

**Title:**

A5.2-D3 [3.4] Context Service Component Specification

**Author(s)/Organisation(s):**

Daniel Fitzner/FHG-IGD, Ana Belén Antón/ETRA

**Working Group:**

Architecture Team/WP05

**References:**

- A5.2-D3 [3.0] A Lightweight Introduction to the HUMBOLDT Framework V3
- A5.2-D3 [3.1] Specification Introduction and Overview V3
- A5.2-D3 [3.2] Mediator Service Component Specification
- A5.2-D3 [3.5] Workflow Design and Construction Service Specification
- A5.2-D3 [3.7] Information Grounding Service Component Specification
- A5.3-D3 Humboldt Commons Specification / Framework Common Data Model V3

**Quality Assurance:**

- Review WP Leader: Thorsten Reitz (FhG)
- Review dependent WP leaders:
- Review Executive Board:
- Review others: Moses Gone (FhG)

**Delivery Date:** 30.11.2009

**Short Description:**

This document describes the specification of the Context Service as part of the HUMBOLDT software framework. For an overview of the entire framework v3.0 description, please refer to the main specification document A5.2-D3 [3.0] and [3.1]. The Context Service component specification follows the RM-ODP (ISO 10476), and is aimed at providing information on the responsibilities and collaborations with other components of the service component described herein.

The Context Service is a service component that provides context and user management for HUMBOLDT end users. It will also provide management of harmonisation and business constraints definition.

**Keywords:**

Framework specification, logical architecture, physical architecture, requirements, use cases, context service, user management.

<b>History:</b>			
Version	Author(s)	Status	Comment
001	Ana Belén Antón	new	Initial draft of the document V2.0... structure of the document and initial draft of the Enterprise Viewpoint
002	Ana Belén Antón	rfc	Improvement of the Enterprise Viewpoint. Scenario integration and requirements are reviewed.
003	Ana Belén Antón	rfc	Computational and Information Viewpoint updated.
004	Ana Belén Antón	rfc	Text updated after Daniel Fitzner revision. More explanations and examples given where they were required.
005	Daniel Fitzner	rfc	Specification updated for version 3. WSDL included. Computational Viewpoint refined, scenario integration aligned to Spec. Intro & Overview document.

## Table of content

<b>1</b>	<b><i>Introduction</i></b>	<b>5</b>
1.1.	<b>Purpose of this document</b>	<b>5</b>
1.2.	<b>Abbreviations and Definitions used in this document</b>	<b>5</b>
1.3.	<b>Standards used in this document</b>	<b>6</b>
1.3.1	OGC Web Map Context (OGC 05-005)	7
1.3.2	W3C Web Service	7
<b>2</b>	<b><i>Enterprise viewpoint</i></b>	<b>8</b>
2.1.	<b>Requirements</b>	<b>8</b>
2.1.1	Requirements on elements of a harmonization product	8
2.1.2	Requirements on the Context Service	9
2.2.	<b>Actors in this component</b>	<b>10</b>
2.2.1	End Users	10
2.2.2	Data Custodians (Data Providers)	10
2.2.3	Data Integrators (Service Providers)	10
2.3.	<b>Business process overview</b>	<b>11</b>
2.4.	<b>Scenario Integration</b>	<b>12</b>
<b>3</b>	<b><i>Computational viewpoint</i></b>	<b>14</b>
3.1.	<b>How is a Context used?</b>	<b>14</b>
3.1.1	Discovery	14
3.1.2	Retrieval	15
3.1.3	HUMBOLDT Contexts vs. OGC Web Map Contexts	15
3.2.	<b>Logical Architecture of the Context Service</b>	<b>17</b>
3.2.1	Description of the Context Management Service	17
3.2.2	Description of the User Management Service	19
<b>4</b>	<b><i>Information Viewpoint</i></b>	<b>22</b>
4.1.	<b>User Management Service</b>	<b>22</b>
<b>5</b>	<b><i>Summary &amp; Outlook</i></b>	<b>23</b>
<b><i>Annex....</i></b>		<b>24</b>
<b>Annex A: Context Service WSDL</b>		<b>24</b>

## Figures

Figure 1: Context Service Business Process .....	11
Figure 2: The role of the Context Service when building the capabilities .....	14
Figure 3: The role of the Context Service in data retrieval.....	15
Figure 4: Component diagram for the Context Service.....	17
Figure 5: Interface of the User Management Service .....	21
Figure 6: Complete information model for the UMS.....	22

## Tables

Table 1: Abbreviations used in this document .....	6
Table 2: Definitions of terms used in this document .....	6
Table 3: Functional Requirements for this component.....	9
Table 4: Non-Functional Requirements for this component.....	10
Table 5: API descriptions of the Context Service.....	19
Table 6: API descriptions of the User Management Service.....	20

# 1 Introduction

## 1.1. Purpose of this document

This document contains the specification of the Context Service, a component of the HUMBOLDT Framework (as such part of Deliverable A5.2-D3). This document also contains information on the internal data models and message structures used by this service component.

The Context Service is a service component that provides context and user management/preferences information for HUMBOLDT end users. It will also provide management of harmonisation and business constraints definition. For a more detailed description of the contexts maintained and administrated by the context service, please refer to the document *A5.3-D3 Humboldt Commons Specification / Framework Common Data Model V3*.

## 1.2. Abbreviations and Definitions used in this document

This section summarizes all abbreviations used and the valid definitions within the description of the Context Service component.

<b>Abbrev.</b>	<b>Name</b>	<b>Definition</b>
ASP	Application-Specific Product	It refers to a product designed for a specific domain or scenario. For example, in the User Story "Planning and management of a protected area" the ASP is a web portal where all geo-information of a concrete regional park in Italy is sharable with various external administration and decision levels, also this information is provided to people working in the park and citizens. Each of them access to the information with different purposes.
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.
MS	Mediator Service	HUMBOLDT Workflow Execution engine that is responsible for executing transformation definitions. Contexts are exchanged between Context Service and Mediator Service.  Specified in A5.2-D3 [3.2] Mediator Service Component Specification
CMS	Context Management Service	Service offered by the Context Service Component to provide basic and derived context information. It is enriching OGC-standards interfaces with specific harmonisation requirements which define the final product.
PA	Protected Areas	One of the scenarios defined in HUMBOLDT.

<b>Abbrev.</b>	<b>Name</b>	<b>Definition</b>
UMS	User Management Service	Service offered by the Context Service Component to offer user management capability and control over a whole set of attributes for each user. Contexts will be available depending on these attributes. It means that if a user is member of one organization and a specific context has been designed for this organization, the context will be available for him. Furthermore contexts will be matched with user groups but also with individual users.
USP	Unique Selling Proposition	It refers to any aspect of an object that differentiates it from similar objects.
WSDL	Web Services Description Language	An XML-based description language for describing the interface of web services.

**Table 1: Abbreviations used in this document**

<b>Concept</b>	<b>Definitions</b>
Context	A Context is a set of constraints on geospatial data sets. See <i>A5.3-D3 Humboldt Commons Specification / Framework Common Data Model V3</i> for a detailed description of the constraint types.
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.
WSDL	The Web Services Description Language (WSDL) is an XML-based language that provides a model for describing Web services. A client program connecting to a web service can read the WSDL to determine what functions are available on the server. Any special datatypes used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the functions listed in the WSDL.

**Table 2: Definitions of terms used in this document**

### 1.3. Standards used in this document

In this section, standards specific to the Context Service are listed and described shortly and why they are used.

### **1.3.1 OGC Web Map Context (OGC 05-005)**

A Web Map Context is a XML-document that unambiguously describes “the state of a WMS Client application in a manner that is independent of a particular client and that might be utilized by different clients to recreate the application state” (see OGC Web Map Context Specification). Such documents include e.g. the list of layers on the map, with each of them including a pointer to the WMS that provides it. WMS Clients that implement / support that specification can exchange such context files and then display exactly the same map to users as the WMS client who created the context file.

### **1.3.2 W3C Web Service**

A Web service is defined by the W3C as "a software system designed to support interoperable Machine to Machine interaction over a network." Web services are frequently just Web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

The implementation of the ContextService as a Web service simplifies the development of client software, because it is based on a standardized SOAP protocol. This also enables a simple integrating of the ContextService in a distributed system.

## 2 Enterprise viewpoint

The Context Service is a service component that provides context and user management for HUMBOLDT end users. It will also provide management of harmonisation and business constraints definition.

On one hand, the Context Service manages user/organisation profiles and their harmonisation requirements / products. On the other hand, when an end user, who is using a GIS client, requests geodata, the Context Service provides the context information (the constraints which were imposed by the user) to the component/s in charge of managing the geodata transformation process.

### 2.1. Requirements

This section gives the scenario requirements for elements of harmonization products (Section 2.1.1) as well as functional requirements on the Context Service itself (Section 2.1.2).

#### 2.1.1 Requirements on elements of a harmonization product

Many of the requirements that were collected from the HUMBOLDT scenarios directly indicate the need for users to define a harmonisation product. The requirements have been collected and clustered, resulting in the following set of requirement.

*The SYSTEM shall allow a USER to define a harmonisation product using the following parameters:*

**HP01:** Spatial (CRS, Geonames/Gazetteer, BB 2D/3D+, complex query geometries based on stored features)

**HP02:** Temporal (TRS, start, end, interval resolution, runtime and forecast time)

**HP03:** Licenses and Prices

**HP04:** Thematic (Schema->FeatureType)

**HP05:** Situative context presentation/usage which reflect to predefined interesting keywords (qnames)

**HP06:** Definition of Data provider

**HP07:** Level of Detail, Scale

**HP08:** ISO Quality Model, geometric precision, completeness, correctness

**HP09:** Service Type, URL

**HP10:** Certain attribute values or attribute value ranges based on the definition of the selected Thematic constraint, using a controlled vocabulary such as the one used in SeaDataNet.

The context service as specified in this document allows the user to manage harmonization products based on the above parameters. However, the detailed list of parameters / constraints that are part of a harmonization product definition can be found in the document *A5.3-D3 HUMBOLDT Commons Specification / Framework common data model 3.0*.

## 2.1.2 Requirements on the Context Service

This chapter contains both the functional requirements derived from the Scenarios/Use Cases and the non-functional requirements of the HUMBOLDT scenarios and other sources.

For more details on the requirements, please see the HUMBOLDT Requirements Site (<http://humboldt.etra.es/>). Note that the REQ\_ID is given according the available identifiers there. The rationales, acceptance criteria and some comments of the requirements are described in the aforementioned site.

### Functional Requirements

<b>REQ_ID</b>	<b>Description</b>
REQ_039	The system should enrich the standard interfaces with harmonization requirements without having to change any of the existing interfaces. Please, note that this requirement is satisfied by the joint capabilities of several HUMBOLDT components, not just the Context Service.
REQ_040	The system should provide the means for saving the user personal style (including maps or reports), both for the working sessions and the outputs to be produced.
REQ_041	The system must provide a central configuration point for individual and organizational harmonisation constraints.
REQ_042	The system must provide a central configuration point for user management and link them to contexts (preferences).
REQ_043	The system should identify the initial user's preferences from the local machine relevant properties and the specific application domain context.
REQ_044	The client application should be able to access the configuration point (see REQ_041 and REQ_042), which can either be local, organization-wide or global.
REQ_045	The system must provide the user with an Identifier/location for his saved context.
REQ_046	The context identifier (see REQ_045) should contain information of the user's context, such as his organization's and level domain.
REQ_047	The system shall maintain a set of configuration parameters to use as defaults for geographical data queries.
REQ_048	The system must provide a context identifier to easily access and display the content, independent of the client application.
REQ_049	The system must be able to create, change and save Contexts.
REQ_050	The system must save and retrieve his work as a Web Map context locally.

**Table 3: Functional Requirements for this component**

## Non-Functional Requirements

The Context service should fulfill these main non-functional requirements:

REQ_ID	Description
REQ_051	The Context Service must be made available as a Web Service to be used by both client applications and other HUMBOLDT service components.

**Table 4: Non-Functional Requirements for this component**

## 2.2. Actors in this component

### 2.2.1 End Users

The HUMBOLDT End-users consist of a large group who want to solve a problem and decide to use geospatial data for their applications / purposes –they are not interested in the harmonization process itself but only in its results (refer to User Involvement Document – Third Version).

Two further subgroups can be distinguished within this user role: *End-users of geodata*, who are directly working geographical data itself –harmonized or not or integration at all- and *End-users of spatial information*, who use geodata indirectly –using navigation systems, online routing services, etc. An example of the first group would be *GIS experts* and of the second one *decision makers, citizens...*

Both of two subgroups are main users in the Context Service. They need this service to use the common context defined by data custodians and data integrators or to re-define it according to their **user specific constraints** (e.g. bounding box coordinates and map projection of displayed map) or organizations constraints. Therefore they will be able to retrieve the geoinformation in compliance with their specific harmonization requirements.

### 2.2.2 Data Custodians (Data Providers)

HUMBOLDT Data Custodians are people or institutions who are offering data which have been adapted to given standards (harmonized) because of legal or market requirements. They provide data and have the responsibility to do so in a format that may be different from the one they normally use. They are mainly data providers (refer to User Involvement Document – Third Version). Examples of this group are the *national INSPIRE-responsible bodies, territorial governments in charge, mapping agencies* etc.

Data Custodians use the Context Service to define and serve **generic contexts** according their generic data harmonization requirements keeping in mind the target of the data they are responsible for. The target of data refers to the domain where the data they are providing is going to be used. One example could be to define the *resolution constraint* which defines the best resolution of the harmonized map in the area where the territorial government is responsible.

### 2.2.3 Data Integrators (Service Providers)

HUMBOLDT Data Integrators are people who have to use heterogeneous geodata to meet the requirements of their daily job (e.g. integration of data for complex analysis). They need data themselves and access this from different facilities potentially in different formats. They have to combine various data sources, harmonize them to make use of them for their own purposes. They are

mainly service providers (refer to User Involvement Document – Third Version). Some examples of this group are *planners and officers, but also researchers*.

Data Integrators use the Context Service for defining the **common context** for the group of end users which are going to use the service which they are providing. This is the product definition. For example, officials doing the planning of routes to be made accessible and to be marketed for tourists based on a full set of data ranging from available paths to hydrology and habitats information; they are offering a service to citizens integrating heterogeneous geodata -service which has to be defined with a default context. Other situation would be planners making municipal plans; again they have to define a final product to be reviewed by decision makers.

### 2.3. Business process overview

This section gives an overview of the general business processes involving the Context Service component and also its *unique selling proposition (USP)*.

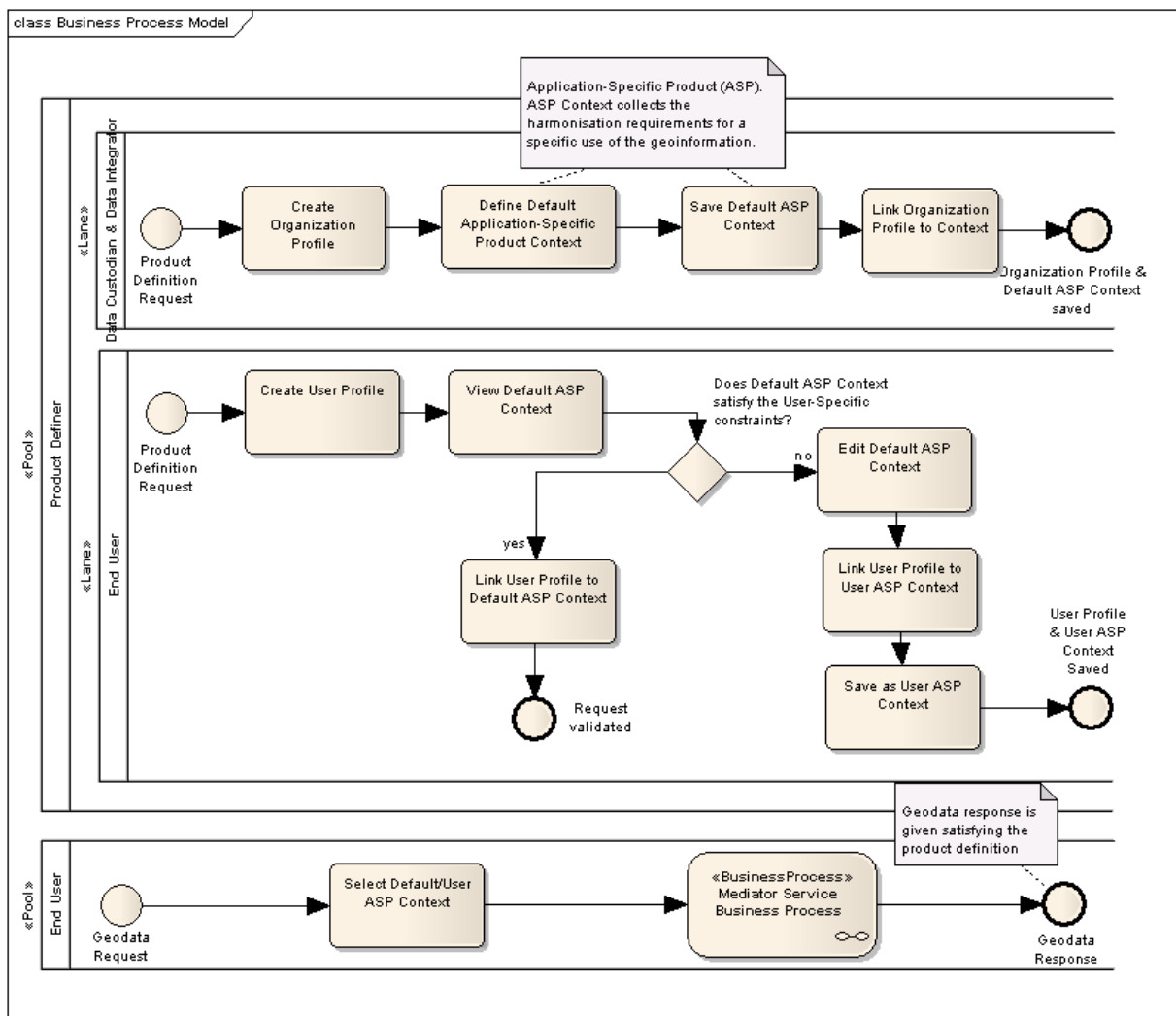


Figure 1: Context Service Business Process

The business process depicted in the above figure shows two main activities. The first one is focused on *Product Definition* and the second one on *Data Retrieval*:

**1. Product Definition:** This is the main purpose of this service. It consists of defining the user preferences (requirements) when geo information is requested. Data Custodians or Data Integrators should define Default Contexts for specific domains' or scenarios' users when they are publishing harmonized geodata. Therefore the named *Default Context* refers to the common requirements for possible stakeholders according to their organization profile or domain profile; it also includes the conceptual schema to be used in its application domain. The organization profile should be linked to a specific *Default Context*, which will be able to be re-defined by end users according to their specific constraints. So they can use either *Default Context* pre-defined by Data Custodians/Data Integrators for them or their own *User Contexts*.

**2. Data retrieval:** In case, a user requests data from the HUMBOLDT Mediator Service the Context Service provides the user preferences. This context gives to the Mediator Service the constraints which will define the necessary fusion, transformation and encoding processes.

## 2.4. Scenario Integration

All HUMBOLDT scenarios: Border security, Urban planning, Forest, Protected areas, Water, Ocean, Galileo, EriskA and the new ones, require context information whenever the HUMBOLDT system is accessed. More in general, context and user preferences information are always used to tailor what (and how) is going to be processed and presented. For this reason the CS usage within the scenario is shown in the context of the already described end users scenarios. A concrete example focused on Protected Areas scenario is going to be given in this section. The complete example covering all steps and all HUMBOLDT components involved can be found in the document *A5.2-D3 [3.1] Specification Introduction and Overview*.

*Mario*, an end user of geodata / geoinformation wants to receive data on *sustainable hiking paths*. He has further constraints on the data. For setting these requirements and constraints, *Mario* accesses the *HUMBOLDT Context Service* by using the *Context Service Client*. The context he specifies using is shown in Table 1 and explained in the following.

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

**Table 1: A user context**

The Feature Type requested is *Sustainable Hiking Paths*. This Feature Type is picked from a certain conceptual schema, namely *Integrated Protected Areas*. The data to be returned should adhere to the reference system *WGS84* and additionally cover a certain area represented by the bounding box or spatial constraint. Further, the requested language is *italian* and the data sets involved must have been *collected within the last five years*.

#### A5.2-D3 [3.4] Context Service Component Specification

Now, Mario accesses the *HUMBOLDT Mediator Service* via some OGC conformant client and passes the *id* of his newly created context. Depending on the operation he invoked, he receives either the capabilities (in case of *GetCapabilities*) or geodata (in case of *GetFeature/GetMap/GetCoverage*). The capabilities or the data is harmonised to the context, this means, only capabilities or data related to the FeatureType "**Sustainable Hiking Paths**" are retrieved. Further, those data sources are retrieved (or included in the capabilities) that either satisfy all constraints imposed by the user context directly or those that can be transformed such that they satisfy the constraints.

### 3 Computational viewpoint

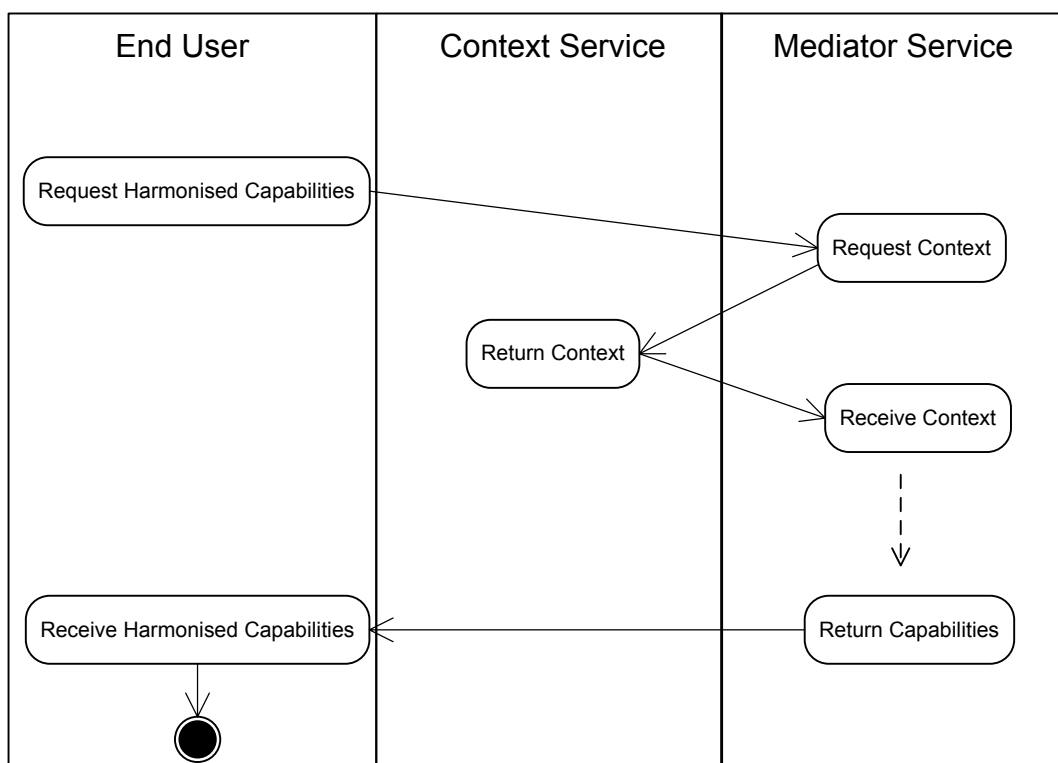
This section gives more detailed information on the processes and interplay of components, the Context Service is involved in.

#### 3.1. How is a Context used?

Organisations and users specify a context that contains constraints on the data sets they are interested in. The context then plays a role in two different processes, as described in the following.

##### 3.1.1 Discovery

First, in case the user issues a *GetCapabilities* request via the interface of the *HUMBOLDT Mediator Service*, the returned capabilities file is assembled based on the context. Hence, the context is used for discovery in HUMBOLDT and the capabilities users receive represent a snapshot of the *Spatial Data Infrastructure* (SDI) known to the MS, specifically tailored to the context. Note: For this reason, what comes back in the capabilities file can change even if the context remains the same, since the capabilities are calculated on the fly each time the *GetCapabilities* is invoked. This means, the discovery results, i.e. what is included in the capabilities, can improve over time when more and more data sources are registered to the system.



**Figure 2: The role of the Context Service when building the capabilities**

Figure 2 shows the interplay of components (MS and CS) when building the capabilities. The dotted arrow indicates that other HUMBOLDT framework components are needed for building the capabilities. The interface specification of the CS as exposed to the MS can be found in **Annex A** in the form of a WSDL-document.

### 3.1.2 Retrieval

The context is further used as a source for filtering / transforming instance data. This transformation can either be applying a filter on the grounding service that delivers the data, applying some processing implemented within the framework or applying some external processing component, e.g. external WPS registered to the *HUMBOLDT Workflow Design and Construction Service*, specified in the document A5.2-D3 [3.5]. As an example, consider the context containing the constraint that all data must be available in the spatial reference system WGS84. In such case, the capabilities delivered by the MS would contain all data sources that satisfy this constraint either directly (e.g. the default SRS of a layer served by a WFS is WGS84) or that can be transformed to WGS84. In case, the user requests a FT / Layer (via GetFeature, GetMap or GetCoverage) that needs to be transformed to the target reference system, the system transforms the data automatically such that it satisfies the constraint.

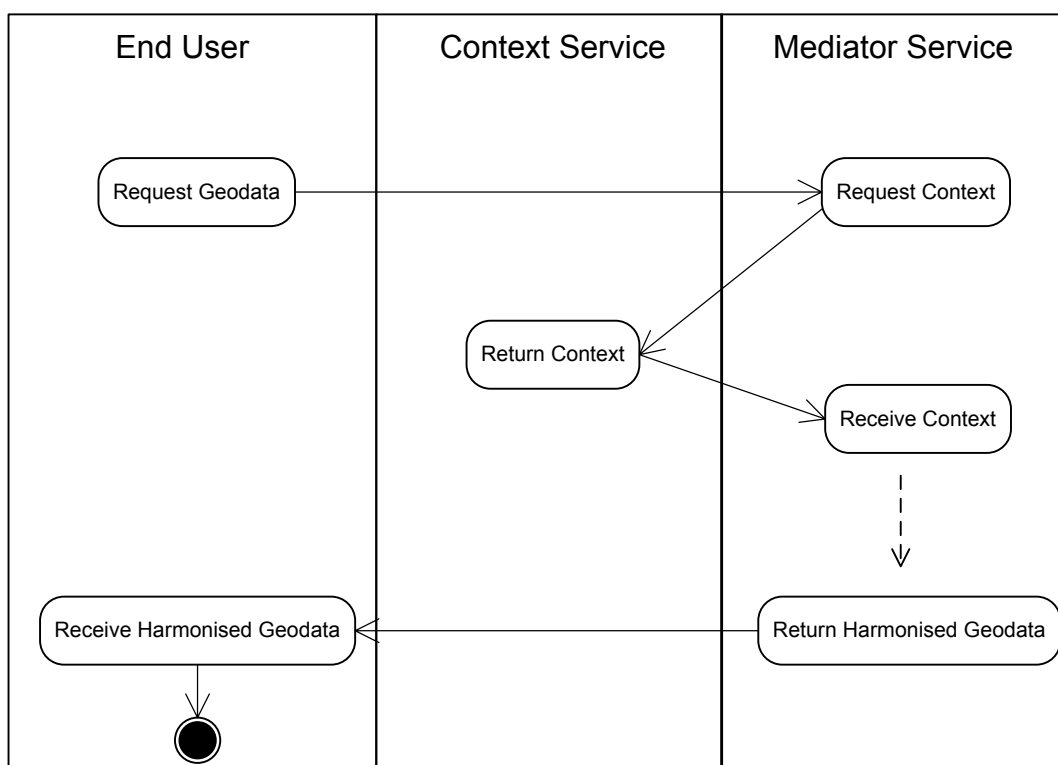


Figure 3: The role of the Context Service in data retrieval

Figure 3 shows the interplay of components when the user requests geodata. The dotted line indicates that other HUMBOLDT components, mainly the *HUMBOLDT WDCS* and the *HUMBOLDT Information Grounding Service* (IGS), specified in A5.2-D3 [3.7] support the MS in achieving this task.

### 3.1.3 HUMBOLDT Contexts vs. OGC Web Map Contexts

As described above, HUMBOLDT Contexts are sets of constraints on data sets that are used for discovery and retrieval. OGC Web Map Contexts are different in the sense that they do not contain constraints on data sets but store the state of a *WMS client application*, e.g. by storing the bounding box that is currently displayed as well as pointers to specific WMS that provide the layers.

Therefore, HUMBOLDT contexts allow users to specify **the behaviour of an OGC conformant web service**. For example, in case a conceptual / application schema is specified within a context, the HUMBOLDT Mediator Service acts as if it only serves data specified in that particular schema.

#### A5.2-D3 [3.4] Context Service Component Specification

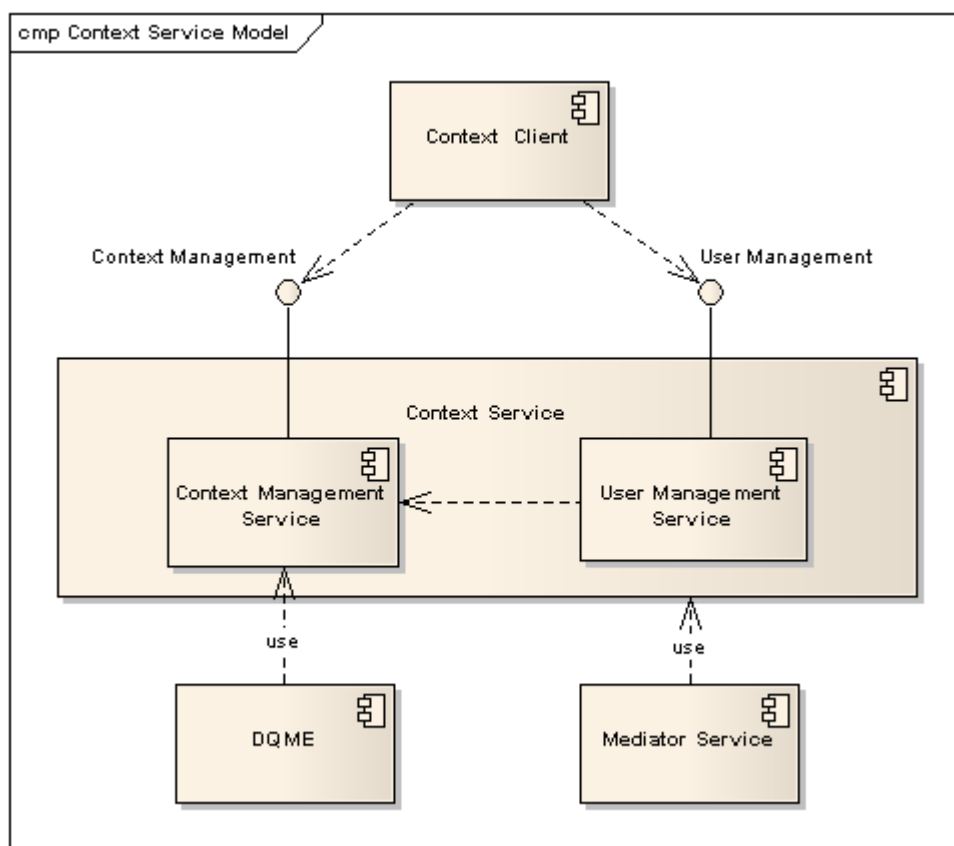
However, we want to enable users to additionally store the state of their map-view client applications using the OGC Web Map Context Specification standard (OGC WMC). Such a document would then include all information necessary for clients to display / reproduce maps delivered by the HUMBOLDT Mediator Service, probably composed of different layers etc. Since the HUMBOLDT Mediator Service acts differently (provides different layers etc.) depending on the *HUMBOLDT context* used, such an OGC WMC document is always specifically tailored to a HUMBOLDT context and therefore attached to it. However, a Web Map Context is not part of a context since the context is exchanged between different HUMBOLDT components (namely the CS and MS) and a WMC is only relevant on client side and is not used within the framework.

To give an example: A user specifies in his context a list of schemas to be used, e.g. a list of INSPIRE application schemas. Hence, the HUMBOLDT Mediator Service acts as if it would only provide data sets that deliver the data within that schema. Further, assume the user requests the WMS capabilities from the HUMBOLDT Mediator Service and subsequently requests some layers via the *GetMap* operation. After retrieving the data, he wants to store the map view. His WMC enabled client is capable of building a WMC file. The user can then store this via invoking the *storeWebMapContext(..)* method on the context service, offered via a web service interface (see Annex A), by passing the WMC and the context identifier of the *HUMBOLDT context*, the WMC is attached to.

## 3.2. Logical Architecture of the Context Service

This chapter outlines the logical architecture of the Context Service. It gives an overview of the composition of this service; describing its internal modules, interfaces and interactions with other components.

As described in Figure 4, the Context Service exposes two main interfaces through which a consumer can access its capabilities: to specify harmonization and business constraints -*context management*; and to manage and link contexts with user profiles -*user management*. Because of that, two modules compose the Context Service: the *CMS (Context Management Service)* and the *UMS (User Management Service)*, which are linked between them in order to provide a specific context to a user profile.



**Figure 4: Component diagram for the Context Service**

The Context Client can be either a Web Frontend (of a web portal where geo-information is provided), or a standard GIS Client allowing user and context configuration. Both of them are offering a dedicated GUI to provide administration services in order to create and edit users and organizations, and also to edit the preferences of these users. The Context Service manages and links this information enriching the OGC standard interfaces.

### 3.2.1 Description of the Context Management Service

**Full Name:** Context Service (CS) → Context Management Service (CMS)

### Responsibilities of the Component/Module

- The component manages contexts, i.e. it allows the creation, modification and removal of a context. This includes Web Map Context 1.1 documents, which can be created for every managed context.
- The component allows administrators to specify harmonization and business constraints for users who want to use a standard GIS client.

### Collaboration

- The Context Client enables users to manage (create, update, delete etc.) contexts.
- The CMS uses the UMS for creating DefaultContexts from the client profile.
- The Mediator Service uses the CMS for retrieving the context using a context identifier.

### Actions fulfilled by this Component

- To create, edit and remove the Context, which can contain the Default ASP context, user-specific and organization-specific information.
- To automatically create a Default Context from the user/organization profile.
- To link a Context with a user/organization profile in order to set a list of Contexts per user/organization.
- To provide to the Mediator Service the Context.

### Interface overview

The CMS interface provides the following methods/operations that forms the main interfaces exposed by this module. Please note that there are three methods that are exposed via a web service interface to clients, namely the *getContext(.)*, *getWebMapContext(.)* and *storeWebMapContext(..)* methods. A WSDL-description can be found in Annex A. All other methods / operations are management operations and are supposed to be used via direct java-method invocation by the *Context Service Client*.

Return	Method/remarks
Context	<code>getContext(java.util.UUID contextId)</code> Returns a Context given a unique context identifier. This method is exposed to clients (i.e. mainly the HUMBOLDT Mediator Service) via a web service interface. A WSDL description can be found in Annex A.
String	<code>getWebMapContext(java.util.UUID contextId)</code> Gets the Web Map Context associated to a given HUMBOLDT context. This method is exposed to external clients via WSDL / SOAP (see Annex A).
void	<code>storeWebMapContext(String webMapContext, java.util.UUID contextID)</code> Allows clients to store an OGC Web Map Context attached to a

	HUMBOLDT context (identified by the UUID).
java.util.UUID	<code>createContext(java.util.UUID profileId)</code> Creates a Context automatically from a profile definition given through its identifier. It returns the ContextId of the new Context.
java.util.UUID	<code>putContext(Context context)</code> Creates a new context given the context object to be added to the service. It returns the unique identifier assigned to the newly created context.
boolean	<code>removeContext(java.util.UUID contextId)</code> Removes an existing context given its unique identifier. It returns true if the delete operation is successful.
boolean	<code>updateContext(java.util.UUID contextId, Context newContext)</code> Updates an existing context given its unique identifier. It returns true if the update operation is successful.

**Table 5: API descriptions of the Context Service.**

### 3.2.2 Description of the User Management Service

**Full Name:** User Management Service (UMS)

#### Responsibilities of the Component/Module

- This component handles the configuration for all users and organizations which want to use the data harmonization capabilities that HUMBOLDT offers.
- The component can manage users and organizations, i.e. allow creation, modification and removal of them.

#### Collaboration

- The Context Client uses the UMS for defining the client profile.
- The UMS is used by the CMS for creating DefaultContexts from the client profile.

#### Actions fulfilled by this Component

- To create, get, update and remove a User/Organization, which are the owners of the Context objects managed by this component.

#### Interface overview

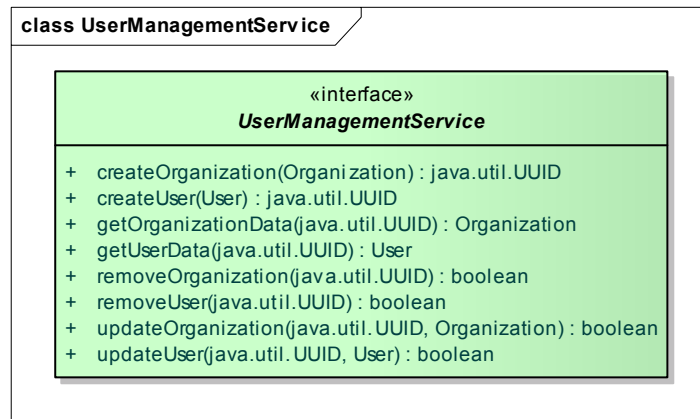
The UMS interface provides the following methods/operations that forms the main interfaces exposed by this module as illustrated in Figure 5.

Return	Method/remarks
--------	----------------

java.util.UUID	<pre>createOrganization(Organization organization)</pre> <p>Creates a new Organization object containing all organization-specific information. It returns the unique identifier for the new Organization object.</p>
java.util.UUID	<pre>createUser(User user)</pre> <p>Creates a new User object containing all user-specific information. It returns the unique identifier for the new User object.</p>
Organization	<pre>getOrganizationData(java.util.UUID organizationId)</pre> <p>Gets the Organization information given its unique identifier. This method should be used by the Context Client to simplify the <code>createContext</code> operation of the CMS. From this information the CMS is able to define some basic constraints as <code>SpatialConstraint</code> or <code>LanguageConstraint</code>.</p>
User	<pre>getUserData(java.util.UUID userId)</pre> <p>Gets the User information given its unique identifier. This method should be used by the Context Client to simplify the <code>createContext</code> operation of the CMS. From this information the CMS is able to define some basic constraints as <code>SpatialConstraint</code> or <code>LanguageConstraint</code>.</p>
boolean	<pre>removeOrganization(java.util.UUID organizationId)</pre> <p>Removes the Organization object given its unique identifier. It could be deleted if (and only if) its user list is empty and there are no other references to this Organization. It returns true is the delete operation is successful.</p>
boolean	<pre>removeUser (java.util.UUID userId)</pre> <p>Removes the User object given its unique identifier. It could be deleted if (and only if) the list of this context in the system is empty. It returns true is the delete operation is successful.</p>
boolean	<pre>updateOrganization (java.util.UUID organizationId, Organization organization)</pre> <p>Edits and stores the Organization specific information given its unique identifier and the new information itself It returns true is the update operation is successful.</p>
boolean	<pre>updateUser (java.util.UUID userId, User user)</pre> <p>Edits and stores the User specific information given its unique identifier and the new information itself. It returns true is the update operation is successful.</p>

**Table 6: API descriptions of the User Management Service.**

A5.2-D3 [3.4] Context Service Component Specification



**Figure 5: Interface of the User Management Service**

## 4 Information Viewpoint

This chapter describes the information viewpoint of the Context Service component and covers data structures that are being used for storage and exchange between the services components defined in the computational viewpoint. The description of specific structures that the CS component exchanges with other components is given in the HUMBOLDT Commons Specification document.

### 4.1. User Management Service

Figure 6 shows the main attributes that are meaningful to define a user and an organization profiles. Where possible, standard types are used, e.g. date/time formats, time zones, UUID, lists and sets.

The operations defined (which are quite self-explanatory) allow an authorized consumer to access these attributes.

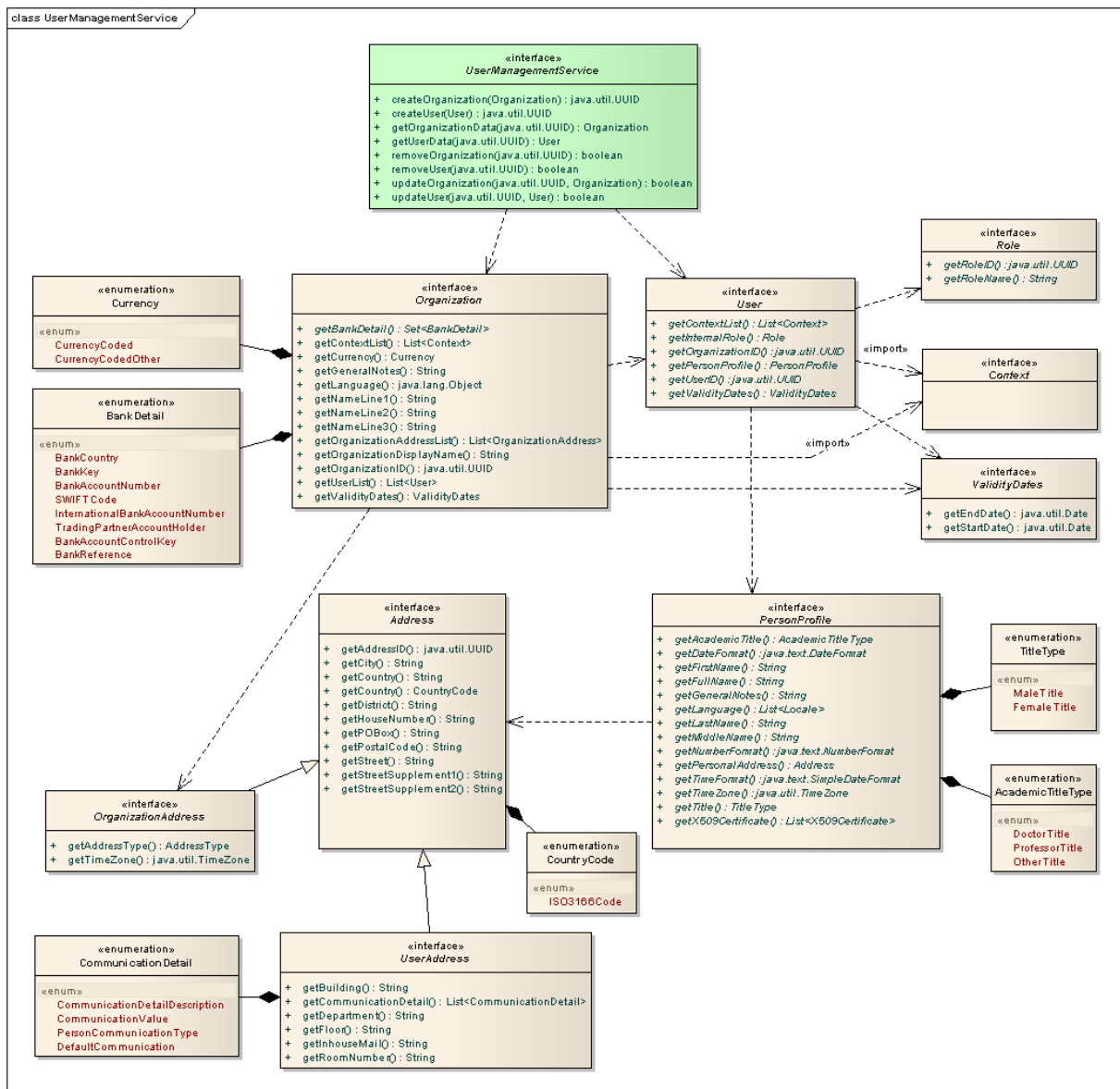
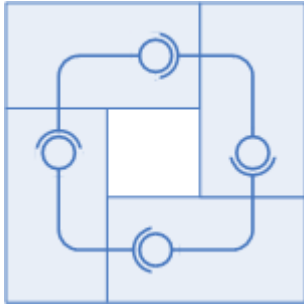


Figure 6: Complete information model for the UMS.

## 5 Summary & Outlook



This specification provided an improvement of the previous Context Service specification –version 2.0 (see <https://intranet.esdi-humboldt.eu/documents/details/751/2>).

The Enterprise Viewpoint has been re-written given more details and examples focused on Protected Areas scenario. Moreover some new capabilities have been added, extending the interfaces already defined and implemented.

## Annex

### Annex A: Context Service WSDL

The HUMBOLDT context service offers a SOAP (document/literal) binding for the *GetContext*, *GetWebMapContext(.)* and *storeWebMapContext(..)* operations, as shown in the WSDL below.

```
<?xml version="1.0" encoding="UTF-16"?>
<wsdl:definitions name="ContextServiceWSDLFile"
targetNamespace="http://esdi-humboldt.eu/schemas/context.wsdl"
xmlns:tns="http://esdi-humboldt.eu/schemas/context.wsdl"
xmlns:cs="http://esdi-humboldt.eu/schemas/context"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:wmc="http://www.opengis.net/context">
  <wsdl:types>
    <xs:schema targetNamespace="http://esdi-
humboldt.eu/schemas/context.wsdl"
xmlns:xs="http://www.w3.org/2001/XMLSchema" >
      <xs:import namespace=http://esdi-humboldt.eu/schemas/context
schemaLocation="http://esdi-
humboldt.eu/schemas/context/context.xsd"/>
      <xs:import namespace="http://www.opengis.net/context"
chemaLocation="http://schemas.opengis.net/context/1.1.0/context
.xsd"/>
      <element name="contextidentifier" type="xs:string"/>
    </xs:schema>
  </wsdl:types>

  <wsdl:message name="getContextResponse">
    <wsdl:part name="context" element="cs:Context"/>
  </wsdl:message>
  <wsdl:message name="getContextRequest">
    <wsdl:part name="contextID" element="contextidentifier"/>
  </wsdl:message>
  <wsdl:message name="getWebMapContextResponse">
    <wsdl:part name="WMContext" element="wmc:ViewContext"/>
  </wsdl:message>
  <wsdl:message name="getWebMapContextRequest">
    <wsdl:part name="contextID" element="contextidentifier"/>
  </wsdl:message>
  <wsdl:message name="storeWebMapContextRequest">
    <wsdl:part name="WMContext" element="wmc:ViewContext"/>
    <wsdl:part name="contextID" element="contextidentifier"/>
  </wsdl:message>
  <wsdl:portType name="contextServicePortType">
    <wsdl:operation name="getContext">
      <wsdl:input message="tns:getContextRequest"/>
      <wsdl:output message="tns:getContextResponse"/>
    </wsdl:operation>
    <wsdl:operation name="getWebMapContext">
      <wsdl:input message="tns:getWebMapContextRequest"/>
      <wsdl:output message="tns:getWebMapContextResponse"/>
    </wsdl:operation>
    <wsdl:operation>
```

A5.2-D3 [3.4] Context Service Component Specification

```
        <wsdl:input message="tns:storeWebMapContextRequest"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="contextServiceSoap"
type="tns:contextServicePortType">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="getContext">
    <soap:operation style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="getWebMapContext">
    <soap:operation style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="storeWebMapContext">
    <soap:operation style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="ContextService">
    <wsdl:port name="contextServicePort"
binding="tns:contextServiceSoap">
        <soap:address location="http://esdi-
humboldt.eu/schemas/context/contextSOAP"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```