Title:
A5.2-D2 [2.1] HUMBOLDT Commons Specification
Author(s)/Organisation(s):
Ana Belén Antón / ETRA
Working Group:
Architecture Team/WP05
References:
A5.2-D2 Framework Prototype Specification

Short Description:
This document describes the specification of the HUMBOLDT Commons as part of the HUMBOLDT software framework. For an overview of the entire framework v2.0 description, please refer to the main specification document A5.2-D2. This component is not an actual service component like the other HUMBOLDT components which satisfy functional requirements, but represents their common Data Model. Therefore HUMBOLDT Commons integrates all core data structures used in more than one HUMBOLDT component. It means the Data Model of the HUMBOLDT Framework.
Keywords:
Framework specification, logical architecture, physical architecture, common data structures.

History:

Version	Author(s)	Status	Comment
001	Ana Belén Antón	New	Initial specification of framework V2.0, template
002	Ana Belén Antón	rfc	New Commons after some requests
003	Ana Belén Antón	Rfc	Changes according to the ContextService and DQME specifications and requests. Diagrams updated.
004	Ana Belén Antón	Rfc	Updating the Transformer and Workflow description. Minor editing changes.

Table of contents

Figures

Tables

1 Introduction

1.1 Purpose of this document

This component is not an actual service component like the other HUMBOLDT services which satisfy functional requirements, but represents their common Data Model. The HUMBOLDT Commons integrates all core data structures used in more than one HUMBOLDT component for information exchange and it does not have visible functionality for the HUMBOLDT stakeholders. Therefore only the *Information Viewpoint* joint to the *Introduction* will be available in this specification document.

1.2 Abbreviations and Definitions used in this document

This section summarizes all abbreviations which are used specifically for this document. It collects several kinds of abbreviations to provide a single point of reference, including names of modules, protocols, services, standards and tools. If any general abbreviation is not found here, please see Deliverable A5.2-D2.

Abbrev.	***Name***	***Definition***
CS	Context Service	Service component of the HUMBOLDT Framework in charge of the management and provision of the specific contexts. This component consists of two modules: the Context Management Service and the User Management Service. See the document A5.2-D2 [2.4]
DQME	Data Quality Measurement and Evaluation	Service component of the HUMBOLDT Framework in charge of the measurement and evaluation of the data quality. See the document A5.2-D2 [2.7]
MCR	Mediator Complex Request	The MediatorComplexRequest is a data structure used internally by the Mediator Service. It is basically a container for the Constraints and Parameters collected from a User's request by the Context Service.
MS	Mediator Service	Service component of the HUMBOLDT Framework in charge of the management of the actual execution of a transformation and to provide the user with some support in specifying his product definition. See the document A5.2-D2 [2.2]
WCS	Web Coverage Service	International standard defined by the OGC for providing interoperable access to geospatial coverages, such as satellite images, digital aerial photos, digital elevation data, and other phenomena represented by values at each measurement point.
WDCS	Workflow Design and Construction Service	Service component of the HUMBOLDT Framework in charge of the creation of geospatial workflows based on web service technology. See the document A5.2-D2 [2.5]

Abbrev.	*Name*	*Definition*
WFS	Web Feature Service	International standard defined by the OGC for retrieving and manipulating geographic features across the web.
WMS	Web Map Service	International standard defined by the OGC for requesting geo-registered map images from one or more distributed geospatial databases using a simple HTTP interface.
WPS	Web Processing Service	International standard defined by the OGC for requesting the execution of geospatial processes, such as polygon overlay and know how the output from the process is handled.

Table 1: Abbreviations used in the description of the component

This section gives a few definitions valid for the context of this document. As in the section before, if any general definition is not found here, please see Deliverable A5.2-D2.

Context	A Context is defined as the specific user requirements related to the environment in which his/her business operates when they are retrieving geo-information. See chapter 2.2.
Grounding	Groundings represent the geospatial information resources distributed on the Internet, which follow the OGC standards such as WFS, WMS, WPS…
Transformer	A Transformer is a processing artifact to transform data into the required state that was requested by the user. See chapter 2.4.
Workflow	Abstract descriptions of chains of geoprocessing functionality. See chapter 2.5.

Table 2: Definitions used in the description of the component

2 Information Viewpoint

This chapter describes the information viewpoint of the HUMBOLDT Framework v2.0. This covers various data structures that are being used for storage and exchange between some of the HUMBOLDT components defined in that version.

2.1 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 (CS), and are used in the Mediator Service (MS) for the Mediator Complex Request, in the Workflow Design and Construction Service (WDCS) and in the Data Quality Measurement and Evaluation (DQME).

Constraints are used to describe and store different kind of information like metadata, geospatial data, etc.

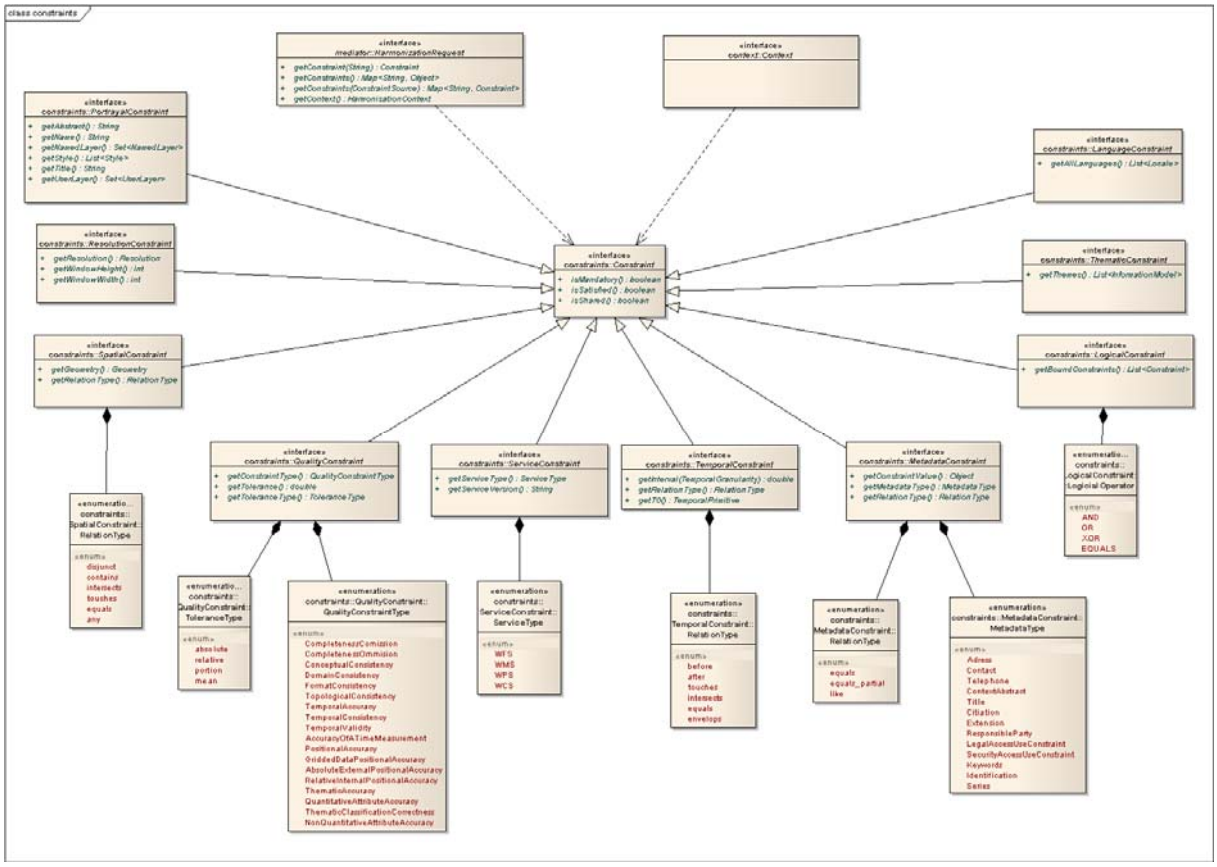


Figure 1: Information Model for the Constraint interface

As a note, to point out that one common attribute between the constraints if it is a constraint is *shared* or *not.* It is used for identifying the interdependencies between input and output data when one transformation process is being designed. For more explanations, please see the Workflow Design and Construction Service (WDCS) specification document.

2.1.1 Language Constraint

The Language Constraint allows the user to express his preferences for the natural language that the data sets should be presented in. For this purpose, ISO 639-2 and ISO 639-3 language codes may be used.

2.1.2 Logical Constraint

Logical Constraints, such as AND, OR, XOR and EQUALS can be used to test 2...* constraints according to the logical operator that was specified. With the logical constraint any kind of constraint can be combined.

2.1.3 Metadata Constraint

This type of Constraint allows testing for conformance to other metadata elements than those which have explicit constraint types, such as responsible party or certain identification such as keywords.

2.1.3.1 MetadataType of the MetadataConstraint

The current scope of this specification covers the following metadata types to be filtered:

- Title (ISO19139: title)
- ContextAbstract (ISO19139: abstract)
- Keywords (ISO19139: keywords)

2.1.3.2 RelationType of the MetadataConstraint

The following RelationTypes are defined and can be used:

- equals
- equals_partial
- like

For the request to the Grounding Catalogues these RelationTypes can be rebuild as Filter Encoding.

2.1.4 Portrayal Constraint

A Portrayal Constraint allows access to the Style Inforamtion details like Named Layer, Named Style etc.

2.1.5 Quality Constraint

This constraint allows the definition of quality parameters that the data in question will need to fulfill. For this, ISO 19113 quality definitions have been used. These are also encoded into ISO 19115 (Metadata). However these metadata has been extended and the complete list of Quality elements is:

- CompletenessCommision: excess data presence in the dataset described by the scope.
- CompletenessOmmision: data absent from the dataset as described by the scope.
- ConceptualConsistency: adherence to the rules of the conceptual schema.

- DomainConsistency: adherence of values to the value domains.
- FormatConsistency: the degree to which data is stored in accordance with the physical structure of the dataset as described by the scope.
- TopologicalConsistency: correctness of the explicitly encoded topological characteristics of the data as described by the scope.
- TemporalAccuracy: accuracy of quantitative attributes and the correctness of non-quantitative attributes and of classification of such features and their relationships.
- TemporalConsistency: correctness of ordered events or sequences if reported.
- TemporalValidity: validity of data specified by the scope with respect to time.
- AccuracyOfATimeMeasurement: correctness of a temporal measurement of an item (reporting of the error in time measurement).
- PositionalAccuracy: accuracy of the position of features.
- GriddedDataPositionalAccuracy: closeness of reported coordinate values to values accepted as being true.
- AbsoluteExternalPositionalAccuracy: closeness of the absolute positions of features in the scope of their respective absolute positions accepted as being true.
- RelativeInternalPositionalAccuracy: closeness of the relative positions of features in the scope of their respective relative positions accepted as being true.
- ThematicAccuracy: accuracy of quantitative attributes and the correctness of non-quantitative attributes and of classification of such features and their relationships.
- QuantitativeAttributeAccuracy: accuracy of quantitative attributes.
- ThematicClassificationCorrectness: comparison of classes assigned to features or their attributes to a universe of discourse.
- NonQuantitativeAttributeAccuracy: accuracy of non-quantitative attributes.

2.1.6 Resolution Constraint

A Resolution Constraint allows definition of the Ground Sample distance to use and the definition of the size in which an actual map product is to be put out.

2.1.7 Spatial Constraint

A Spatial Constraint is used to express a region of interest; this is a *spatial filter*. The most commonly used SpatialRequest contains just a bounding box, but this constraint also allows for more complex geometry. It corresponds to the OGC Filter Encoding for spatial operators.

2.1.8 Temporal Constraint

A Temporal Constraint is a special metadata constraint used to express a temporal boundary for the geoinformation to be used. It contains a temporal filter reporting both data validity according to time and correctness of ordered events or sequences. As relation types only “Before”, “Equals” and “After” are allowed. In a combination of both, the temporal extent can be matched to a period of time.

2.1.9 Thematic Constraint

This type of constraint expresses what kind of information is to be retrieved for a certain request. The Thematic Constraint is therefore expressed as a set of concepts from one or multiple Information Models. As an example, consider the following: Somebody is requesting a map product with five layers. Out of these five layers, three are basic geoinformation, such as a shared DGM, the road network and the parcels. In addition, he requests two layers from more specific thematic domains, such as the zoning plan and a plan of protected ground water areas. This would represent three Concepts from one Information Model and two from another Information Model. Consequently, getThemes() would return a list containing two Information Model, one containing two Concepts, the other containing three Concepts.

2.1.10 Service Constraint

A Service Constraint is used to filter specific OGC services and/or service versions (e.g. filter to OGC WMS with version 1.1.1).

2.2 Context

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. The Context is built when the HUMBOLDT stakeholders are creating their specific product definition. Please see the Context Service document (A5.2-D2 [2.4]).

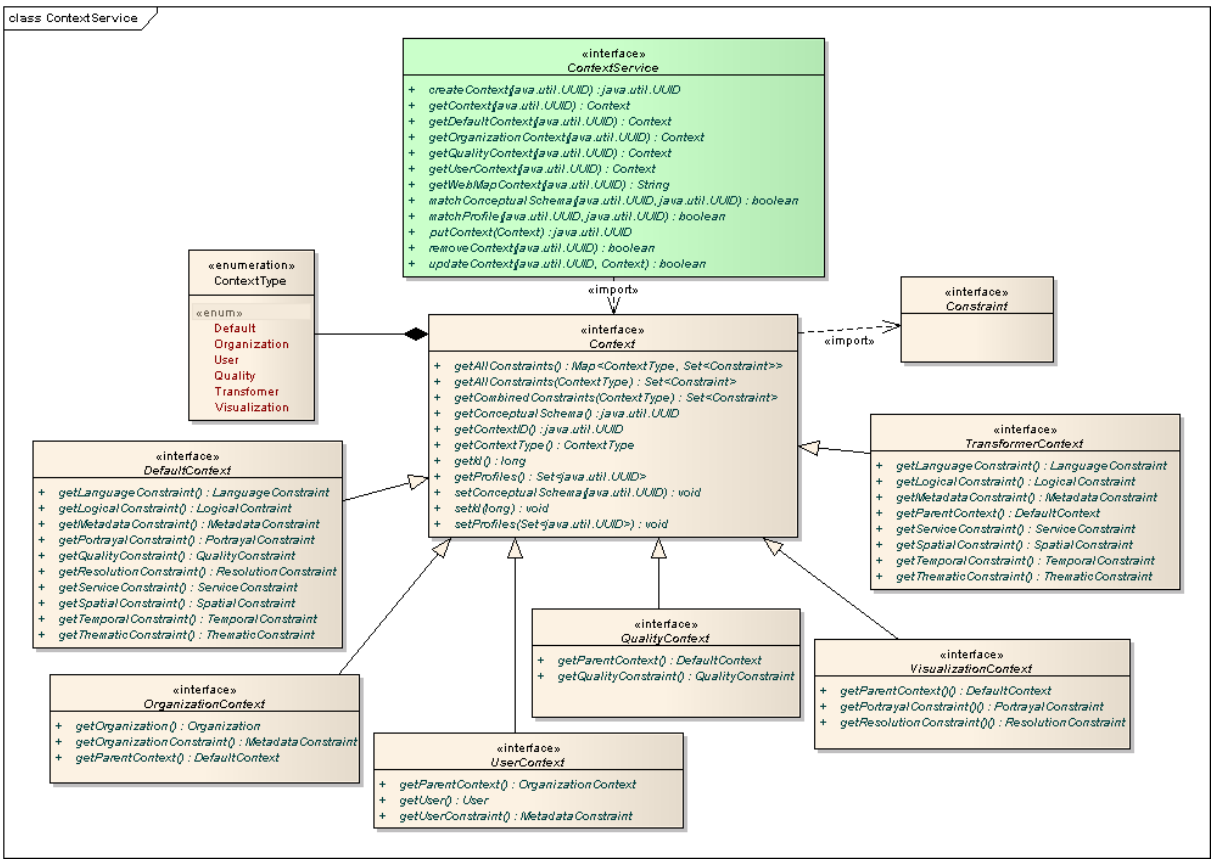


Figure 2: Information Model for the Context

Apart from the three context types named previously, there are three more which are defined as part of a Context. They are specific contexts depending on their purpose. Therefore there are six types of contexts defined:

- **Default:** the standard set of attributes defining the common context for a specific domain or application, giving access to general preferences like language, resolution…
- **Organization:** the refinement of a Default Context for its use in a specific organization.
- **User:** the refinement of a Default Context for the specific use of the client.
- **Quality:** the QualityConstraint, as part of a Context.
- **Transformer:** it provides the subset of constraints related to transformers, as part of a Context.
- **Visualization:** it only provides the constraints related to visualization, as part of a Context.

2.3 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 used by the Mediator Service and the Workflow Service. 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, WPS and WCS, and is enriched by information from a Context document.

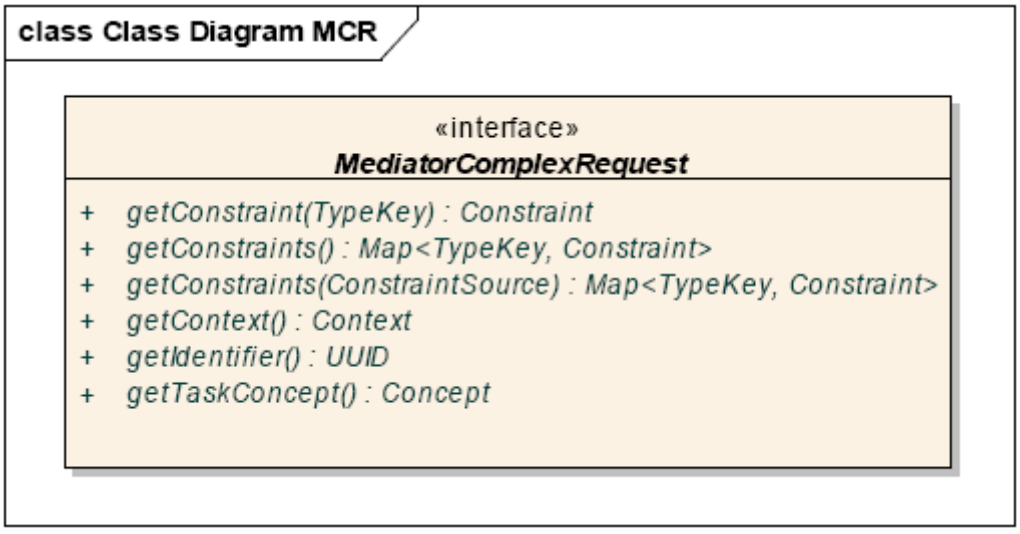


Figure 3: The core MCR interface

The MCR has the following key elements that define it:

- *A Set of Constraints:* Each Constraint can be understood as a rule that the result to be returned to the requesting client has to fulfill. There are very different types of Constraints, as detailed in the section 2.1.

- *A Context:* The context valid for this request. The context is the second main source for Constraints apart from the original request itself. All Constraints from the Context are merged with those from the original request, using precedence rules as defined by the system administrator. Please see section 2.2
- *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.
- *A Task Concept:* This element is used to identify the operation a user requested. A task concept is in the first line a representation of the goal that can be achieved by calling a certain operation on one of the interfaces offered by the Mediator Service – it is thus dependent on the operation and the interface that was invoked, such as GetMap on a WMS. In addition, the task concept can be refined further by making the selection of a task concept also depend on request parameters. As an example, for the WMS GetMap operation with a file type of img/jpeg, the matching task concept would be GetStyledRasterMap. All Task Concepts together form the System Task Taxonomy (STT). The approach of using a System Task Taxonomy allows the definition of basic workflows that can be extended and modified for more specific tasks, adopting strengths of the object-oriented programming paradigm for workflow management and execution. For more information on the STT and the Task Concept, please refer to the Workflow Service Component Specification.

2.4 Transformer

A Transformer is a HUMBOLDT internal representation of processing functionality that is either encapsulated within an OGC Web Processing Service implementation or directly implemented within the HUMBOLDT framework. It is a processing artifact used during mediation to transform data into the required state that was requested by the user.

Transformers are the processing nodes in a workflow that wrap the transformation functionalities required during harmonization. Transformers use other processing services e.g. WPS to be able to accomplish the transformations. Thus a Transformer is a super-interface for all algorithm implementations which perform transformation. A Transformer also implements WebProcessingService interface, which has methods for execution of the Transformer. Each Transformer implementation is required to provide metadata that describes the processing it applies so that a full lineage for a transformed dataset can be maintained.

Transformers have inputs, outputs and descriptions of these inputs and outputs. A Transformer may be capable of executing, for instance, a buffering process given a set of inputs, and providing certain output. However, the actual buffering functionality can be offered by an OGC WPS service that might in turn require that data be transformed into a given coordinate system before it is used as input, in which case additional Transformer that provides this functionality can be used.

Depending on the functionality the transformers implement, different transformer types have been defined:

- **Harmonisation Transformers:** Implement harmonization transformations, such as the spatial reference system transformation, language transformation, …
- **GIS Transformers:** Implement well-Known GIS transformations, such as Buffer, Dissolve, Overlay…

- **Application specific Transformers:** Implement specific transformations within a certain application area, such as a climate change model.

These two last groups can be also referred to as **Non-harmonisation Transformers.**

Figure 4 shows the Transformer model joint to the Workflow.

2.5 Workflow

A Workflow is an aggregation of Transformers together with Groundings definition (please see Figure 4) in order to achieve a certain task which can not be achieved by single data or a single process. Workflows provide abstract descriptions of chains of geoprocessing processes and all information necessary for their execution.

From an internal perspective, a Workflow is also a Transformer and can thus be handled just like a simple Transformer by the Mediator Service. This structure it is used by the MS and the WDCS.

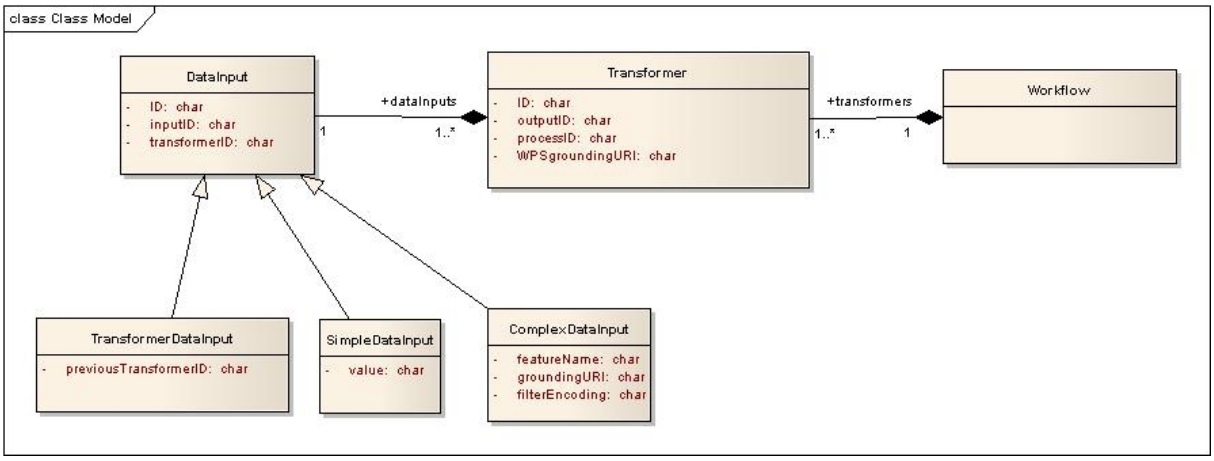
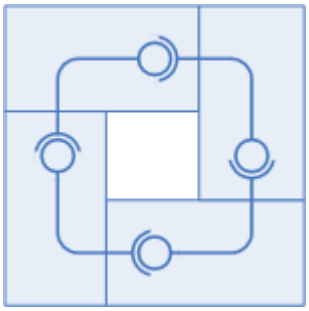


Figure 4: Workflow model

The Workflows according to that model contain everything needed for executing the workflow, i.e. pointers to data services (Groundings), all Transformers (their URIs, the process identifiers) and all the connections between the Transformers.

3 Summary & Outlook



This specification provided an overview of the Information Viewpoint of the HUMBOLDT Framework. It is the data structures shared by different services defined in HUMBOLDT.

In previous versions of the specification, this “component” was not defined, but the lessons learned in the version 1.0 of the specification have demonstrated that common structures must be organized in order to share information in a common manner within the HUMBOLDT Framework.

In this document the Constraints structure has been extended satisfying different needs of some services and for next versions the refinement of Transformers and consequently of Workflows will be necessary once the processing group be more mature.