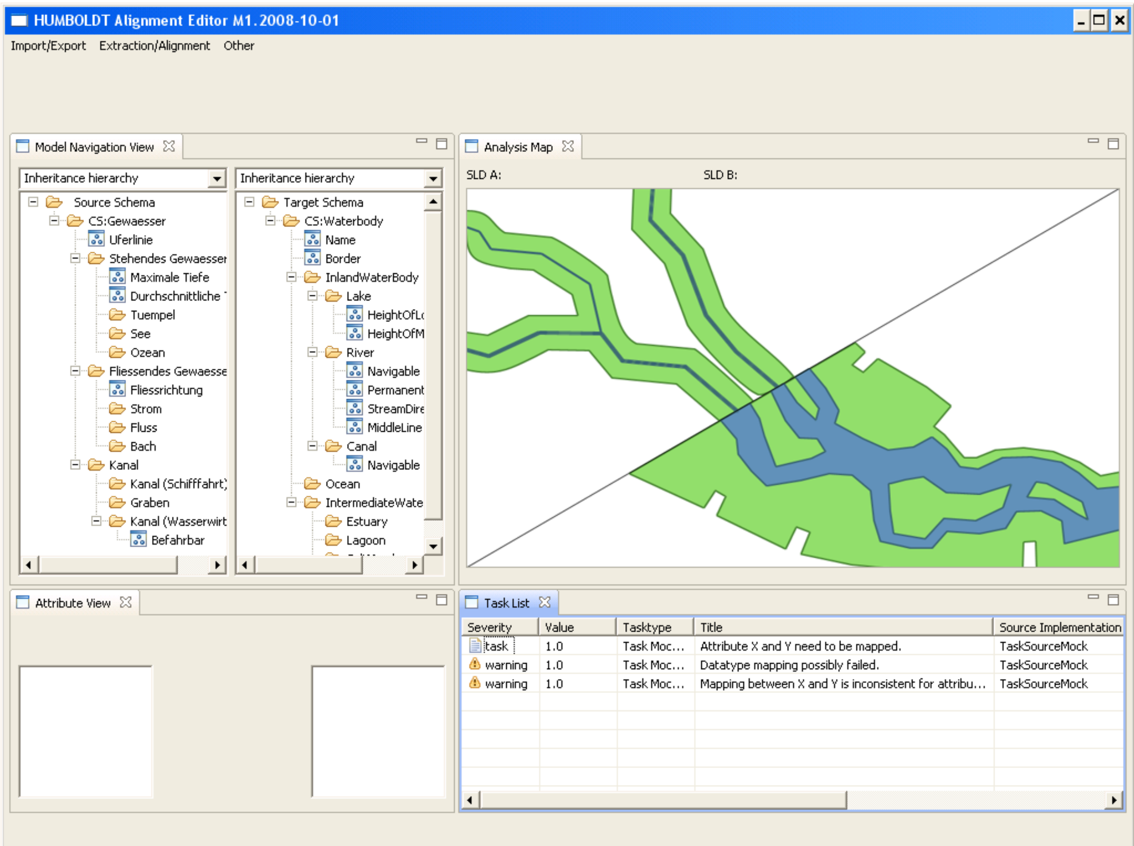


Figure 20: A screenshot of the Alignment Editor's main perspective

The term FeatureType is used synonymously to Spatial Object Type (INSPIRE wording) and Concept (Ontology Engineering).

Collaboration

- This component is part of the Harmonisation Toolkit.
- This component stores mappings using the Model Repository.
- This component makes use of the harmonised application schemas created using the HUMBOLDT Model Editor.

Additional Information

- This component addresses several of the high-level requirements of this project (GMES02, HUMB01, HUMB04).

3. Workflow Design and Construction Service Front-end

The Workflow Design and Construction Service front-end is used by data integrators to define basic workflows and to discover and register new processing capabilities.

Additional Information

The WSF is developed in an agile manner, in close cooperation with its users. It does therefore not have a full specification, but only a set of requirements. Also, some ideas have been collected in the Workflow Design and Construction Service Component Specification, such as the screen design shown in Figure 21 below. The thematic working group for it is lead by Telespazio.

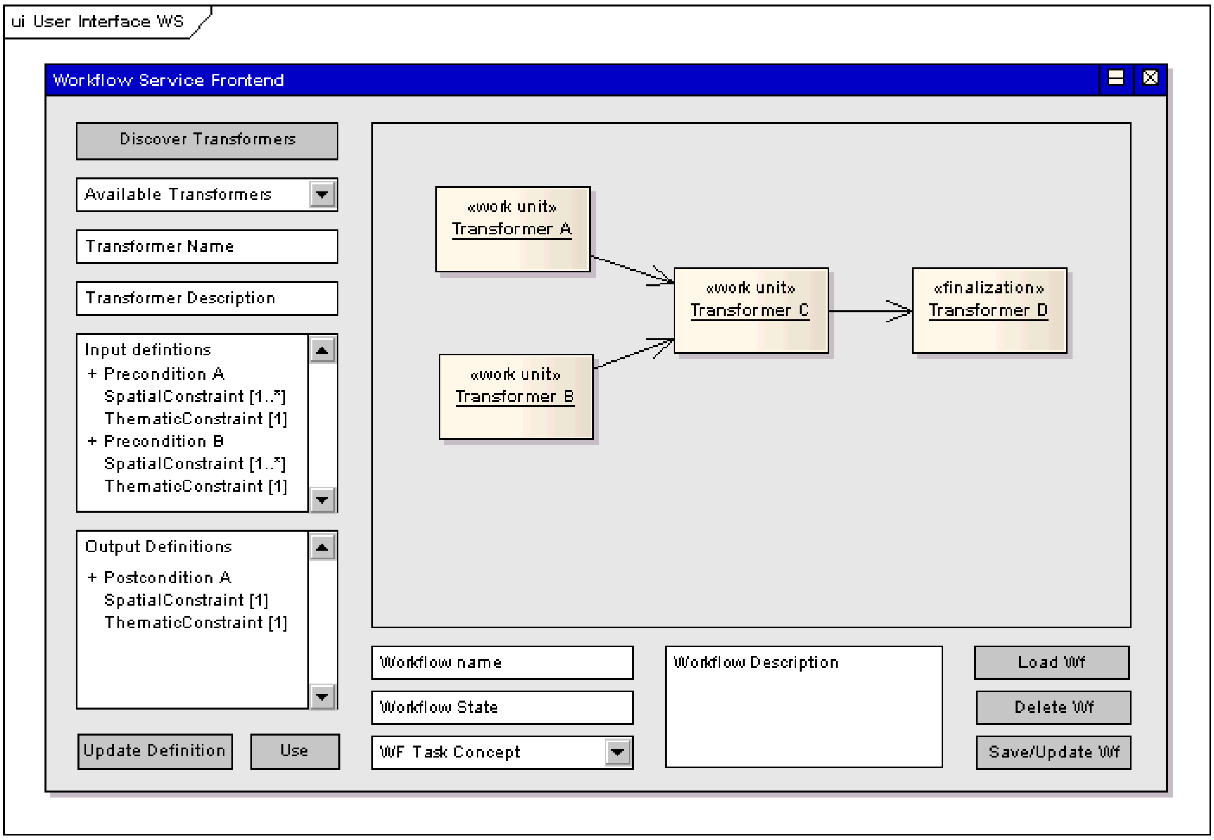


Figure 21: Possible layout of the Workflow Design and Construction Service Frontend, showing a view for the design of a basic (abstract) workflow

5.3.5 Data Transformation Services

The Data Transformation Services are a set of components with different processing capabilities that are invoked by the Mediator Service during the harmonization and transformation of a geodata set. They follow usual patterns for Web Processing Services and are developed in such a way that they can be deployed and invoked locally or distributed.A detailed introduction to the HUMBOLDT Processing Components General Model and the individual processing / transformation service can be found in A5.2-D3 [3.6].

Collaboration

- These components are being used by the Mediator Service in the execution of a workflow.
- These components are being registered for usage in workflow construction by the Workflow Design and Construction Service

1. Conceptual Schema Transformer

One of the main harmonisation problems is surely the integration of data coming from different structures. The Conceptual Schema Transformer is a Web Processing Service (WPS 1.0.0) that is able of applying a schema transformation to a source WFS datasets (Application Schema A) in order to provide a target datasets (Application Schema B). A schema mapping between schema A and schema B has to be defined in order to accomplish the transformation.

Collaboration

- This component uses mappings stored in the Model Repository.
- This component is invoked by the Mediator Service.
- This component makes use of the harmonised application schemas created using the HUMBOLDT Model Editor for validation purposes.

Additional Information

- This component addresses several of the high-level requirements of this project (INSPIRE03, HUMB04).

2. Language Transformer

The Language Transformer is capable of transforming/translating all information that becomes visible to a user from one language to another (and vice versa). Language transformation might also be necessary if user input is given at a language different from the one used in data processing. Thus, language transformation should include the translation of messages (user interface – both input and output) as well as that of data (e.g. attribute names and values). Besides “classical” language transformation, it would also be important to handle other issues such as conversion between different date/time formats (or different units of measurement).

According to this general view, the actual problem is similar to those dealt with in the scope of „internationalization” (i18n) and „localization” (or L10n) widely used in software development.

In some simple cases, language translation can be solved by a thesaurus service containing the meaning of different words in multiple languages. However, taking into account the complexity of the problem, it is more straightforward to apply an extensible solution. On the other hand, localization problems (e.g. transforming date and time formats, units of measures, etc.) cannot be solved by a simple word lookup and require appropriate handling.

Collaboration

- This component is invoked by the Mediator Service.

3. Multiple Representation Merger

The Multiple Representation Merger is capable of fusing Features of data sets with a spatial overlap, such as along a common border, where water bodies are part of both data sets. It contains an algorithm that uses the (vector) geometry of features to identify multiple representations of the same real-world feature. Based on this identification, the Features are merged, with one of the geometries being selected (the one with greater detail, i.e. point density) and all other attributes being copied. If no duplicates are found for a feature, it is always returned.

It is important to know that the service assumes both data sets to use the same application schema, otherwise it won't work. This means that the execution of a Conceptual Schema Transformer beforehand will sometimes be necessary. This is also depicted in the example for the Protected Areas scenario below.

Collaboration

- This component is invoked by the Mediator Service.

Additional Information

- The identification algorithms used are taken from the Java Conflation Suite (JCS).

4. Coordinate Transformation Service

The Coordinate Transformation Service allows to transform coordinates between various geographic reference systems, i.e. Geoids and projections.

The CTS has three core usage scenarios:

1. Input of a file that doesn't come from a service which provides CRS selection capability.
2. Requiring usage of a specific, non-standard algorithm with high precision, which cannot be reached by generic methods.
3. Usage of an exotic CRS that is not supported (or not well enough) by generic solutions.

Collaboration

- This component is invoked by the Mediator Service.

Additional Information

- The transformation is based on the Geoid and Projection Parameters database of the EPSG, and on the generic transformation algorithm provided by GeoTools 2.4. Any more specific algorithms can be plugged in on demand.
- We chose not to implement the OGC Web Coordinate Transformation Service (WCTS) specification so that we would only have to deal with one type of interface both in the Workflow and Mediator Service. Of course, a WCTS can easily be "hidden" behind a WPS façade.

5. Edge Matching Service

The Edge Matching Service (EMS) is a service that aligns edges and points of vector geometries so that they will be gapless. It has two modes of operation:

1. *Align-to-Reference:* In this case, all candidate data sets will be transformed so that points are moved up to a maximum distance also provided as input. As an example, consider a data set with high-quality country borders that a set of regional borders ha to use as reference. Another example would be river networks of different scale, where the higher-scale data can be used as reference.
2. *Distribute-Errors:* In this case, there is no reference data set that can be used as "ground truth", therefore all geometries will be transformed. Again, no point or edge will be moved further than a client-specified amount. This approach is useful for data sets of even quality and legal bindingness.

For each of the two modes of operation, the EMS has three core usage scenarios:

1. A data set should be without gaps, but there are small gaps in between the individual features that need to be filled.
2. Two data sets should have identical geometry over a shared feature (such as a common administrative border), but the geometry varies.
3. Two data sets have identical geometry for a shared feature, but the geometry is translated from the position where it should be located.

Collaboration

- This component is invoked by the Mediator Service.

Additional Information

- The EMS makes use of the Java Conflation Suite and of the Java Topology Suite.

5.4. Information Viewpoint

This chapter gives an overview of the basic set of concepts and rules addressing the scope and nature of information specifications. It defines the semantics of information as used within HUMBOLDT on a high level. Although different concepts and structures are specific of each component specification and they are detailed in the corresponding specification document, the most important concepts are briefly defined in this section:

5.4.1 Data structures used across multiple components

There are several data structures that have been defined that are used across multiple components. These form the core data models of the HUMBOLDT framework and are described in the following section.

1. Constraints

A constraint expresses a single rule that a data set or service has to fulfil in order to complete a task. This includes the HUMBOLDT services themselves as much as upstream, non-HUMBOLDT services. Constraints are persisted as part of contexts in the Context Service, and are used in the Mediator Complex Request.

2. Mediator Complex Request

The Mediator Complex Request (MCR) represents a combination of a user's query and his context. Thus, the MCR is the main message structure used in the HUMBOLDT framework when it comes to expressing requests. It is synthesized from the different logical and physical schemas used to express requests in the most common interface specification in the geospatial domain, such as WFS, WMS and WCS. The MCR has the following key elements that define it:

– *A Context*: The context document valid for this request. The context is the second main source for Constraints apart from the original request itself. It can either be a user-specific Context, and organization-specific Context, or a default Context if no information was included in the original request. All Constraints from the Context are merged with those from the original request, using precedence rules as defined by the system administrator.

– *A unique identifier:* This identification is used throughout the HUMBOLDT system to always be able to track back which request led to the creation of a sub-request or response. It can be used for authentication and communication aspects., such as relaying back the answer to the original requester.

3. Geospatial Workflow

A geospatial workflow is a chain of geospatial data and processing / transformation services. HUMBOLDT differs between *Basic Workflows*, representing abstract chains of geoprocessing functionality and Executable Workflows, incorporating all information necessary for execution. In HUMBOLDT, Basic Workflows are manually created by users and do not contain information on data services that deliver input. Executable Workflows are automatically constructed either out of a Basic Workflow or – if there is no Basic Workflow for a requested FeatureType – for the purpose of transforming/harmonising discovered data sources according to end user requirements. Executable Workflows contain all information necessary for execution, such as pointers to data services that deliver input.

Transformer:

Transformer are the building blocks of workflows. They are HUMBOLDT internal representations of geoprocessing functionality, which is usually implemented within OGC WPS. Transformer can be divided into those that transform data according to the HUMBOLDT constraint model, e.g. spatial reference system transformation, and those, that perform some calculation independent from the constraints, such as Overlay or Buffer operations.

Preconditions:

Preconditions of Transformers are constraints on the input of the processing that the Transformer (respectively the WPS process in the background) performs. For example, a precondition specifies a (GML-) schema on which the process executes. The purpose of preconditions are syntactic compatibility checks between Transformers. They e.g. prevent the user from chaining raster and vector processes pr processes that operate on different (GML-) schema.

Postconditions:

Postconditions represent constraints on the output of a Transformer. They have the same purpose as preconditions.

5.4.2 Additional data structures in the Model Repository

The Model Repository uses two additional data structures. The first one is used to store the conceptual schemas created in the HUMBOLDT Model Editor or any other tool able to create a conceptual schema in one of several formats.

For the definition of mapping rules between conceptual schemas, we are using the Ontology Mapping Working Group's Ontology Mapping Language (OMWG OML D7.1/2), which allows both to make assertions about relations between concepts (feature types) and allows to define transformation functions on an attribute level. We have profiled this language so that it also contains some geo-specific elements, such as queries that can be expressed using Filter Encoding or the OGC's CQL.

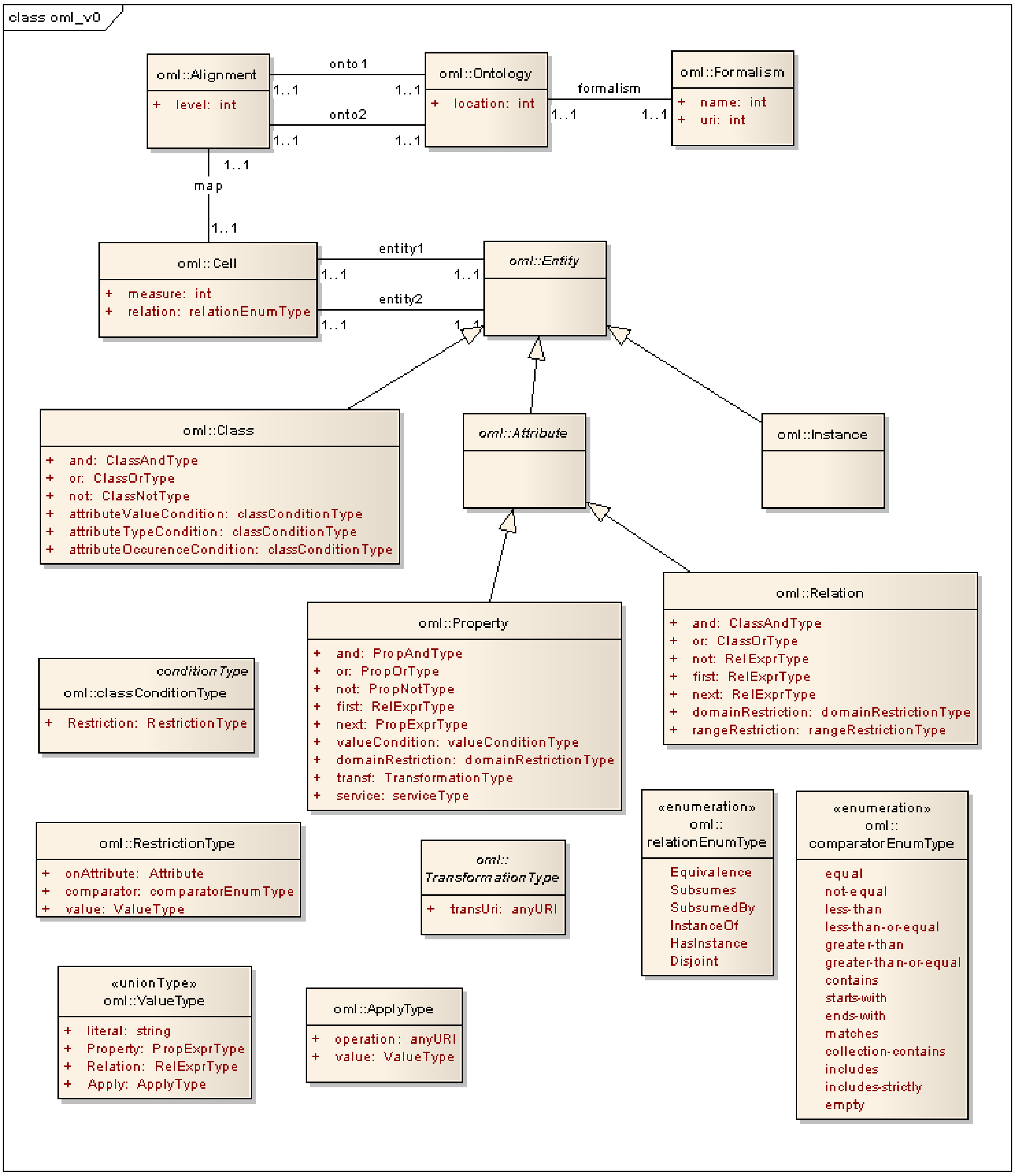


Figure 22: Main classes of the Ontology Mapping Language D7.2

For detailed information on the conceptual schema model and on the mapping/alignment mechanism used, please refer to the Model Repository Service Component Specification [2.3.3], chapter 4.

5.4.3 Additional data structures in the Information Grounding Service

The IGS internally makes use of two additional structures, specifically the Grounding Catalogue and a structure representing Grounding Services:

Grounding Catalogue:

The Grounding catalogue is a database with information about the geospatial entities available for a user community. The Grounding Catalogue stores the descriptive data (metadata) regarding the grounding services, not the own service. The interfaces which describe the catalogue are defined by the OGC standards - the Catalogue Service Web (CSW).

Grounding Service:

Geospatial information resources distributed on the Internet. They are the available services which provide an interface allowing requests for any subsets of a multi-dimensional and multitemporal geo-spatial data for a specific geographic region. They follow OGC specifications: WFS, WMS, WPS …

5.5. Engineering Viewpoint

The engineering viewpoint is primarily concerned with providing the mechanisms and functions for supporting distribution between objects in an open distributed system, whereas the Technology Viewpoint focuses on the concrete standards, tools ans libraries used to implement the specifications. Thus, the engineering viewpoint refers to the distribution of components and the services they provide.

As explained in the introductory section of chapter 4, one primary goal when coming to deployment was to isolate functionality from the context into which it is deployed, allowing usage of developed components in many different scenarios. The actual deployment to different environments can be as a (OGC or W3C) web service (usually as a web application), as a standalone application or even as a plug-in to an existing desktop GIS. To make this possible, concrete wiring of components is isolated as shown in Figure 23.

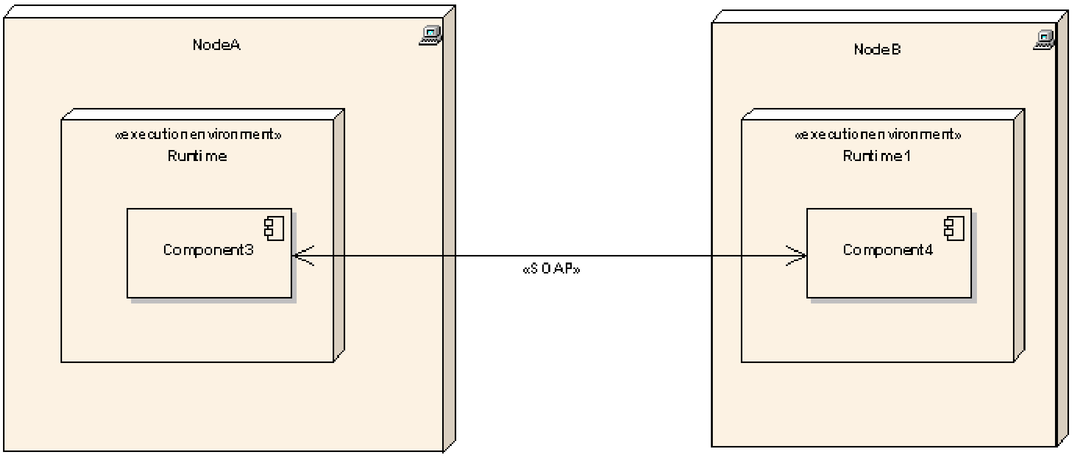


Figure 23: Example concrete wiring on technology level between two components

Furthermore, framework components are designed in such a way that they can be easily recombined into different services, as the following examples show.

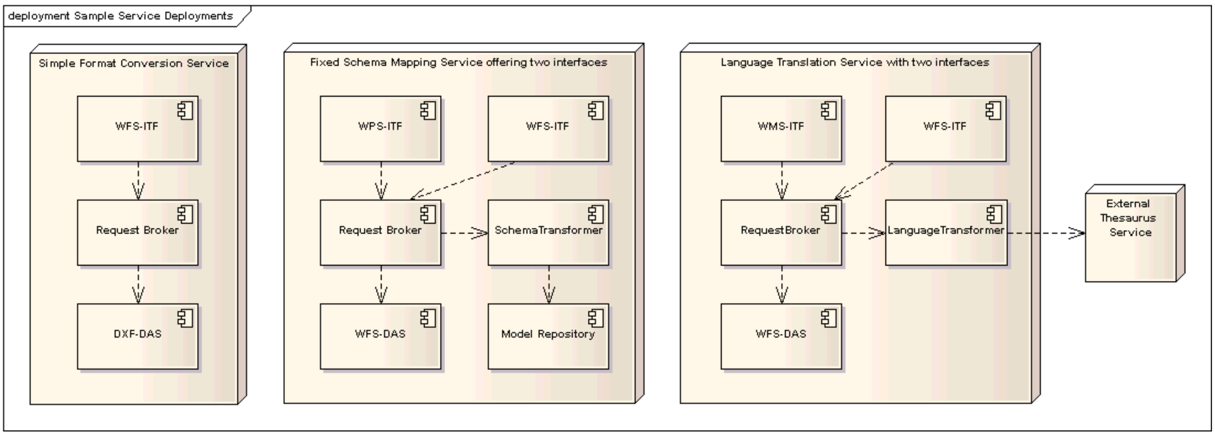


Figure 24: Three fixed-structure sample deployments.

Figure 24 shows three fixed-structure service deployments based on parts of the HUMBOLDT Framework:

– *Left*: A simple *Format Conversion Service* that translates DXF files to GML served via a WFS. Such a service can use multiple DASes and ITFs to provide more than one physical translation.

- *Middle*: A *Schema Mapping Service* that uses an internal Schema Transformer and an internal Model Repository. It exposes the transformation capabilities via WFS (+Context) and WPS and can read in WFS.
- *Right*: A *Language Translation Service* which uses an existing, non-HUMBOLDT Thesaurus Service and can either offer a map directly via WMS or just translated features via a WFS.

Such deployments of services can be used in a fixed structure or as part of a flexible structure, which is outlined below. Creating and deploying them is easily possible for a developer requiring a specific transformation service or application.

In addition to fixed-structure deployments with predefined capabilities, the Framework Components can also be used to build a flexible transformation environment, creating a highly generic harmonization service that can adapt itself to different user needs. The flexible deployment uses Workflow capabilities to discover and combine concrete transformation capabilities into a complex transformation service chain.

Such a deployment can be used for harmonisation, processing and general transformation needs of an organization. The known transformation capabilities, independent of their actual deployment as Web Processing Services or as in-memory components, are combined to a transformation workflow

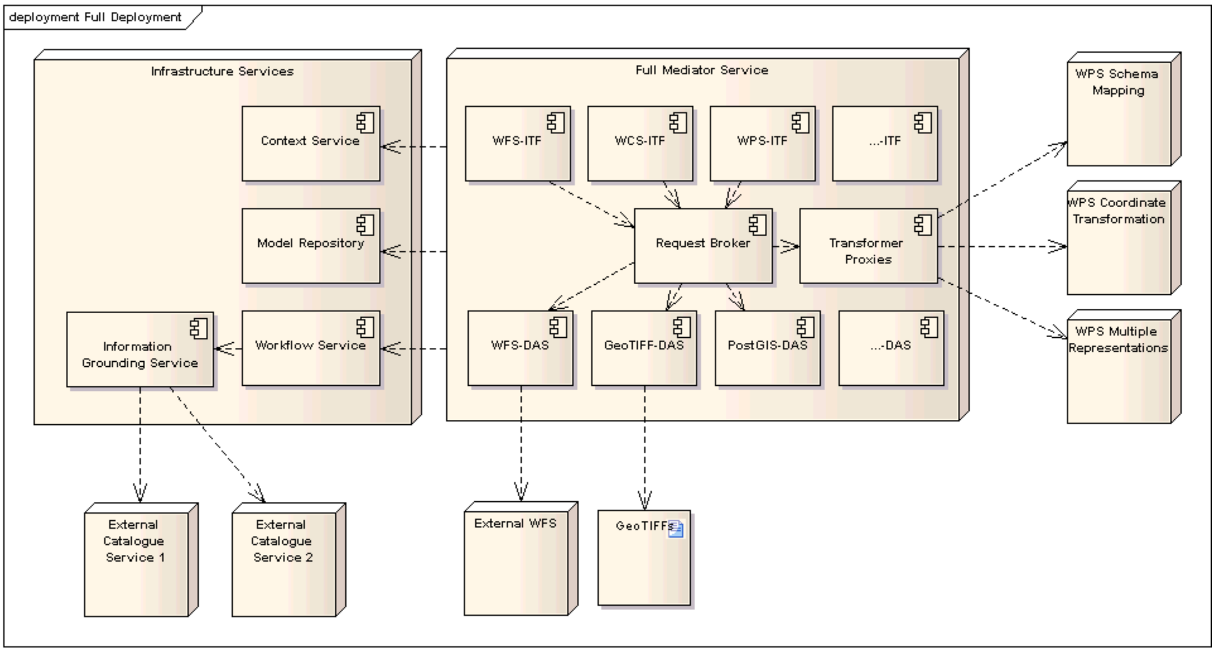


Figure 25: Deployment of the full HUMBOLDT framework into two nodes

that harmonizes resources exactly to the product definition provided by the user. By separating the Infrastructure Services, the Mediator Service and the actual processing services, the deployment can also be scaled efficiently. In another scenario, all these framework components can also be packaged into a single (web) application.

The component diagram in Figure 26 below provides an example of how the service interactions between components can be illustrated.

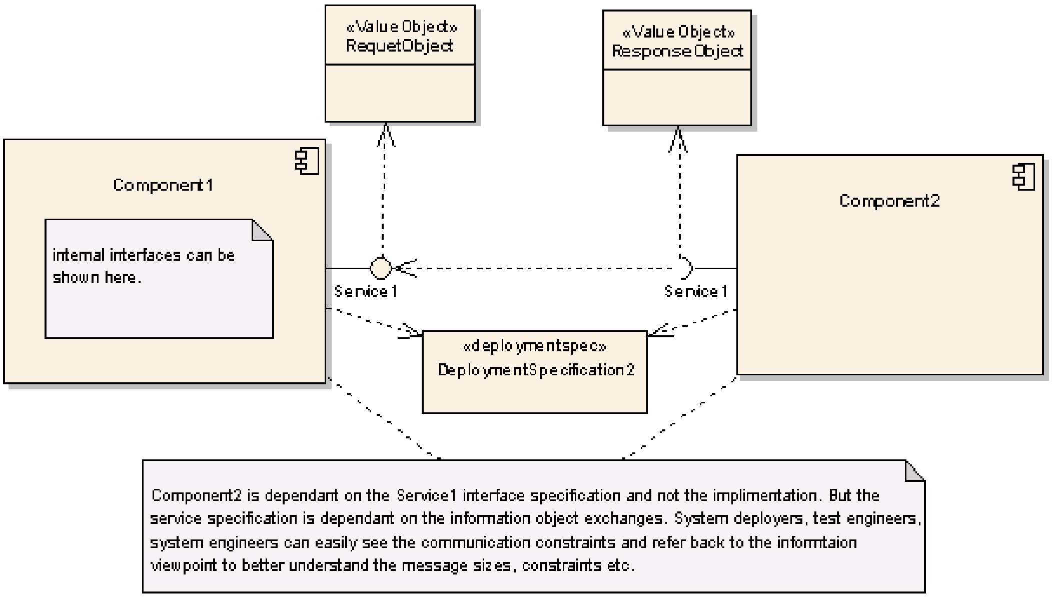


Figure 26: Binding of components via interfaces, not implementations

The approach of handling the linkage of services to deployment distribution with other components and as such are extremely relevant to system deployment experts who are not concerned with detailed composition of the service.

5.5.1 Distribution and Deployment

Detailed rules on distribution and deployment of the diverse applications and services can be found in the build process guidelines published by WP08. To summarize, all elements required for building and deploying need to be found in the Subversion repository, including binary files and build scripts.

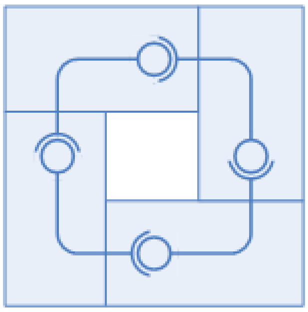
5.6. Technology Viewpoint

As explained above, the technology viewpoint focuses on concrete technologies to use. For the HUMBOLDT framework development, the following technologies have been selected as the working base, starting from the Handbook of Standards and the State of the Art deliverables. Deviations are possible but have to be agreed on on the Architecture Team.

Category	***Technology***	***Applicability***	***Bindingness***
Language	Java 1.5	Standard language to use for all service component implementations.	Mandatory
Platform	Java 1.5	Standard platform to use for all service component implementations.	Mandatory
Language	C#	Optional language to use for toolbox and scenario applications	Optional
Platform	.NET 3.0	Optional platform to use for toolbox and scenario applications	Optional
Library	GeoAPI 2.2	Standard library to use in specification of interfaces	Mandatory
Library	GeoTools 2.5	Standard library to use for implementation classes of the GeoAPI.	Optional
Container	Apache Tuscany SCA 1.0	Standard container to use when a Service Component is deployed.	Optional
Container	Apache Tomcat 5.5	Standard container to use for simple service deployments.	Optional
Library/Container	Spring Framework, Spring-WS	Standard container to use for service deployments.	Mandatory

Table 5: Technologies selected as a baseline for component and application implementation

6 Summary & Outlook



In M2.0, our focus was put on the Harmonisation toolkit and on concrete processing services required for data harmonisation. In M3.0, the specifications have now been completed based on the requirements derived from the HUMBOLDT scenarios. Further, the specification documents have been made more explicit where needed, involving feedback from WP8 and application contacts and a lightweight high-level introduction to the HUMBOLDT software has been provided (A5.2-D3 [3.0]).